ISSN 2007-9737

Application of Auditory Filter-Banks in Polyphonic Music Transcription

Omar Velázquez López, José Luis Oropeza Rodríguez,
Sergio Suárez Guerra

Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Laboratorio de Procesamiento Digital de Señales,
Mexico

{ovelazquezl2018, joropeza, ssuarez@cic.ipn.mx}

Abstract. In this paper we present a frame-level transcription system for polyphonic piano music by using nonnegative matrix factorization (NMF) technique based on Fourier spectrogram as the representation of the musical signal and enhanced by application of an auditory filter-bank based on a new cochlear frequency-position equation, which was developed solving a biomechanical cochlea model without the need of physiological or psychoacoustic experiments. It is important to mention that in our days in music transcription task, a set of auditory bank filters have been used and this paper is focused precisely in this search field. Evaluation using a set of polyphonic piano pieces is performed against the system itself when it does not use filtered spectrograms and also against another system in the state-of-the-art, in both cases it is showed that the proposed method in this paper achieves an increment in precision measure.

Keywords. Automatic music transcription, auditory filter-bank, nonnegative matrix factorization.

1 Introduction

Automatic Music Transcription (AMT) consists of the challenging task in signal processing and artificial intelligence fields that seeks to emulate capability human to perceive the music. It comprises two main subtasks: Multiple Pitch Estimation (MPE) and Onset Detection.

AMT can be organized into four categories depending on the level of abstraction and the structures modeled frame level, note level, stream level, and notation level.

A frame-level transcription system estimates the number and pitch of notes that that coincide in the same frame.

This system does not consider the concept of musical notes or structures [1]. On the other hand, the way humans perceive music is related to their ability to identify signals coming from multiple separate sources.

A transcriber system performs a similar function by detecting notes from each individual source, classifying and grouping them into structures called dictionaries. To perform the above, in the state of the art we can find works that have developed and used machine learning algorithms; among them the nonnegative matrix factorization (NMF) method [2], able of decomposing an input spectrogram as a parts-based representation of sources or notes [3, 4].

Furthermore, cochlear models show the underlying physical processes involved in the auditory perception and have been used in audio applications [5].

In the literature, different methods have been developed for obtaining cochlear functions; in [6] a frequency-position function was found by critical bandwidths experiments while in [7] another function was obtained by an analysis of mechanical resonance. In this paper, we present a frame-level transcription system that estimates multi-pitch and onset time of piano notes by using

doi: 10.13053/CyS-26-4-4271

ISSN 2007-9737

NMF. Firstly, we developed a new cochlear function solving a biomechanical cochlea model and used it for designing auditory filter-banks.

Secondly, we applied NMF algorithm using filtered spectrograms as input and representation of the musical signal. Through the experiments, we show that the proposed system improves the estimation performance further over unmodified spectrograms. Finally, we show all results mentioned before.

2 Auditory Model Analysis

2.1 Cochlear Model and Function Solution

In order to obtain an accurate cochlear frequency-position function, we analyzed a mathematical model of cochlear biomechanics, which is divided into the macro-mechanics and micro-mechanics systems [8].

In the macro-mechanics system, cochlea represents a fluid-filled box, which is separated into upper and lower halves by a flexible partition, while in the micromechanical one it is modeled as a two degrees-of-freedom system, illustrated in figure 1.

The previous model was adapted, from representing a cat cochlea to representing a human cochlea [9]. Besides Perdigao [10] made an electrical-mechanical system equivalence, where the pressure for each point k is defined as following:

$$P(k) = P(k-1)\left(\frac{1-z_{ser}(k)}{z_{eq}(k)}\right). \quad (1)$$

This allowed us to calculate the pressure in the basilar membrane for 0 < x < 4 with 251 points, where each stimulus frequency value corresponding to a x-value, obtaining the graph in figure 2.

Given the behavior of the curve in figure 2, we used a non-linear regression method that allows finding a function that approximates the curve data, considering the exponential function (2):

$$f = ae^{bx}. \quad (2)$$

where f is the frequency and x the position along of the cochlea, and a and b parameters were estimated by least-squares method.

Then, the functions (3) and (4) were obtained:

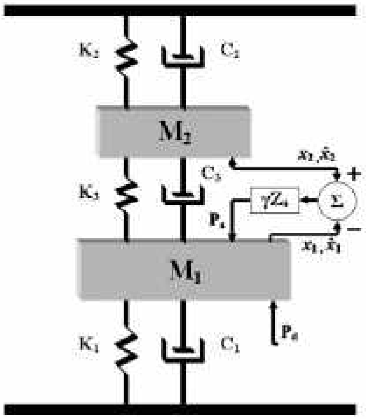


Fig. 1. Block diagram of the micromechanical model of Neely

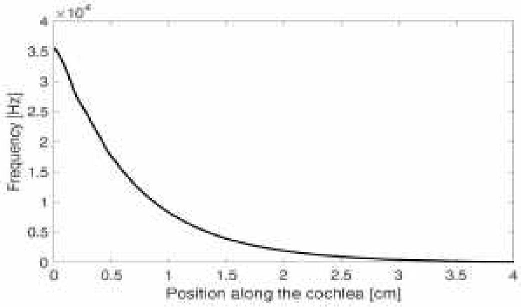


Fig. 2. Frequency as a function of the position along the cochlea

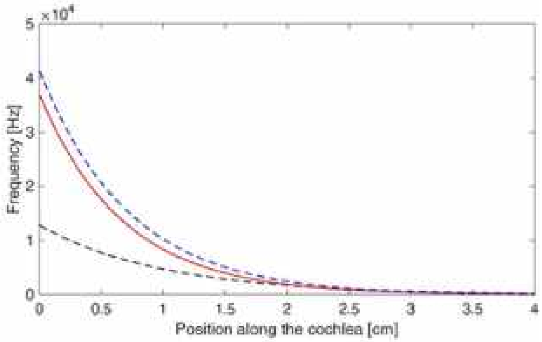


Fig. 3. Frequency as a function of position along the cochlea (a) function (3) in red (b) function [6] in blue (c) function [7] in black:

$$f(x) = 3.695 \times 10^4 e^{-1.485x}, \quad (3)$$

$$x(f) = 7.0824\, ln\, f. \quad (4)$$

doi: 10.13053/CyS-26-4-4271

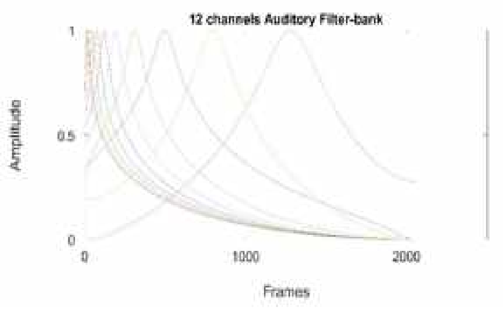


Fig. 4. Bank of twelve auditory filters

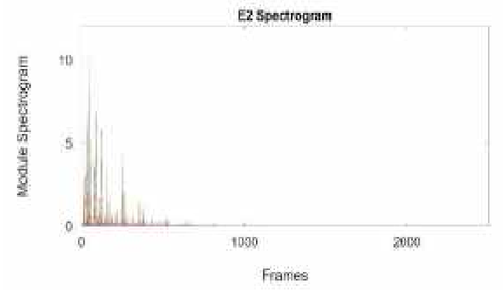


Fig. 5. Fourier spectrogram of E2 piano note

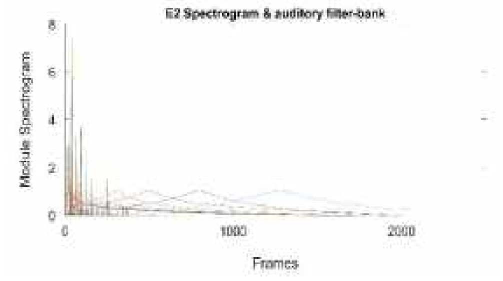


Fig. 6. Auditory filter-bank applied on Fourier spectrogram of E2 piano note

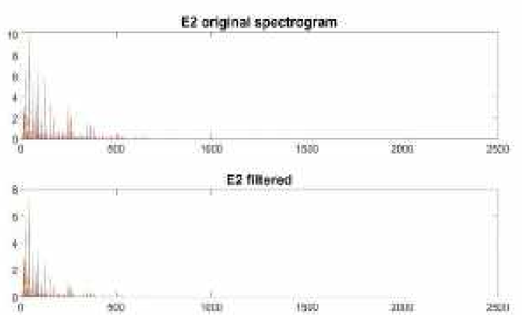


Fig. 7. Comparison between original Fourier spectrogram and filtered spectrogram of E2 piano note

Then, we compare these functions behavior with [6] and [7].

The graph obtained from the function (3) is shown in figure 3.

2.2 Auditory Filter-Bank

The gamma-tone filter is a well-known model, inspired by human auditory system behavior based in auditory “channels”.

The filter widths are normally measured with critical band values and although it is possible to vary them [11], we only focus our attention on the frequency spacing between the channels.

A criterion to allocate frequency spacing between the channels consist of to specify the highest and lowest frequencies along with the desired number of channels and so calculate a set of frequencies uniformly on the equivalent rectangular bandwidth (ERB) scale [12], where lowest frequency depends on the application and the highest value corresponds to half the sampling frequency.

We instead applied the functions (3) and (4) to calculate the center frequency of each channel at the same range and normalized them to constant energy of 1. Results are showed in figure 4.

3 NMF Model and Signal Filtering

3.1 NMF Model

NMF is a popular and used technique in AMT, whose goal is to factorize a nonnegative matrix $\in \mathbb{R}_{\geq 0}^{M\times N}$, a time–frequency representation with $M \in \mathbb{N}$ as the feature dimensionality and $N \in \mathbb{N}$ as the number of elements or frames along the time axis, into two other nonnegative matrices $\boldsymbol{W} \in \mathbb{R}_{\geq 0}^{M\times R}$ and $\boldsymbol{H} \in \mathbb{R}_{\geq 0}^{R\times N}$, both called dictionary and activation matrix, respectively.

The rank $\boldsymbol{R} \in \mathbb{N}$ of the approximation is an important parameter that needs to be specified beforehand, we fix its value depending on the number of pitches in the piece musical.

We applied the version of NMF described and implemented in the toolbox of [13].

Columns of W are labelled and initialized with spectra of synthetized isolated piano notes.

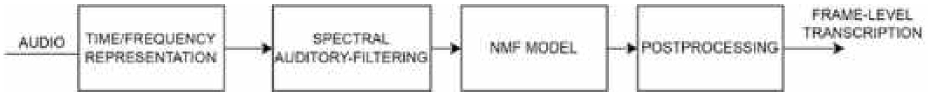

AUDIO
TIME/FREQUENCY REPRESENTATION
SPECTRAL AUDITORY-FILTERING
NMF MODEL
POSTPROCESSING
FRAME-LEVEL TRANSCRIPTION

Fig. 8. Diagram for the proposed polyphonic transcription system

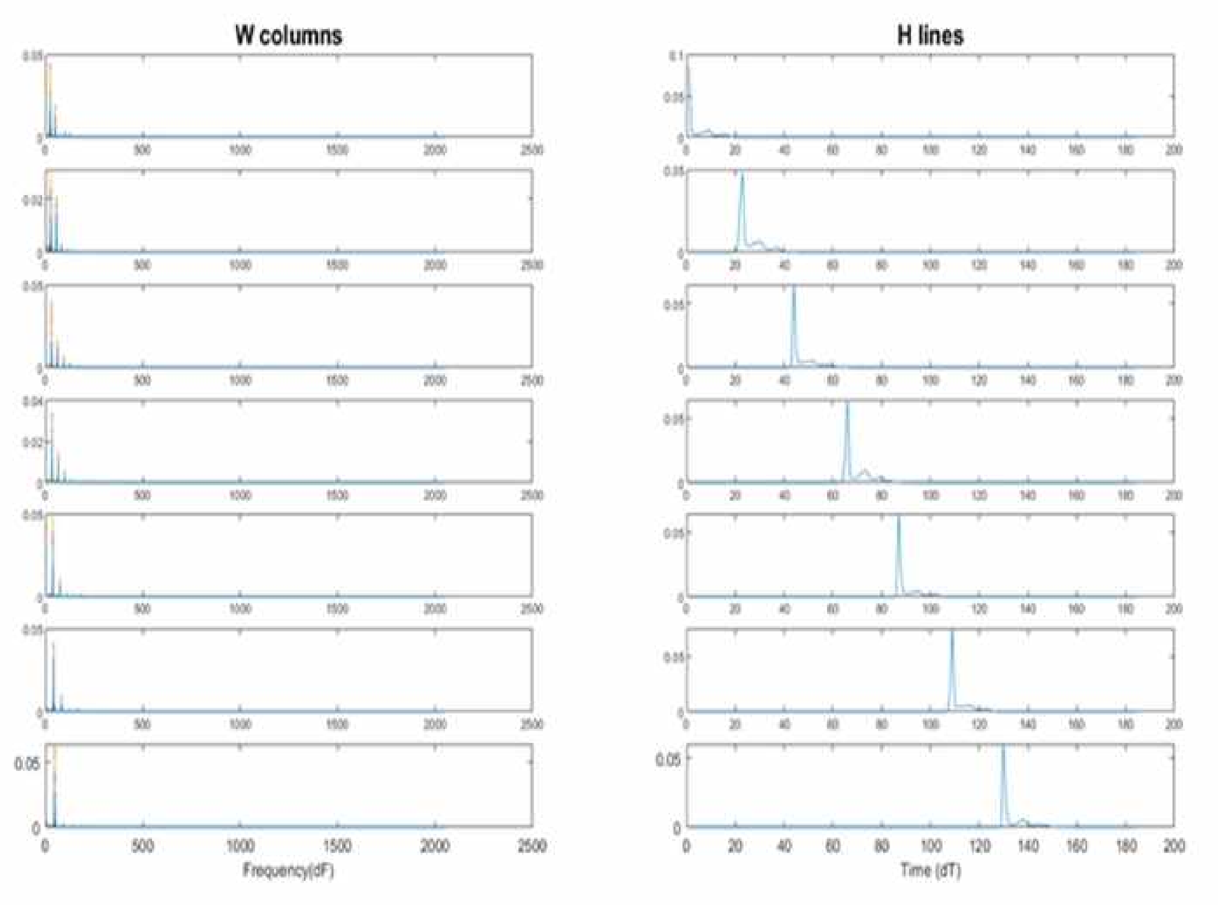


Fig. 9. NMF decomposition of the analysis of Figure 10

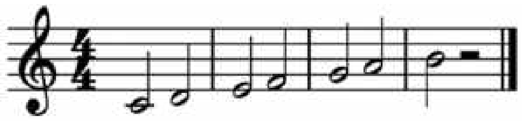


Fig. 10. Example of a major scale in the 4th octave of piano

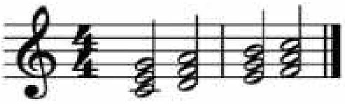


Fig. 11. Example of four piano chords.

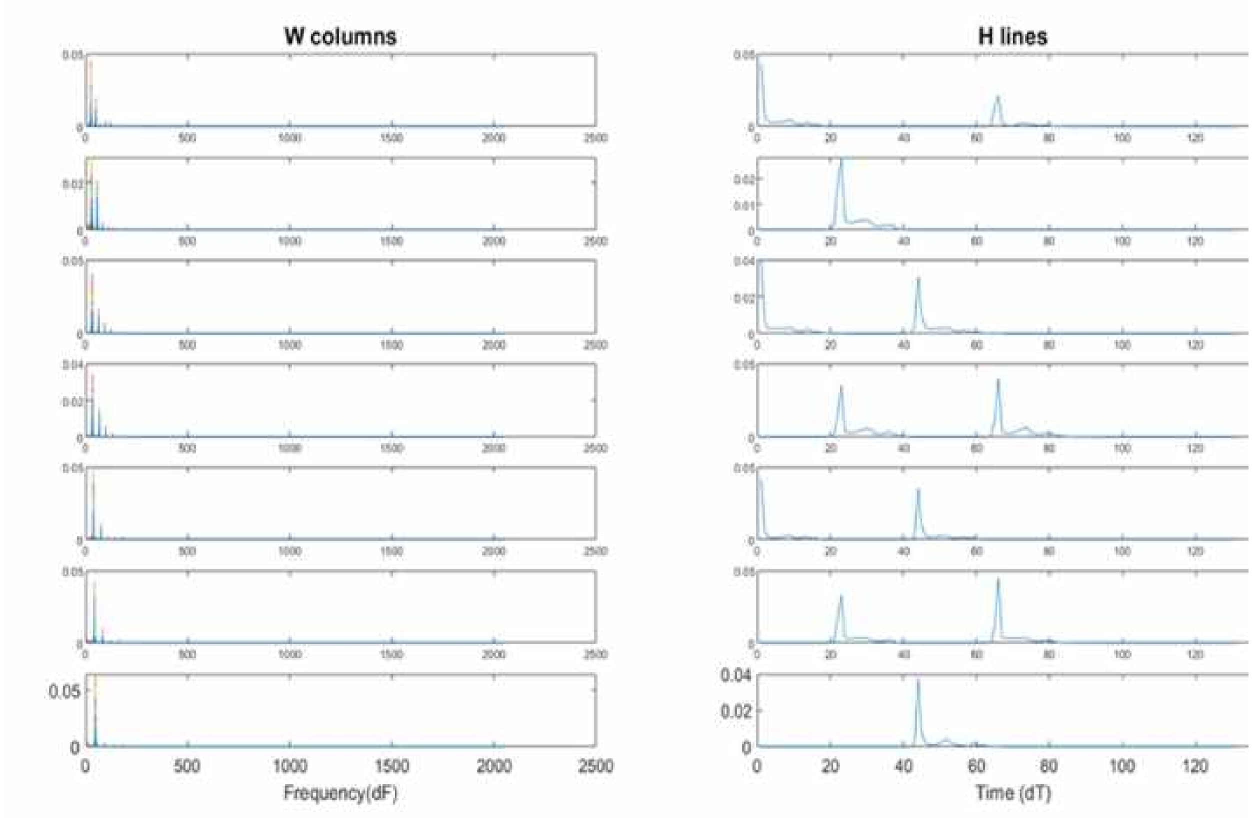


Fig. 12. NMF decomposition of the analysis of Figure 11

3.2 Processing. Signal Filtering

We used the short-time Fourier transform (STFT) to compute the spectrogram of the musical signal, using Hanning window with a 50%, discarding the mirror spectrum (2049 samples per frame). Then we take its module as the non-negative matrix **V**.

The **V** spectrogram is passed through the filter-bank explained on section 2.2. We assume that the modified spectrogram contains the main harmonics of note. Figure 6 and 7 shows the analysis over E2 piano note.

4 Postprocessing

Factors W and H represent pitched elements and their temporal activity, respectively.

Each label of **W** represents the note spectrum and rows of **H** are associated with each one, describing their onsets and offsets.

We looked the optimal detection threshold to get the best performance over the test database. Thus, after different experiments we fixed empirically a threshold of a constant value (1×10^{-4}) on H lines and obtain a note event described by its pitch, onset, and duration.

The product of this step is a frame-level transcription able to be compared to a MIDI file to that serves as the ground truth.

5 Experiments

5.1 Mono and Polyphonic Music Tests

In order to verify an appropriate separation of components and understand the operation of NMF for analyzing alone and simultaneous activity of pitches in music, two preliminary tests were performed.

First test consisted of an analysis over individual notes played. An example of major scale based on C in the 4th octave of piano was created and synthetized to audio using a music software [14].

This scale consists of the pitches C4, D4, E4, F4, G4, A4, and B4, shown in figure 10.

The example was analyzed by NMF, results are shown in figure 9, left graphs represent the frequency content while the ones on the right show temporal occurrences.

The second test was performed over a polyphonic example of execution of four chords in the same octave piano, shown in figure 11.

Each chord C major (C, E, G), D minor (D, F, A), E minor (E, G, B) and F major (F, A, C) was played consecutively.

Although the software used to generate synthetic sounds makes notes of strictly identical pitch and therefore, a rather ideal situation, these test aids to understand operation of the NMF technique, demonstrating that the algorithm is useful in describing activity of each pitch.

Because of the maximum number of notes to recognize for both tests are seven, the number of components in algorithm was set to R = 7.

5.2 Main Experiment

We evaluate the performance of the proposed system using a set of thirty polyphonic classical pieces in MIDI format included in [15] and synthetized to audio (WAVE format) by same software used in 7.1.

We use the whole piece in all cases, MIDI files served as the ground truth and allowed comparison with the performed transcription.

The number of components was set to R = 60, including the notes within the range from the second to the sixth octave of piano.

We worked with the unmodified and filtered STFT spectrogram. Experiments with the different number of filters inside of the auditory filter-bank implementation were performed.

6 Results

6.1 Evaluation Criteria

MIREX [16] criteria for frame-based evaluation in MPE task states to compare and report the transcribed output and the MIDI ground-truth frame by frame using a 10 msec.

Three metrics evaluated in each frame are precision, recall and accuracy. Precision measure is the value of $\frac{TP}{TP+FP}$, recall measure corresponds to $\frac{TP}{TP+FN}$, and overall accuracy is defined by $\frac{TP}{TP+FP+FN}$. Where a true positive (*TP*) is the number of correctly detected pitches, false positive (*FP*) is the number of wrong pitches detected and false negative (*FN*) the number of missing pitches.

6.2 Performance Results

Table 1 shows the average transcription performance over the polyphonic dataset applying 8, 12 and none filters to input spectrogram of the NMF model. Although both recall and accuracy of unmodified spectrogram got over the proposed system, we achieved a better precision average using the proposed auditory filter-bank.

The results show that a spectrogram modified with eight filters gives the best precision value, thus this is the proposed system in this paper.

After we evaluated the performance of the proposed system against another system in the state-of-the-art, SONIC [17], using the same polyphonic dataset. Table 2 shows results for this evaluation. Considering that SONIC achieved solid results at the 12th running MIREX in 2016 on the evaluation of Multiple Fundamental Frequency Estimation & Tracking task and that it is currently available because of the executable file for testing is provided by the author [18], it is a suitable reference of comparison for this work.

Transcriptions of the pieces performed by proposed system, even when they are complex, reach an adequate rate of precision. Typical errors presented during evaluation include note detection errors wrong pitches (depending on the threshold value), difficulty recognizing low-pitched notes, and missing notes in chords of various notes. [4] explains that these same tendencies are present

doi: 10.13053/CyS-26-4-4271

Table 1. Frame-based evaluation of system applying 8, 12 and none filters

Number of filters	**Precision (%)**	**Recall (%)**	**Accuracy (%)**
None	86.09	**26.92**	**25.49**
8	**86.59**	24.96	23.69
12	86.05	24.37	23.07

Table 2. Frame-based evaluation of proposed system vs state of art

	Precision (%)	**Recall (%)**	**Accuracy (%)**
Proposed system	**86.59**	24.96	23.69
SONIC system	85.30	**66.68**	**59.30**

when they analyzed synthetic or real sounds, so a similar performance could be expected in those cases.

Last results in Table 2 show that proposed system achieving a better precision average again. However, in that case, recall and accuracy measures remain much lower. To improve them, a more sophisticated method could be investigated in future.

7 Conclusion and Future Work

We have presented a new approach for designing an auditory filter-bank and applied it in the problem of automatic transcription of polyphonic music. We implemented a NMF toolbox, using a spectrogram filtered and decomposed the audio content of the input music file in factors W and H. After the postprocessing stage gives a frame level transcription and we evaluate its performance.

Although comparison between performance of the system when a filtered or non-filtered spectrogram is used does not highlight a clear superiority of one of them upon the other, evaluation results in precision show that the proposed approach is viable.

Then we use this system vs another one in the state art, achieving similar results.

The work can be improved in four aspects. First, within the set of transcribed pieces, there are some that include notes outside the R range.

Therefore, expanding the R range to the eighty-eight notes of the piano may help improve the performance of the system. Second, even with the application of filters to the time-frequency representation of musical signal, this aspect may improve if Fourier spectrogram is substituted by a representation that avoid its well-known effect of resolution trade-off.

Third, a more sophisticated and finely tuned method should be developed in future for improving the performance in recall and accuracy metrics. Last but not least, although our algorithm creates a basic music sheet from the information of frame-level transcription, a specialized method that comprehend the music structure, say, a note-level transcription, could be developed in future work.

References

1. **Benetos, E., Dixon, S., Duan, Z., Ewert. S. (2019).** Automatic Music Transcription: An Overview. IEEE Signal Processing Magazine, Vol. 36, No. 1. pp. 20–30, DOI: 10.1109/MSP. 2018.2869928.
2. **Lee, D., Seung, H. (1999).** Learning the parts of objects by non-negative matrix factorization. Nature, Vol. 401, pp. 788–791. DOI: 10.103 8/44565.
3. **Smaragdis, P., Brown. J. C. (2003)**. Non-negative matrix factorization for polyphonic music transcription. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (Cat. No.03TH8684), pp. 177–180. DOI: 10.1109/ASPAA.2003.1285860.
4. **Bertin, N., Badeau, R., Richard, G. (2007).** Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the Benchmark. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'07, pp. I-65–I-68. DOI: 10.1109/ICASSP.2007.366617.
5. **Oropeza, J. Guerra S. (2019).** Cochlear mechanical models used in automatic speech

recognition tasks. Computación y Sistemas. Vol. 23, No. 3. pp. 1099–1114. DOI: 10.13053/ cys-23-3-2965.

6. **Greenwood D. D. (1990).** A cochlear frequency-position function for several species-29 years later. The Journal of the Acoustical Society of America, Vol. 87, No. 6, pp. 2592–2605. DOI: 10.1121/1. 399052.
7. **Jimenez, M. (2018).** Distance-frequency relation in a two dimensional cochlear model by mechanical resonance. International Conference on Electronics, Communications and Computers, pp. 106–109. DOI:10.1109/ CONIELECOMP/2018.8327184.
8. **Neely, S. T., Kim, D. O. (1986).** A model for active elements in cochlear biomechanics. J the Journal of the Acoustical Society of America, Vol. 79, No. 5, pp.1472–1480. DOI: 10.1121/1.393674.
9. **Emery, M. K., Elliott, S. J. (2008).** Statistics of instabilities in a state space model of the human cochlea. The Journal of the Acoustical Society of America, Vol. 124, No. 2, pp. 1068–1079. DOI: 10.1121/1.2939133.
10. **Santos-Perdigão, F., Vieira de Sá, L. (1998).** Modelo computacional da cóclea humana**.** Acústica'98 Congreso Ibérico de Acústica, Lisbon.
11. **Slaney, M., Seltzer, M. L. (2014).** The influence of pitch and noise on the discriminability of filterbank features. INTERSPEECH´14, pp. 2263–2267.
12. **Malcolm, S. (1998).** Auditory Toolbox (version 2). Interval Research Corporation Technical, pp. 1–52.
13. **López-Serrano, P., Dittmar, C., Özer, Y., Müller, M. (2019).** NMF Toolbox: Music Processing Applications of Nonnegative Matrix Factorization. Proceedings of the International Conference on Digital Audio Effects DAFx´19, pp. 2–6.
14. **MuseScore.** Accessed April 30, 2022, https://musescore.org/es.
15. **Midi Sheet Music.** Accessed April 30, 2022, http://midisheetmusic.com.
16. **MIREX. (2021).** Music Information Retrieval Evaluation eXchange (MIREX). Accessed April 30, 2022 http://music-ir.org/mirexwiki.
17. **Marolt. M. (2004).** A connectionist approach to automatic transcription of polyphonic piano music. IEEE Trans. Multimedia, Vol. 6, No. 3, pp. 439–449.
18. **SONIC System. (2022). A**ccessed May 30, 2022, http://lgm.fri.uni-lj.si/research/piano-music-transcription.

Article received on 12/02/2022; accepted on 26/05/2022. Corresponding author is José Luis Oropeza Rodríguez.

ISSN 2007-9737

Modelación matemática para zapatas combinadas de correa en esquina apoyadas sobre el terreno: Parte 2

Marina Lourdes Garcia Graciano, Arnulfo Luévanos Rojas,
Sandra López Chavarría, Manuel Medina Elizondo

Universidad Autónoma de Coahuila,
Instituto de Investigaciones Multidisciplinaria,
México

{marina_gagra, arnulfol_2007, sandylopez5}@hotmail.com,
drmanuelmedina@yahoo.com.mx

Resumen. Este trabajo de investigación muestra un modelo matemático para diseño de zapatas combinadas de correa en esquina apoyadas sobre el terreno que soportan una carga concentrada y dos momentos ortogonales en cada columna, y la presión se considera que varía linealmente. La primera parte de esta investigación muestra un modelo para obtener la superficie mínima o área óptima en planta. La metodología se basa en el concepto de que la integral del cortante por flexión es el momento. El modelo tradicional considera una presión uniforme (presión máxima) del suelo sobre la zapata de esquina y las dos zapatas de borde, porque la reacción del suelo sobre cada zapata se aplica en el centro de cada zapata. Cuatro ejemplos numéricos se muestran para el diseño de zapatas combinadas de correa en esquina. Este modelo es más general porque se puede aplicar a zapatas combinadas de esquina, simplemente considerando los lados en dirección X de las zapatas 1 y 3 iguales, y los lados en dirección Y de las zapatas 1 y 2 iguales.

Palabras clave. Modelación matemática, zapatas combinadas de correa en esquina, modelo para diseño.

Mathematical Modeling for Corner Strap Combined Footings Resting on the Ground: Part 2

Abstract. This research work shows a mathematical model for the design of corner strap combined footings supported on the ground that supports a concentrated load and two orthogonal moments in each column and it is considered that the pressure on the ground varies linearly. The first part of this investigation shows a model to obtain the minimum surface or optimal area in plan. The methodology is based on the concept that the integral of the bending shear is the moment. The traditional model considers a uniform pressure (maximum pressure) of the soil on the corner footing and the two edge footings, because the reaction of the soil on each footing is applied in the center of each footing. Four numerical examples are shown for the design of corner strap combined footings. This model is more general because it can be applied to corner combined footings, simply considering the sides in the X direction of the footings 1 and 3 equal, and the sides in the Y direction of the footings 1 and 2 equal.

Keywords. Mathematical modeling, corner strap combined footings, model for design.

1. Introducción

La cimentación es una parte estructural que se localiza por debajo de la superficie del suelo y que transmite las cargas de la estructura al suelo o roca subyacentes.

Todos los suelos, al someterlos a las cargas, se comprimen y causan asentamientos en la estructura soportada.

Los dos requisitos principales en el diseño de cimentaciones son: 1) que el asentamiento total de la estructura esté restringida a una cantidad tolerablemente pequeña; 2) que el asentamiento diferencial de las distintas partes de la estructura se elimine.

La presión del suelo debajo de una zapata depende del tipo de suelo, la rigidez relativa del suelo y la zapata, y la profundidad de la cimentación al nivel de contacto entre la zapata y el suelo.

doi: 10.13053/CyS-26-4-4080

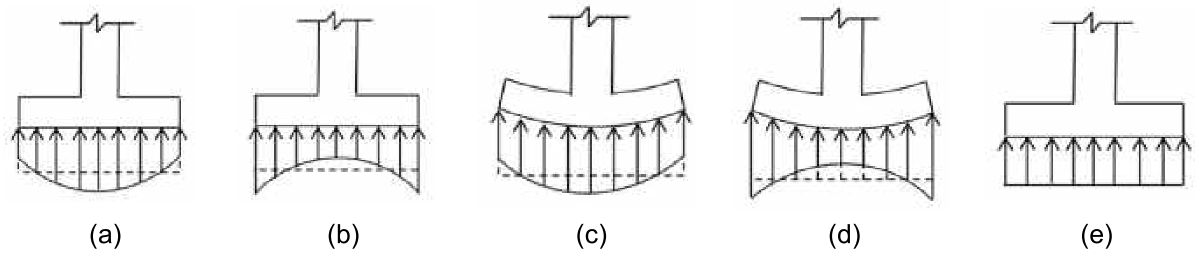


Fig. 1. Diagrama de presiones debajo de una zapata. (a) Zapata rígida sobre suelo arenoso; (b) Zapata rígida sobre suelo arcilloso; (c) Zapata flexible sobre suelo arenoso; (d) Zapata flexible sobre suelo arcilloso; (e) Distribución uniforme usada en el diseño

La Figura 1 muestra la distribución de presión del suelo debajo de la zapata según el tipo de suelo y la rigidez de la zapata.

Los modelos matemáticos para diseño de zapatas aisladas rectangulares, cuadradas y circulares (cimentaciones poco profundas o superficiales) han despertado gran interés en diversos investigadores para obtener soluciones más simplificadas [1-13].

Los modelos matemáticos para diseño de zapatas combinadas como las rectangulares, trapezoidales, en forma de T y de esquina apoyadas sobre el terreno han ofrecido grandes beneficios en la ingeniería estructural e ingeniería geotécnica por el ahorro de tiempo en los cálculos estructurales [14-24].

Los trabajos más cercanos al tema de diseño para zapatas combinadas de correa en esquina son: Yáñez-Palafox *et al.* [21] desarrollaron un modelo matemático para diseño de zapatas combinadas de correa para dos columnas (una columna de frontera y la otra interior), por lo tanto, no se puede usar para obtener el diseño para zapatas de esquina.

Luévanos-Rojas *et al.* [23] propusieron un modelo matemático para diseño de zapatas combinadas de esquina para obtener el espesor y las áreas de acero de refuerzo de la zapata, esta contribución puede ser útil cuando las zapatas tienden a traslaparse una sobre otra, pero cuando las zapatas no se traslapan se deben de usar vigas de correa o de liga (zapatas combinadas de correa en esquina), es decir, las zapatas se unen mediante vigas.

Este trabajo de investigación presenta un modelo para diseño de zapatas combinadas de correa en esquina que soporta una carga concentrada y momentos alrededor de los ejes “X” e “Y” en cada columna, y la presión del suelo se considera que varía linealmente.

La primera parte de esta investigación muestra un modelo para obtener la superficie mínima o área óptima en planta. La metodología se basa en el concepto de que la integral del cortante por flexión es el momento.

El modelo tradicional considera una presión uniforme (presión máxima) del suelo sobre la zapata de esquina y las dos zapatas de borde, porque la reacción del suelo sobre cada zapata se aplica en el centro de cada zapata.

Cuatro ejemplos completos para diseño se muestran para zapatas combinadas de correa en esquina: Ejemplo 1: Lados libres en las direcciones X e Y. Ejemplo 2: Lado restringido en la dirección X y libre en la dirección Y. Ejemplo 3: Lado restringido en la dirección Y y libre en la dirección X. Ejemplo 4: Lado restringido en las direcciones X e Y.

2. Formulación matemática del modelo

Los esfuerzos en las dos direcciones ortogonales (ejes principales X e Y) son:

$$\sigma(x,y) = \frac{R}{A} + \frac{M_{xT}y}{I_x} + \frac{M_{yT}x}{I_y}, \quad (1)$$

dónde: *R, M_{xT}, M_{yT}, A, I_x e I_y* se presentan en las ecuaciones (22) a (25), y en las ecuaciones (30) y (31) de la parte 1.

2.1. Cortantes por flexión y momentos

Los momentos de acuerdo al reglamento [25] se presentan en las secciones *a-a*, *b-b*, *c-c*, *d-d*, *e-e*, *f-f* y *g-g* (ejes paralelos al eje X), y en las secciones *h-h*, *i-i*, *j-j*, *k-k*, *l-l*, *m-m* y *n-n* (ejes paralelos al eje Y) (ver. Figura 2).

Los cortantes por flexión de acuerdo al reglamento [25] se presentan en las secciones *o-o*, *b-b*, *d-d*, *p-p*, *q-q* y *r-r* (ejes paralelos al eje X), y en las secciones *s-s*, *i-i*, *k-k*, *t-t*, *u-u* y *v-v* (ejes paralelos al eje Y) (ver. Figura 3).

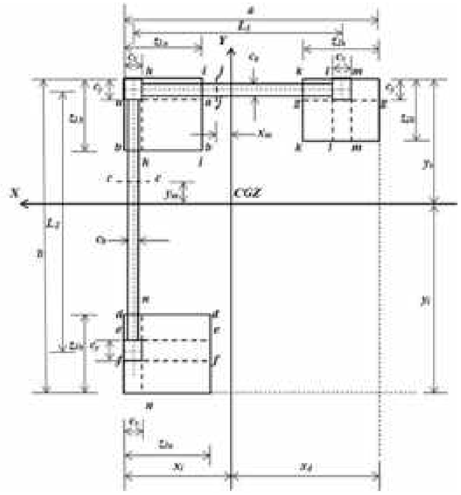


Fig. 2. Momentos críticos de acuerdo al reglamento [25]

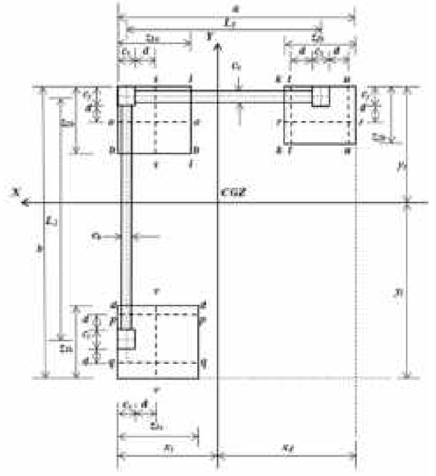


Fig. 3. Cortantes por flexión críticos de acuerdo al reglamento [25]

Los pasos para obtener las ecuaciones son los siguientes:

Paso 1: El cortante por flexión "V_x" sobre el eje X se obtiene por integración cerrada del volumen de presión del área formada partir del lado inferior de la zapata hasta el eje de estudio paralelo al eje "X" (el límite superior en dirección de "Y" se toma como la variable y).

Paso 2: El momento "M_x" sobre el eje X se obtiene por integración y tomando una condición conocida para evaluar las constantes de integración, por ejemplo: para el primer tramo de "$y_s - b \leq y \leq y_s - b + z_{3b}/2$", la condición conocida es "$y = y_s - b$" y "$M_X = 0$", en este intervalo se obtiene $M_{CentroC3}$ (momento en el centro de la columna 3); para el segundo tramo de "$y_s - b + z_{3b}/2 \leq y \leq y_s - b + z_{3b}$", la condición conocida es "$y = y_s - b + \xi/2$" ($\xi = z_{3b}$ para columna sin frontera y $\xi = c_y$ para columna con frontera), y "$M_X = M_{CentroC3} + M_{x3}$", en este intervalo se obtiene M_d (momento en el límite de la zapata 3); para el tercer tramo de "$y_s - b + z_{3b} \leq y \leq y_s - z_{1b}$", la condición conocida es "$y = y_s - b + z_{3b}$", y "$M_X = M_d$", en este intervalo se obtiene M_b (momento en el límite de la zapata 1); para el cuarto tramo de "$y_s - z_{1b} \leq y \leq y_s - c_y/2$", la condición conocida es "$y = y_s - z_{1b}$", y "$M_X = M_b$"; para el tramo de "$y_s - z_{2b} \leq y \leq y_s - c_y/2$", la condición conocida es "$y = y_s - z_{2b}$", y "$M_X = 0$".

Paso 3: Los pasos 1 y 2 se desarrollan de manera similar para obtener el cortante por flexión "V_y" y el momento "M_y" sobre el eje Y, siendo $\lambda = z_{2a}$ para columna sin frontera y $\lambda = c_x$ para columna con frontera.

Las ecuaciones finales para el cortante por flexión y momento se muestran s continuación:

Para el tramo sobre el eje x-x en la zapata 3 de $y_s - b \leq y \leq y_s - b + z_{3b}/2$.

Para el tramo sobre el eje x-x en la zapata 3 de:

$$y_s - b \leq y \leq y_s - b + z_{3b}/2: V_x = -\frac{Rz_{3a}(y_s-b-y)}{A} - \frac{M_{xT}z_{3a}[(y_s-b)^2-y^2]}{2I_x} - \frac{M_{yT}z_{3a}(2x_i-z_{3a})(y_s-b-y)}{2I_y}, \tag{2}$$

$$M_x = \frac{Rz_{3a}(y_s-b-y)^2}{2A} + \frac{M_{xT}z_{3a}[2(y_s-b)^3-3(y_s-b)^2y+y^3]}{6I_x} \tag{3}$$

Para el tramo sobre el eje x-x en la zapata 3 de $y_s - b + z_{3b}/2 \leq y \leq y_s - b + z_{3b}$:

$$V_x = -P_3 - \frac{Rz_{3a}(y_s-b-y)}{A} - \frac{M_{xT}z_{3a}[(y_s-b)^2-y^2]}{2I_x} - \frac{M_{yT}z_{3a}(2x_i-z_{3a})(y_s-b-y)}{2I_y}, \tag{4}$$

$$M_x = \frac{Rz_{3a}(y_s - b - y)^2}{2A} + \frac{M_{xT}z_{3a}[2(y_s - b)^3 - 3(y_s - b)^2 y +}{6I_x} + M_{x3} + P_3\left(y_s - y - b + \frac{\xi}{2}\right). \tag{5}$$

Para el tramo sobre el eje x-x en la viga entre la zapata 3 y la zapata 1 de $y_s - b + z_{3b} \leq y \leq y_s - z_{1b}$:

$$V_x = \frac{R[c_b(y - y_s + b - z_{3b}) + z_{3a}z_{3b}]}{A} + \frac{M_{xT}\{c_b[y^2 - (y_s - b + z_{3b})^2] + z_{3a}z_{3b}(2y_s - 2b + z_{3b})\}}{2I_x} + \frac{M_{yT}[c_b(2x_i - c_x)(y - y_s + b - z_{3b}) + z_{3a}z_{3b}(2x_i - z_{3a})]}{2I_y} - P_3, \tag{6}$$

$$M_x = +P_3\left(y_s - y - b + \frac{\xi}{2}\right) + \frac{R[c_b(y_s - y - b + z_{3b})^2 - z_{3a}z_{3b}(2y_s - 2y - 2b + z_{3b})]}{2A} + \frac{M_{xT}z_{3a}z_{3b}[6(y_s - b)(y - y_s + b - z_{3b}) + z_{3b}(3y - 2z_{3b})]}{6I_x} + \frac{M_{xT}c_b[y^3 - 3y(y_s - b + z_{3b})^2 + 2(y_s - b + z_{3b})^3]}{6I_x} + M_{x3.} \tag{7}$$

Para el tramo sobre el eje x-x en la zapata 1 de $y_s - z_{1b} \leq y \leq y_s - c_y/2$:

$$V_x = \frac{R[z_{1a}(y - y_s + z_{1b}) + c_b(b - z_{1b} - z_{3b}) + z_{3a}z_{3b}]}{A} + \frac{M_{xT}[z_{1a}y^2 - (z_{1a} - c_b)(y_s - z_{1b})^2]}{2I_x} - \frac{M_{xT}[c_b(y_s - b + z_{3b})^2 - z_{3a}z_{3b}(2y_s - 2b + z_{3b})]}{2I_x} + \frac{M_{yT}z_{1a}(2x_i - z_{1a})(y - y_s + z_{1b})}{2I_y} + \frac{M_{yT}[c_b(2x_i - c_x)(b - z_{1b} - z_{3b}) + z_{3a}z_{3b}(2x_i - z_{3a})]}{2I_y} - P_3, \tag{8}$$

$$M_x = \frac{Rz_{3a}z_{3b}(2b - 2y_s + 2y - z_{3b})}{2A} + \frac{Rc_b(b - z_{1b} - z_{3b})(b - 2y_s + 2y + z_{1b} - z_{3b})}{2A} + \frac{Rz_{1a}(y_s - y - z_{1b})^2}{2A} + \frac{M_{xT}z_{3a}z_{3b}[6(b - y_s)(z_{3b} + y) + z_{3b}(3y - 2z_{3b}) - 6(y_s - b)^2]}{6I_x} + \frac{M_{xT}[z_{1a}y^3 + (z_{1a} - c_b)(y_s - z_{1b})^2(2y_s - 3y - 2z_{1b})]}{6I_x} + \frac{M_{xT}c_b(y_s - b + z_{3b})^2(2y_s - 2b - 3y + 2z_{3b})}{6I_x} + M_{x3} + P_3\left(y_s - y - b + \frac{\xi}{2}\right). \tag{9}$$

Para el tramo sobre el eje x-x en la zapata 2 de $y_s - z_{2b} \leq y \leq y_s - c_y/2$:

$$V_x = -\frac{Rz_{2a}(y_s - y - z_{2b})}{A} - \frac{M_{xT}z_{2a}[(y_s - z_{2b})^2 - y^2]}{2I_x} - \frac{M_{yT}z_{2a}(2x_i - 2a + z_{2a})(y_s - y - z_{2b})}{2I_y}, \tag{10}$$

$$M_x = \frac{Rz_{2a}(y_s - z_{2b} - y)^2}{2A} + \frac{M_{xT}z_{2a}[(y_s - z_{2b})^2(2y_s - 2z_{2b} - 3y) + y^3]}{6I_x}. \tag{11}$$

Para el tramo sobre el eje y-y en la zapata 2 de $x_i - a \leq x \leq x_i - a + z_{2a}/2$:

$$V_y = -\frac{Rz_{2b}(x_i - a - x)}{A} - \frac{M_{xT}z_{2b}(2y_s - z_{2b})(x_i - a - x)}{2I_x} - \frac{M_{yT}z_{2b}[(x_i - a)^2 - x^2]}{2I_y}, \tag{12}$$

$$M_y = \frac{Rz_{2b}(x_i - a - x)^2}{2A} + \frac{M_{yT}z_{2b}[2(x_i - a)^3 - 3(x_i - a)^2 x + x^3]}{6I_y}. \tag{13}$$

Para el tramo sobre el eje y-y en la zapata 2 de $x_i - a + z_{2a}/2 \leq x \leq x_i - a + z_{2a}$:

$$V_y = -P_2 - \frac{Rz_{2b}(x_i - a - x)}{A} - \frac{M_{xT}z_{2b}(2y_s - z_{2b})(x_i - a - x)}{2I_x} - \frac{M_{yT}z_{2b}[(x_i - a)^2 - x^2]}{2I_y}, \tag{14}$$

$$M_y = \frac{Rz_{2b}(x_i - a - x)^2}{2A} + \frac{M_{yT}z_{2b}[2(x_i - a)^3 - 3(x_i - a)^2 x + x^3]}{6I_y} + M_{y2} + P_2\left(x_i - x - a + \frac{\lambda}{2}\right). \tag{15}$$

Para el tramo sobre el eje y-y en la viga entre la zapata 2 y la zapata 1 de $x_i - a + z_{2a} \leq x \leq x_i - z_{1a}$:

$$V_y = \frac{R[c_a(x - x_i + a - z_{2a}) + z_{2a}z_{2b}]}{A} + \frac{M_{xT}[c_a(2y_s - c_y)(x - x_i + a - z_{2a}) + z_{2a}z_{2b}(2y_s - z_{2b})]}{2I_x} + \frac{M_{yT}\{c_a[x^2 - (x_i - a + z_{2a})^2] + z_{2a}z_{2b}(2x_i - 2a + z_{2a})\}}{2I_y} - P_2, \tag{16}$$

$$M_y = P_2\left(x_i - x - a + \frac{\lambda}{2}\right) + \frac{R[c_a(x_i - x - a + z_{2a})^2 - z_{2a}z_{2b}(2x_i - 2x - 2a + z_{2a})]}{2A} + \frac{M_{yT}z_{2a}z_{2b}[6(x_i - a)(x - x_i + a - z_{2a}) + z_{2a}(3x - 2z_{2a})]}{6I_y} + \frac{M_{yT}c_a[x^3 - 3x(x_i - a + z_{2a})^2 + 2(x_i - a + z_{2a})^3]}{6I_y} + M_{y2}. \tag{17}$$

Para el tramo sobre el eje y-y en la zapata 1 de $x_i - z_{1a} \leq x \leq x_i - c_x/2$:

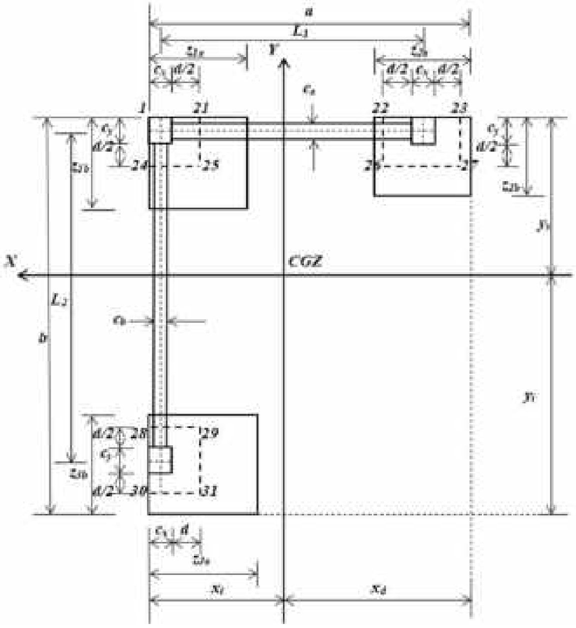


Fig. 4. Cortantes por penetración críticos de acuerdo al reglamento [25]

$$
\begin{aligned}
&V_y \\
&= -P_2 + \frac{R[z_{1b}(x - x_i + z_{1a}) + c_a(a - z_{1a} - z_{2a}) + z_{2a}z_{2b}]}{A} \\
&+ \frac{M_{xT}z_{1b}(2y_s - z_{1b})(x - x_i + z_{1a})}{2I_x} \\
&+ \frac{M_{xT}[c_a(2y_s - c_y)(a - z_{1a} - z_{2a}) + z_{2a}z_{2b}(2y_s - z_{2b})]}{2I_x} \\
&+ \frac{M_{yT}[z_{1b}x^2 - (z_{1b} - c_a)(x_i - z_{1a})^2]}{2I_y} \\
&- \frac{M_{yT}[c_a(x_i - a + z_{2a})^2 - z_{2a}z_{2b}(2x_i - 2a + z_{2a})]}{2I_y},
\end{aligned}
\tag{18}
$$

$$
\begin{gathered}
M_y = \frac{Rz_{2a}z_{2b}(2a-2x_i+2x-z_{2a})}{2A} + \\
\frac{Rc_a(a-z_{1a}-z_{2a})(a-2x_i+2x+z_{1a}-z_{2a})}{2A} + \frac{Rz_{1b}(x_i-x-z_{1a})^2}{2A} + \\
\frac{M_{yT}c_a(x_i-a+z_{2a})^2(2x_i-2a-3x+2z_{2a})}{6I_y} + \\
\frac{M_{yT}\{z_{2a}z_{2b}[6(a-x_i)(z_{2a}-x)+z_{2a}(3x-2z_{2a})-6(x_i-a)^2]\}}{6I_y} + \\
\frac{M_{yT}[z_{1b}x^3+(z_{1b}-c_a)(x_i-z_{1a})^2(2x_i-3x-2z_{1a})]}{6I_y} + M_{y2} + \\
P_2\left(x_i - x - a + \frac{\lambda}{2}\right).
\end{gathered}
\tag{19}
$$

Para el tramo sobre el eje y-y en la zapata 3 de $x_i - z_{3a} \leq x \leq x_i - c_x/2$:

$$
\begin{gathered}
V_y = -\frac{Rz_{3b}(x_i-x-z_{3a})}{A} - \frac{M_{xT}z_{3b}(2y_s-2b+z_{3b})(x_i-x-z_{3a})}{2I_x} - \\
\frac{M_{yT}z_{3b}[(x_i-z_{3a})^2-x^2]}{2I_y},
\end{gathered}
\tag{20}
$$

$$
\begin{gathered}
M_y = \frac{Rz_{3b}(x_i-z_{3a}-x)^2}{2A} + \\
\frac{M_{yT}z_{3b}[(x_i-z_{3a})^2(2x_i-2z_{3a}-3x)+x^3]}{6I_y}.
\end{gathered}
\tag{21}
$$

2.2. Cortantes por penetración o cortantes por punzonamiento

Los cortantes por penetración de acuerdo al reglamento [25] se presentan en un perímetro localizado a una distancia de d/2 a partir de la cara de la columna. Para la columna 1 el perímetro crítico está formado por los puntos 1, 21, 24 y 25.

Para la columna 2 el perímetro crítico está formado por los puntos 22, 23, 26 y 27. Para la columna 3 el perímetro crítico está formado por los puntos 28, 29, 30 y 31 (ver. Figura 4).

Los cortantes por penetración de acuerdo al reglamento [25] se obtienen de la diferencia de la carga axial y el volumen de presión del área delimitada por el perímetro de la sección crítica.

Para la columna 1 es:

$$
\begin{gathered}
V_{p1} = P_1 - \frac{R(2c_x+d)(2c_y+d)}{4A} - \\
\frac{M_{xT}(2c_x+d)(2c_y+d)(4y_s-2c_y-d)}{16I_x} - \\
\frac{M_{yT}(2c_x+d)(2c_y+d)(4x_i-2c_x-d)}{16I_y}.
\end{gathered}
\tag{22}
$$

Tabla 1. Áreas mínimas de los 4 ejemplos

Concepto	Ejemplo							
	1		2		3		4	
	Primera solución	Solución propuesta	Primera solución	Solución propuesta	Primera solución	Solución propuesta	Primera solución	Solución propuesta
σ_a	250							
σ_{ad}	214.15		212.35		212.80		212.35	
I_x	111.70	116.01	108.81	114.28	96.94	98.86	94.11	97.16
I_y	169.90	180.02	143.63	146.89	168.37	179.08	141.50	145.94
M_{xT}	0	58.69	0	201.42	0	-70.29	0	71.15
M_{yT}	0	267.31	0	34.27	0	267.43	0	31.37
R	3200	3200	3200	3200	3200	3200	3200	3200
a	9.25	9.30	8.40	8.40	9.25	9.30	8.40	8.40
b	8.13	8.15	8.12	8.15	7.40	7.40	7.40	7.40
x_i	3.43	3.52	3.43	3.45	3.43	3.52	3.43	3.44
x_d	5.82	5.78	4.97	4.95	5.82	5.78	4.97	4.96
y_s	2.64	2.66	2.64	2.70	2.64	2.62	2.64	2.66
y_i	5.49	5.49	5.48	5.45	4.76	4.78	4.76	4.74
$z_{1a} = z_{1b}$	2.07	2.10	1.90	1.90	1.93	2.00	1.75	1.80
$z_{2a} = z_{2b}$	2.11	2.20	2.35	2.40	2.10	2.20	2.34	2.40
$z_{3a} = z_{3b}$	1.85	1.90	1.83	1.90	2.08	2.10	2.04	2.10
σ_1	214.15	211.63	212.35	211.82	212.80	206.31	212.35	206.25
σ_2	214.15	208.51	212.35	211.37	212.80	203.32	212.35	205.86
σ_3	214.15	201.09	212.35	210.42	212.80	195.71	212.35	204.96
σ_4	214.15	197.82	212.35	209.86	212.80	192.42	212.35	204.44
σ_5	214.15	208.49	212.35	211.28	212.80	203.36	212.35	205.83
σ_6	214.15	201.06	212.35	210.33	212.80	195.74	212.35	204.92
σ_7	214.15	208.33	212.35	210.76	212.80	203.57	212.35	205.61
σ_8	214.15	200.91	212.35	209.80	212.80	195.96	212.35	204.70
σ_9	214.15	210.57	212.35	208.47	212.80	207.73	212.35	204.93
σ_{10}	214.15	210.49	212.35	208.46	212.80	207.66	212.35	204.92
σ_{11}	214.15	210.05	212.35	208.39	212.80	207.21	212.35	204.86
σ_{12}	214.15	207.45	212.35	208.02	212.80	204.75	212.35	204.55
σ_{13}	214.15	199.97	212.35	206.19	212.80	197.27	212.35	203.20
σ_{14}	214.15	196.71	212.35	205.63	212.80	193.99	212.35	202.69
σ_{15}	214.15	208.39	212.35	200.79	212.80	210.00	212.35	202.36
σ_{16}	214.15	207.95	212.35	200.72	212.80	209.56	212.35	202.29
σ_{17}	214.15	205.65	212.35	200.36	212.80	206.94	212.35	201.92
σ_{18}	214.15	208.47	212.35	200.80	212.80	210.08	212.35	202.37
σ_{19}	214.15	207.51	212.35	197.45	212.80	211.57	212.35	200.83
σ_{20}	214.15	204.69	212.35	197.01	212.80	208.44	212.35	200.38
A_{min}	14.94	15.61	15.07	15.51	15.04	15.77	15.07	15.72

Para la columna 2 no limitada en la dirección X (columna de borde):

$$\begin{aligned} V_{p2} &\\ = P_2 - \frac{R(c_x + d)(2c_y + d)}{2A} &\\ - \frac{M_{xT}(c_x + d)(2c_y + d)(4y_s - 2c_y - d)}{8I_x} &\\ - \frac{M_{yT}(c_x + d)(2c_y + d)(2x_i - 2a + z_{2a})}{4I_y}. & \end{aligned} \tag{23}$$

Para la columna 2 limitada en la dirección X (columna en esquina):

$$V_{p2} = P_2 - \frac{R(2c_x+d)(2c_y+d)}{4A} - \frac{M_{xT}(2c_x+d)(2c_y+d)(4y_s-2c_y-d)}{16I_x} - \frac{M_{yT}(2c_x+d)(2c_y+d)(4x_i-4a+2c_x+d)}{16I_y}. \tag{24}$$

Para la columna 3 no limitada en la dirección Y (columna de borde):

$$\begin{aligned} V_{p3} &\\ = P_3 - \frac{R(2c_x + d)(c_y + d)}{2A} &\\ - \frac{M_{xT}(2c_x + d)(c_y + d)(2y_s - 2b + z_{3b})}{4I_x} &\\ - \frac{M_{yT}(2c_x + d)(c_y + d)(4x_i - 2c_x - d)}{8I_y}. & \end{aligned} \tag{25}$$

Para la columna 3 limitada en la dirección Y (columna en esquina):

$$V_{p3} = P_3 - \frac{R(2c_x+d)(2c_y+d)}{4A} - \frac{M_{yT}(2c_x+d)(2c_y+d)(4x_i-2c_x-d)}{16\ {}_y} - \frac{M_{xT}(2c_x+d)(2c_y+d)(4y_s-4b+2c_y+d)}{16I_x}. \tag{26}$$

3. Aplicación numérica

En este apartado se presentan cuatro ejemplos numéricos para obtener el costo mínimo de diseño para zapatas combinadas de correa en esquina que soportan tres columnas, el diseño incluye las tres zapatas y las dos vigas que unen las zapatas como se muestra en la Figura 2 de la parte 1.

Ejemplo 1: Lados libres en las direcciones X e Y, es decir, $c_x/2 + L_1 + z_{2a}/2 = a$ y $c_y/2 + L_2 + z_{3b}/2 = b$. Ejemplo 2: Lado restringido en la dirección X y libre en la dirección Y, es decir, $c_x/2 + L_1 + c_x/2 = a$ y $c_y/2 + L_2 + z_{3b}/2 = b$. Ejemplo 3: Lado libre en la dirección X y restringido en la dirección Y, es decir, $c_x/2 + L_1 + z_{2a}/2 = a$ y $c_y/2 + L_2 + c_y/2 = b$. Ejemplo 4: Lados restringidos en las direcciones X e Y, es decir, $c_x/2 + L_1 + c_x/2 = a$ y $c_y/2 + L_2 + c_y/2 = b$. Para todos los casos se consideran: las mismas cargas y momentos, y las zapatas de diferentes dimensiones y cuadradas.

Los datos a considerar para costo mínimo de las zapatas combinadas de correa en esquina son: $c_x = c_y = 40$ cm; $c_a = c_b = 30$ cm; $L_1 = 8.00$ m; $L_2 = 7.00$ m; H (profundidad a la que se desplanta la zapata) = 2.0 m; $P_{D1} = 250$ kN; $P_{L1} = 350$ kN; $M_{Dx1} = 70$ kN-m; $M_{Lx1} = 80$ kN-m; $M_{Dy1} = 90$ kN-m; $M_{Ly1} = 110$ kN-m; $P_{D2} = 550$ kN; $P_{L2} = 850$ kN; $M_{Dx2} = 110$ kN-m; $M_{Lx2} = 140$ kN-m; $M_{Dy2} = 150$ kN-m; $M_{Ly2} = 200$ kN-m; $P_{D3} = 500$ kN; $P_{L3} = 700$ kN; $M_{Dx3} = 90$ kN-m; $M_{Lx3} = 110$ kN-m; $M_{Dy3} = 130$ kN-m; $M_{Ly3} = 170$ kN-m; f'c (resistencia del concreto) = 28 MPa; f_y (resistencia del acero) = 420 MPa; σ_a (capacidad de carga admisible del suelo) = 250 kN/m^2; γ_{pez} (peso específico de la zapata) = 24 kN/m^3; γ_{pes} (peso específico del suelo de relleno) = 15 kN/m^3.

La capacidad de carga admisible disponible del suelo "σ_{ad}" se obtiene por: la capacidad de carga del suelo "σ_a" se le resta el peso específico de la zapata "γ_{pez}" por el espesor de la zapata, y también el peso específico del suelo de relleno "γ_{pes}" por el espesor del suelo de relleno.

Las cargas y momentos que actúan sobre cada una de las zapatas son: $P_1 = 600$ kN, $M_{x1} = 150$ kN-m, $M_{y1} = 200$ kN-m, $P_2 = 1400$ kN, $M_{x2} = 250$ kN-m, $M_{y2} = 350$ kN-m, $P_3 = 1200$ kN, $M_{x3} = 200$ kN-m, $M_{y3} = 300$ kN-m.

La función objetivo para la superficie mínima (parte 1) se obtiene por la ecuación (25), y las funciones de restricción se obtienen: por las ecuaciones (2) a (24) y (26) a (33) para el ejemplo 1, por las ecuaciones (2) a (24), (26) a (31), (34) y (35) para el ejemplo 2, por las ecuaciones (2) a (24), (26) a (31), (36) y (37) para el ejemplo 3, por las ecuaciones (2) a (24), (26) a (31), (38) y (39) para el ejemplo 4.

Las superficies mínimas para las zapatas combinadas de correa en esquina se obtienen usando el software MAPLE-15.

La Tabla 1 muestra la primera solución y la solución final para el área mínima de los cuatro ejemplos.

Para el diseño de elementos estructurales se deben de factorizar las cargas y momentos que actúan, según el código ACI [25].

Tabla 2. Elementos mecánicos de los 4 ejemplos

Concepto	**Ejemplo**							
	1		**2**		**3**		**4**	
	Permitidos	Actuantes	Permitidos	Actuantes	Permitidos	Actuantes	Permitidos	Actuantes
d_z	57		77		72		77	
d_v	117		117		117		117	
M_{uxT}		128.00		312.00		-56.00		128.00
M_{uyT}		324.00		2.00		324.00		-44.00
R_u		4600.00		4600.00		4600.00		4600.00
M_{uxa}		-2029.16		-2395.75		-2015.79		-2381.65
M_{uxb}		-2401.03		-2569.97		-2544.91		-2697.97
M_{uxc}		-2478.20		-2655.30		-2560.54		-2717.83
$y_c = y_{max}$		1.06		1.34		0.85		1.13
M_{uxd}		-357.76		-377.97		-1622.61		-1653.66
M_{uxe}		303.15		295.24		-10.55		-11.86
M_{uxf}		154.38		150.90		0		0
M_{uxg}		1054.42		1436.27		1037.53		1410.44
M_{uyh}		-1858.93		-1862.50		-2126.67		-2092.70
M_{uyi}		-2431.83		-2575.72		-2553.72		-2665.03
M_{uyj}		-2462.39		-2591.39		-2589.64		-2674.57
$x_j = x_{max}$		1.73		0.95		1.87		1.64
M_{uyk}		-351.47		-2044.41		-367.64		-2062.74
M_{uyl}		475.93		2.93		470.27		2.46
M_{uym}		253.77		0		251.06		0
M_{uyn}		638.04		634.01		896.05		886.21
V_{uxo}	915.25	427.44	1118.64	104.45	1101.06	411.04	1059.77	183.56
V_{uxb}	268.38	-285.34	268.38	-311.36	268.38	-109.31	268.38	-149.16
V_{uxd}	268.38	-657.58	268.38	-693.56	268.38	-404.61	268.38	-454.16
V_{uxp}	828.09	-758.55	1118.64	-682.66	1156.11	-485.88	1236.39	-1016.25
V_{uxq}	828.09	100.33	1118.64	0	1156.11	0	1236.39	0
V_{uxr}	958.84	777.83	1413.02	882.74	1211.16	671.65	1413.02	870.57
V_{uys}	915.25	529.07	1118.64	502.99	1101.06	305.95	1059.77	389.94
V_{uyi}	268.38	-182.91	268.38	84.96	268.38	-212.06	268.38	56.15
V_{uyk}	268.38	-626.17	268.38	-288.39	268.38	-653.35	268.38	-316.90
V_{uyt}	958.84	-836.47	1413.02	-1175.86	1211.16	-765.89	1413.02	-1189.22
V_{uyu}	958.84	207.85	1413.02	0	1211.16	111.09	1413.02	0
V_{uyv}	828.09	518.48	1118.64	394.36	1156.11	611.76	1236.39	560.93
V_{up1}	1791.28 3008.85 1159.06	717.84	2773.06 5329.38 1794.33	673.33	2510.41 4687.65 1624.38	688.97	2773.06 5329.38 1794.33	678.41
V_{up2}	3059.56 4634.56 1979.71	1828.10	2773.06 5329.38 1794.33	1650.63	4360.18 7225.00 2821.29	1780.00	2773.06 5329.38 1794.33	1656.67
V_{up3}	3059.56 4634.56 1979.71	1523.73	4839.60 8215.41 3131.51	1458.85	2510.41 4687.65 1624.38	1548.97	2773.06 5329.38 1794.33	1456.55

Las cargas y momentos factorizados que actúan sobre cada una de las zapatas son: P_{u1} = 860 kN, M_{ux1} = 212 kN-m, M_{uy1} = 284 kN-m, P_{u2} = 2020 kN, M_{ux2} = 356 kN-m, M_{uy2} = 500 kN-m, P_{u3} = 1720 kN, M_{ux3} = 284 kN-m, M_{uy3} = 428 kN-m. Las cargas y momentos factorizados resultantes se obtienen por las ecuaciones (22) a (24) de la parte 1.

Los momentos que actúan sobre cada eje se obtienen según sea el caso: Sustituyendo $y = y_s - c_y$ en la ecuación (9) se obtiene M_a. Sustituyendo $y = y_s - z_{1b}$ en la ecuación (9) o en la ecuación (7) se obtiene M_b.

Sustituyendo $y = y_c$ en la ecuación (7) (intervalo $y_s - b + z_{3b} \leq y \leq y_s - z_{1b}$) o en la ecuación (9) (intervalo $y_s - z_{1b} \leq y \leq y_s - c_y/2$) se obtiene M_c (y_c es el punto donde se ubica el momento máximo, y se obtiene derivando las dos ecuaciones e igualándolas a cero).

Sustituyendo $y = y_s - b + z_{3b}$ en la ecuación (7) o en la ecuación (5) se obtiene M_d. Sustituyendo $y = y_s - b + z_{3b}/2 + c_y/2$ (columna sin frontera) o $y = y_s - b + c_y$ (columna con frontera) en la ecuación (5) se obtiene M_e.

Sustituyendo $y = y_s - b + z_{3b}/2 - c_y/2$ (columna sin frontera) o $y = y_s - b$ (columna con frontera) en la ecuación (3) se obtiene M_f. Sustituyendo $y = y_s - c_y$ en la ecuación (11) se obtiene M_g. Sustituyendo $x = x_i - c_x$ en la ecuación (19) se obtiene M_h.

Sustituyendo $x = x_i - z_{1a}$ en la ecuación (19) o en la ecuación (17) se obtiene M_i. Sustituyendo $x = x_j$ en la ecuación (17) (intervalo $x_i - a + z_{2a} \leq x \leq x_i - z_{1a}$) o en la ecuación (19) (intervalo $x_i - z_{1a} \leq x \leq x_i - c_x/2$) se obtiene M_j (x_j es el punto donde se ubica el momento máximo, y se obtiene derivando las dos ecuaciones e igualándolas a cero).

Sustituyendo $x = x_i - a + z_{2a}$ en la ecuación (17) o en la ecuación (15) se obtiene M_k. Sustituyendo $x = x_i - a + z_{2a}/2 + c_x/2$ (columna sin frontera) o $x = x_i - a + c_x$ (columna con frontera) en la ecuación (15) se obtiene M_l. Sustituyendo $x = x_i - a + z_{2a}/2 - c_x/2$ (columna sin frontera) o $x = x_i - a$ (columna con frontera) en la ecuación (13) se obtiene M_m. Sustituyendo $x = x_i - c_x$ en la ecuación (21) se obtiene M_n.

Los cortantes por flexión que actúan sobre cada eje se obtienen según sea el caso: Sustituyendo $y = y_s - c_y - d$ en la ecuación (8) se obtiene V_o. Sustituyendo $y = y_s - z_{1b}$ en la ecuación (8) o en la ecuación (6) se obtiene V_b.

Sustituyendo $y = y_s - b + z_{3b}$ en la ecuación (6) o en la ecuación (4) se obtiene V_d. Sustituyendo $y = y_s - b + z_{3b}/2 + c_y/2 + d$ (columna sin frontera) o $y = y_s - b + c_y + d$ (columna con frontera) en la ecuación (4) se obtiene V_p. Sustituyendo $y = y_s - b + z_{3b}/2 - c_y/2 - d$ (columna sin frontera) o $y = y_s - b$ (columna con frontera) en la ecuación (2) se obtiene V_q. Sustituyendo $y = y_s - c_y - d$ en la ecuación (10) se obtiene V_r.

Sustituyendo $x = x_i - c_x - d$ en la ecuación (18) se obtiene V_s. Sustituyendo $x = x_i - z_{1a}$ en la ecuación (18) o en la ecuación (16) se obtiene V_i. Sustituyendo $x = x_i - a + z_{2a}$ en la ecuación (16) o en la ecuación (14) se obtiene V_k.

Sustituyendo $x = x_i - a + z_{2a}/2 + c_x/2 + d$ (columna sin frontera) o $x = x_i - a + c_x + d$ (columna con frontera) en la ecuación (14) se obtiene V_t. Sustituyendo $x = x_i - a + z_{2a}/2 - c_x/2 - d$ (columna sin frontera) o $x = x_i - a$ (columna con frontera) en la ecuación (12) se obtiene V_u. Sustituyendo $x = x_i - c_x - d$ en la ecuación (20) se obtiene V_v.

Los cortantes por penetración que actúan sobre cada zapata se obtienen según sea el caso: Por la ecuación (22) para la zapata 1. Por la ecuación (23) para la columna no limitada en la dirección X, y por la ecuación (24) para la columna limitada en la dirección X para la zapata 2.

Por la ecuación (25) para la columna no limitada en la dirección Y, y por la ecuación (26) para la columna limitada en la dirección Y para la zapata 3.

La Tabla 2 muestra el peralte efectivo o profundidad efectiva de las zapatas "d_z", el peralte efectivo o profundidad efectiva de las vigas "d_v", los momentos, los cortantes por flexión y los cortantes por penetración que actúan sobre las tres zapatas, y los cortantes por flexión y los cortantes por penetración que resistente el concreto para los cuatro ejemplos.

La Tabla 2 muestra lo siguiente:

El peralte efectivo o profundidad efectiva mayor de las zapatas se presenta en los ejemplos 2 y 4, y el menor aparece en el ejemplo 1. El peralte efectivo o profundidad efectiva de las vigas son iguales en los cuatro ejemplos.

El momento resultante alrededor del eje X "M_{uxT}" mayor se presenta en el ejemplo 2, y el

Tabla 3. Áreas de acero de los 4 ejemplos

Concepto	Ejemplo							
	1		2		3		4	
	Necesario	Propuesto	Necesario	Propuesto	Necesario	Propuesto	Necesario	Propuesto
Viga 1								
A_{sxsp}	66.93	70.98 (14Ø1”)	71.40	76.05 (15Ø1”)	71.34	76.05 (15Ø1”)	74.37	76.05 (15Ø1”)
A_{sxsmin}	11.70		11.70		11.70		11.70	
A_{sxip}	0	15.21 (3Ø1”)	0	15.21 (3Ø1”)	0	15.21 (3Ø1”)	0	15.21 (3Ø1”)
A_{sximin}	11.70		11.70		11.70		11.70	
A_{vc}	1.98 (1Ø5/8”) @ 23.00 cm		1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 21.00 cm		1.98 (1Ø5/8”) @ 58.00 cm	
A_{ve}	1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 58.00 cm	
Viga 2								
A_{sysp}	67.47	70.98 (14Ø1”)	73.68	76.05 (15Ø1”)	70.32	70.98 (14Ø1”)	75.94	76.05 (15Ø1”)
A_{sysmin}	11.70		11.70		11.70		11.70	
A_{syip}	0	15.21 (3Ø1”)	0	15.21 (3Ø1”)	0	15.21 (3Ø1”)	0	15.21 (3Ø1”)
A_{syimin}	11.70		11.70		11.70		11.70	
A_{vc}	1.98 (1Ø5/8”) @ 21.00 cm		1.98 (1Ø5/8”) @ 19.00 cm		1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 44.00 cm	
A_{ve}	1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 58.00 cm		1.98 (1Ø5/8”) @ 58.00 cm	
Zapata 1								
A_{sx}	0	40.56 (8Ø1”)	0	50.70 (10Ø1”)	0	50.70 (10Ø1”)	0	50.70 (10Ø1”)
A_{sxmin}	39.90		48.77		48.00		46.20	
A_{sy}	0	40.56 (8Ø1”)	0	50.70 (10Ø1”)	0	50.70 (10Ø1”)	0	50.70 (10Ø1”)
A_{symin}	39.90		48.77		48.00		46.20	
Zapata 2								
A_{sx}	22.44	45.63 (9Ø1”)	0.10	65.91 (13Ø1”)	17.45	55.77 (11Ø1”)	0.08	65.91 (13Ø1”)
A_{sxmin}	41.80		61.60		52.80		61.60	
A_{sy}	50.75	55.77 (11Ø1”)	50.57	65.91 (13Ø1”)	14.23	55.77 (11Ø1”)	49.63	65.91 (13Ø1”)
A_{symin}	41.80		61.60		52.80		61.60	
Zapata 3								
A_{sx}	30.36	40.56 (8Ø1”)	22.08	50.70 (10Ø1”)	0	50.70 (10Ø1”)	0	55.77 (11Ø1”)
A_{sxmin}	36.10		48.77		50.40		53.90	
A_{sy}	14.23	40.56 (8Ø1”)	10.21	50.70 (10Ø1”)	33.58	50.70 (10Ø1”)	30.97	55.77 (11Ø1”)
A_{symin}	36.10		48.77		50.40		53.90	

menor aparece en el ejemplo 3 (valor absoluto). El momento resultante alrededor del eje Y “M_{uyT}” mayor se presenta en los ejemplos 1 y 3, y el menor aparece en el ejemplo 2 (valor absoluto).

Las áreas de acero de refuerzo sobre las zapatas combinadas de correa en esquina se obtienen como sigue:

1. Viga 1 (viga que une la zapata 1 y la zapata 2): Acero longitudinal superior principal “A_{sxsp}” se diseña por el momento M_{uyj}. Acero longitudinal inferior principal “A_{sxip}” se diseña con el momento positivo mayor de los momentos M_{uyi} y M_{uyk}, pero en ningún caso debe ser inferior al acero mínimo “A_{smin}” [25]. Acero transversal

se diseña por el cortante V_{uyi} o V_{uyk} el que resulte mayor, estos estribos se colocan entre los ejes i y j, y en los extremos de la viga se coloca acero mínimo por cortante "A_{vmin}" [25].

2. Viga 2 (viga que une la zapata 1 y la zapata 3): Acero longitudinal superior principal "A_{sysp}" se diseña por el momento M_{uxc}. Acero longitudinal inferior principal "A_{syip}" se diseña con el momento positivo mayor de los momentos M_{uxb} y M_{uxd}, pero en ningún caso debe ser inferior al acero mínimo "A_{smin}" [25]. Acero transversal se diseña por el cortante V_{uxb} o V_{uxd} el que resulte mayor, estos estribos se colocan entre los ejes b y d, y en los extremos de la viga se coloca acero mínimo por cortante "A_{vmin}" [25].
3. Zapata 1: Acero inferior en la dirección X "A_{sx}" se diseña por los momentos M_{uyh} y M_{uyi} el que resulte mayor. Acero inferior en la dirección Y "A_{sy}" se diseña por los momentos M_{uxa} y M_{uxb} el que resulte mayor. Si los momentos son negativos se consideran en las vigas, y se propone acero mínimo "A_{smin}" [25].
4. Zapata 2: Acero inferior en la dirección X "A_{sx}" se diseña por los momentos M_{uyl} y M_{uym} el que resulte mayor. Acero inferior en la dirección Y "A_{sy}" se diseña por el momento M_{uxg}. Si los momentos son negativos se consideran en las vigas, y se propone acero mínimo "A_{smin}" [25].
5. Zapata 3: Acero inferior en la dirección X "A_{sx}" se diseña por el momento M_{uyn}. Acero inferior en la dirección Y "A_{sy}" por los momentos M_{uxe} y M_{uxf} el que resulte mayor. Si los momentos son negativos se consideran en las vigas, y se propone acero mínimo "A_{smin}" [25].

La Tabla 3 muestra las áreas de acero finales para las tres zapatas y las dos vigas para los cuatro ejemplos.

4. Resultados

La Tabla 1 muestra lo siguiente:

El momento de inercia sobre el eje X mayor se presenta en el ejemplo 1, y el menor aparece en el ejemplo 4. El momento de inercia sobre el eje Y mayor se presenta en el ejemplo 1, y el menor aparece en el ejemplo 4.

El momento resultante sobre el eje X mayor aparece en el ejemplo 2, y el menor se presenta en el ejemplo 1 (valor absoluto).

El momento resultante sobre el eje Y mayor aparece en el ejemplo 3, y el menor se presenta en el ejemplo 4. La distancia "a" mayor se presenta en los ejemplos 1 y 3, y el menor aparece en los ejemplos 2 y 4.

La distancia "b" mayor aparece en los ejemplos 1 y 2, y el menor se presenta en los ejemplos 3 y 4.

La distancia X del centro de gravedad del lado izquierdo "x_i" mayor se presenta en los ejemplos 1 y 3, y el menor aparece en el ejemplo 4.

La distancia X del centro de gravedad del lado derecho "x_d" mayor aparece en los ejemplos 1 y 3, y el menor aparece en el ejemplo 2.

La distancia Y del centro de gravedad del lado superior "y_s" mayor se presenta en el ejemplo 2, y el menor aparece en el ejemplo 3.

La distancia Y del centro de gravedad del lado inferior "y_i" mayor aparece en el ejemplo 1, y el menor aparece en el ejemplo 4.

La dimensión "$z_{1a} = z_{1b}$" mayor de la zapata 1 se presenta en el ejemplo 1, y la menor aparece en el ejemplo 4. La dimensión "$z_{2a} = z_{2b}$" mayor de la zapata 2 se presenta en los ejemplos 2 y 4, y la menor aparece en los ejemplos 1 y 3.

La dimensión "$z_{3a} = z_{3b}$" mayor de la zapata 3 se presenta en los ejemplos 3 y 4, y la menor aparece en los ejemplos 1 y 2.

El esfuerzo máximo y el esfuerzo mínimo se presenta en $\sigma_{max} = \sigma_1$ y $\sigma_{min} = \sigma_{14}$ (ejemplo 1), $\sigma_{max} = \sigma_1$ y $\sigma_{min} = \sigma_{20}$ (ejemplo 2), $\sigma_{max} = \sigma_{19}$ y $\sigma_{min} = \sigma_4$ (ejemplo 3), $\sigma_{max} = \sigma_1$ y $\sigma_{min} = \sigma_{20}$ (ejemplo 4). El área mínima "A_{min}" mayor se presenta en el ejemplo 3, y la menor aparece en el ejemplo 2.

La Tabla 2 muestra lo siguiente:

El peralte efectivo o profundidad efectiva mayor de las zapatas se presenta en los ejemplos 2 y 4, y el menor aparece en el ejemplo 1.

El peralte efectivo o profundidad efectiva de las vigas son iguales en los cuatro ejemplos.

El momento resultante alrededor del eje X "M_{uxT}" mayor se presenta en el ejemplo 2, y el menor aparece en el ejemplo 3 (valor absoluto).

El momento resultante alrededor del eje Y "M_{uyT}" mayor se presenta en los ejemplos 1 y 3, y el menor aparece en el ejemplo 2 (valor absoluto).

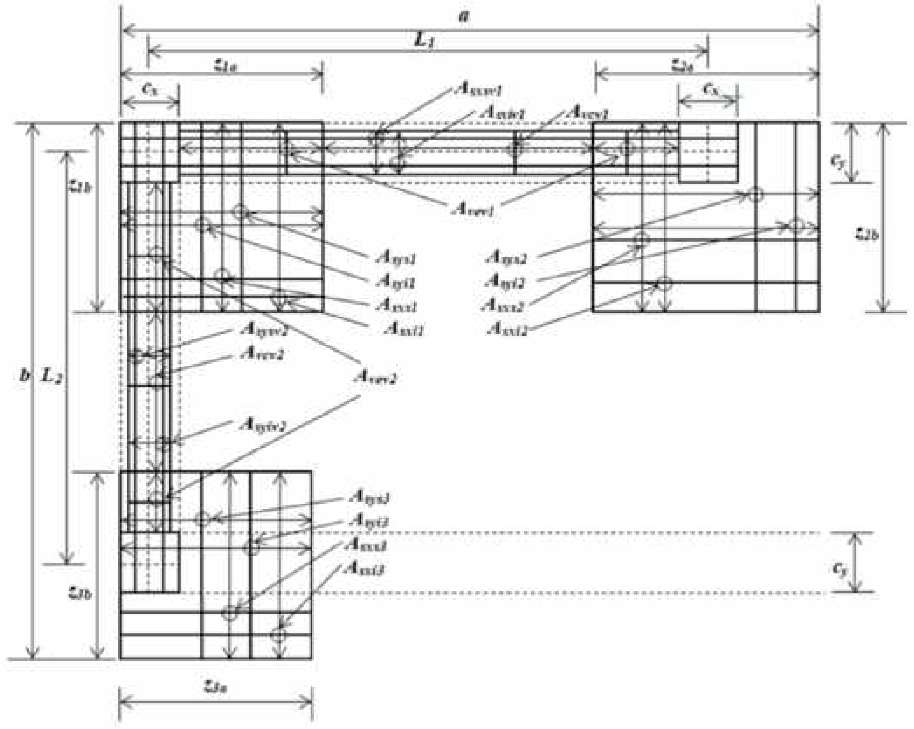


Fig. 5. Acero de refuerzo de la zapata combinada de correa en esquina (las tres zapatas y las dos vigas)

El momento máximo sobre el eje X "M_{uxc}" mayor (valor absoluto) se presenta en el ejemplo 4, y el menor aparece en el ejemplo 1 (valor absoluto).

El momento máximo sobre el eje Y "M_{uyj}" mayor (valor absoluto) se presenta en el ejemplo 4, y el menor aparece en el ejemplo 1 (valor absoluto).

Los cortantes por flexión que actúan cumplen para todos los ejemplos con los permitidos por el concreto, con excepción de los cortantes por flexión "V_{uxb} y V_{uxd}" (viga 2), y "V_{uyi} y V_{uyj}" (viga 1), porque las vigas admiten acero de refuerzo para soportar el cortante excedente por el concreto.

Los cortantes por penetración que actúan cumplen para todos los ejemplos con los permitidos por el concreto, también se observa que el cortante por penetración que rige para el ejemplo 1 es "V_{up2}", para el ejemplo 2 es "V_{up2}", para el ejemplo 3 es "V_{up3}", y para el ejemplo 4 es "V_{up2}".

La Tabla 3 muestra lo siguiente:

1. Viga 1 (viga que une la zapata 1 y la zapata 2): Acero longitudinal superior principal "A_{sxsp}" mayor se presenta en los ejemplos 2, 3 y 4, y el menor aparece en el ejemplo 1. Acero longitudinal inferior principal "A_{sxip}" se presenta el mismo para los cuatro ejemplos y se diseña con el mínimo ya que no hay momentos positivos.Acero transversal en la parte central "A_{vc}" mayor se presenta en el ejemplo 3 (separación mínima), y el menor aparece en los ejemplos 2 y 4 (separación máxima). Acero transversal en los extremos "A_{ve}" se propone acero mínimo "A_{vmin}" [25], porque el cortante por flexión lo soporta el concreto.
2. Viga 2 (viga que une la zapata 1 y la zapata 3): Acero longitudinal superior principal "A_{sysp}" mayor se presenta en los ejemplos 2 y 4, y el menor aparece en los ejemplos 1 y 3. Acero longitudinal inferior principal "A_{syip}" se presenta el mismo para los cuatro ejemplos y se diseña con el mínimo ya que no hay momentos positivos. Acero transversal en la parte central "A_{vc}" mayor se presenta en el ejemplo 2 (separación mínima), y el menor aparece en el ejemplo 3 (separación máxima). Acero

transversal en los extremos “A_{ve}” se propone acero mínimo “A_{vmin}” [25], porque el cortante por flexión lo soporta el concreto.

3. Zapata 1: Acero inferior en la dirección X “A_{sx}” mayor se presenta en los ejemplos 2, 3 y 4, y el menor aparece en el ejemplo 1. Para los cuatro ejemplos rige el acero mínimo “A_{smin}” [25], porque no hay momentos positivos. Acero inferior en la dirección Y “A_{sy}” ocurre lo mismo que en la dirección X. Acero superior en las direcciones X e Y se proporciona acero mínimo “A_{smin}” [25].

4. Zapata 2: Acero inferior en la dirección X “A_{sx}” mayor se presenta en los ejemplos 2 y 4, y el menor aparece en el ejemplo 1. Para los cuatro ejemplos rige el acero mínimo “A_{smin}” [25]. Acero inferior en la dirección Y “A_{sy}” mayor se presenta en los ejemplos 2 y 4, y el menor aparece en los ejemplos 1 y 3. Acero superior en las direcciones X e Y se proporciona acero mínimo “A_{smin}” [25].

5. Zapata 3: Acero inferior en la dirección X “A_{sx}” mayor se presenta en el ejemplo 4, y el menor aparece en el ejemplo 1. Para los cuatro ejemplos rige el acero mínimo “A_{smin}” [25]. Acero inferior en la dirección Y “A_{sy}” mayor se presenta en el ejemplo 4, y el menor aparece en el ejemplo 1. Para los cuatro ejemplos rige el acero mínimo “A_{smin}” [25]. Acero superior en las direcciones X e Y se proporciona acero mínimo “A_{smin}” [25].

   La Figura 5 muestra en detalle el acero de refuerzo de una manera general para la zapata combinada de correa en esquina.

5. Conclusiones

La cimentación de una construcción o edificación es la parte principal para transmitir las cargas de columna o pared al terreno debajo de la estructura. El modelo propuesto presentado en este trabajo de investigación para el diseño de zapatas combinadas de correa en esquina asume que el suelo de soporte es elástico y las zapatas son perfectamente rígidas, que cumplen con las ecuaciones de la flexión biaxial, es decir, la presión del suelo sobre la zapata varia linealmente.

El modelo propuesto presentado en este trabajo para diseño estructural de zapatas combinadas de correa en esquina bajo una carga concéntrica y dos momentos ortogonales en cada columna, también se puede usar para los otros casos: 1) Zapatas bajo una carga concéntrica en cada columna, es decir, todos los momentos son cero; 2) Zapatas bajo una carga concéntrica y un momento en una dirección en cada columna, es decir, los momentos alrededor del eje X o Y son cero.

Las ventajas de las zapatas combinadas de correa son:

1. Ayuda a distribuir la carga de manera más uniforme y transfiere el momento a la zapata adyacente.
2. El empleo de zapatas combinadas de correa puede ser usadas cuando el espacio entre columnas es largo y la zapata combinada regular como la rectangular, trapezoidal o en forma de T no es práctica debido a las excavaciones a gran escala necesarias.

Las principales conclusiones son:

1. El espesor para las zapatas combinadas de correa en esquina es gobernado por el cortante por penetración, y las zapatas combinadas de esquina se rigen por el cortante por flexión.
2. El modelo propuesto para obtener el diseño de zapatas combinadas de correa en esquina se puede usar para el diseño de zapatas combinadas de esquina propuesto por Luévanos-Rojas *et al.* [23].
3. El modelo propuesto se puede utilizar para obtener el diseño de zapatas combinadas de correa en esquina para dos, tres y cuatro líneas de propiedad de lados restringidos (ver Tablas 1).
4. El modelo propuesto se apega más a las condiciones reales con respecto al diseño tradicional, porque el modelo propuesto considera la presión lineal del suelo y el diseño tradicional toma en cuenta una presión uniforme y esta es la máxima en toda la superficie.
5. El modelo propuesto no se limita, mientras que el diseño tradicional toma en cuenta que la fuerza resultante de todas las cargas y

momentos aplicados coincide con la posición del centro geométrico de la zapata, es decir, la presión es uniforme (presión máxima) en todos los puntos de contacto.

Por lo tanto, el modelo propuesto en este trabajo de investigación para obtener el diseño de zapatas combinadas de correa en esquina se puede aplicar a zapatas combinadas de esquina, simplemente considerando los anchos en dirección X de las zapatas 1 y 3 iguales, y los anchos en dirección Y de las zapatas 1 y 2 iguales.

Las sugerencias para los siguientes trabajos de investigación pueden ser:

1. Diseño de unas zapatas combinadas de correa en esquina apoyadas sobre suelos totalmente arcillosos (suelos cohesivos) o suelos totalmente arenosos (suelos granulares), es decir, el diagrama de presión del suelo sobre la zapata es parabólico.
2. Diseño de una cimentación completa para una edificación usando zapatas combinadas de correa.
3. Diseño de una cimentación completa para una edificación usando una losa de cimentación.

Agradecimientos

La investigación descrita en este trabajo fue financiada por el Instituto de Investigaciones Multidisciplinarias de la Facultad de Contaduría y Administración de la Universidad Autónoma de Coahuila. Los autores también agradecen a los revisores y al editor por los comentarios y sugerencias para mejorar la presentación. El estudiante de doctorado Marina Lourdes Garcia Graciano (CVU: 567040) agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico.

Referencias

1. **Amornfa, K., Phienwej, N., Kitpayuck, P. (2012).** Current practice on foundation design of high-rise buildings in Bangkok, Thailand. Lowland Technology International, Vol. 14, No. 2, pp. 70–83.
2. **Luévanos-Rojas, A., Faudoa-Herrera, J. G., Andrade-Vallejo, R. A., Cano-Alvarez M. A. (2013).** Design of isolated footings of rectangular form using a new model. International Journal of Innovative Computing, Information and Control, Vol. 9, No. 10, pp. 4001–4022.
3. **Luévanos-Rojas, A. (2014).** Design of isolated footings of circular form using a new model. Structural Engineering and Mechanics, Vol. 52, No. 4, pp. 767–786. DOI: 10.12989/sem.2014.52.4.767.
4. **Luévanos-Rojas, A. (2016).** A comparative study for the design of rectangular and circular isolated footings using new models. Dyna-Colombia, Vol. 83, No. 196, No. 149–158. DOI: 10.15446/dyna. v83n196.51056.
5. **Abdrabbo, F., Mahmoud, Z. I., Ebrahim, M. (2016).** Structural design of isolated column footings. Alexandria Engineering Journal, Vol. 55, No. 3, pp. 2665–2678. DOI: 10.1016/j.aej.2016.06.016.
6. **Kammari, S., Venkatarathnam, G. (2016).** Analysis and design of isolated and combined footing. International Journal & Magazine of Engineering, Technology, Management and Research, Vol. 3, No. 5, pp. 540–546.
7. **Balachandar, S., Narendra Prasad, D. (2017).** Analysis and design of various types of isolated footings. International Journal of Innovative Research in Science, Engineering and Technology, Vol. 6, No. 3, pp. 3980–3986.
8. **El-kady, M. S., Badrawi, E. F. (2017).** Performance of isolated and folded footings. Journal of Computational Design and Engineering, Vol. 4, pp. 150–157. DOI: 10.10 16/j.jcde.2016.09.001.
9. **López-Chavarría, S., Luévanos-Rojas, A., Medina-Elizondo, M. (2017).** A new mathematical model for design of square isolated footings for general case. International Journal of Innovative Computing, Information and Control, Vol. 13, No. 4 pp. 1149–1168.
10. **Magade, S. B., Ingle, R. K. (2019).** Numerical method for analysis and design of isolated square footing under concentric loading. International Journal of Advanced Structural

Engineering, Vol. 11, pp. 9–20. DOI: 10.1007/s40091-018-0211-3.

11. **Tiwari, R. K., Dhapekar, N. K. (2020).** Design and analysis of isolated footings using RCF Software. International Research Journal of Engineering and Technology (IRJET), Vol. 7, No. 8, pp. 1349–1352.
12. **Rawat, S., Mittal, R. K., Muthukumar, G. (2020).** Isolated rectangular footings under biaxial bending: a critical appraisal and simplified analysis methodology. Practice Periodical on Structural Design and Construction, Vol. 25, No. 3. DOI: 10.1061/(ASCE)SC.1943-5576.0000471.
13. **Al-Ansari, M. S., Afzal, M. S. (2021).** Structural analysis and design of irregular shaped footings subjected to eccentric loading. Engineering Reports, Vol. 3, No. 1. pp. e12283. DOI:10.1002/eng2.12283.
14. **Kramrisch, F., Rogers, P. (1961).** Simplified design of combined footings. Journal of the Soil Mechanics and Foundations Division, Vol. 87, No. 5.
15. **Luévanos-Rojas, A. (2014).** Design of boundary combined footings of rectangular shape using a new model. Dyna-Colombia, Vol. 81, No. 188, pp. 199–208. DOI: 10.15446/dyna.v81n188.41800.
16. **Luévanos-Rojas, A. (2015).** Design of boundary combined footings of trapezoidal form using a new model. Structural Engineering and Mechanics, Vol. 56, No. 5, pp. 745–765. DOI: 10.12989/sem. 2015.56.5.745.
17. **Luévanos-Rojas, A. (2016).** A new model for the design of rectangular combined boundary footings with two restricted opposite sides. Revista ALCONPAT, Vol. 6, No. 2, pp. 172–187. DOI: 10.21041/ra.v6i2.137.
18. **Luévanos-Rojas, A., Barquero-Cabrero, J. D., López-Chavarría, S., Medina-Elizondo, M. (2017).** A comparative study for design of boundary combined footings of trapezoidal and rectangular forms using new models. Coupled Systems Mechanics, Vol. 6, No. 4, pp. 417–437. DOI: 10.12989/csm.2017.6.4.417.
19. **Maheshwari, P. (2017).** Analysis of combined footings on extensible geosynthetic-stone column improved ground. Journal of Civil Engineering, Science and Technology, Vol. 8, No. 2, pp. 57–71. DOI: 10.33736/jcest. 439.2017.
20. **Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M. (2018).** A new model for T-shaped combined footings Part II: Mathematical model for design. Geomechanics and Engineering, Vol. 14, No. 1, pp. 61–69. DOI: 10.12989/gae.2018 .14.1.061.
21. **Yáñez-Palafox, J. A., Luévanos-Rojas, A., López-Chavarría, S., Medina-Elizondo, M. (2019).** Modeling for the strap combined footings Part II: Mathematical model for design. Steel and Composite Structures, Vol. 30, No. 2, pp. 109–121. DOI: 10.12989/scs. 2019.30.2.109.
22. **Pandey, S., Rai, A. (2020).** Comparative study of different types of combined footing for small residential building. Journal of Civil Engineering and Environmental Technology, Vol. 7, No. 2, pp. 221– 225.
23. **Luévanos Rojas, A., López Chavarría, S., Medina Elizondo, M., Sandoval Rivas, R., Farías Montemayor, O. M. (2020).** Un modelo analítico para el diseño de zapatas combinadas de esquina. Revista ALCONPAT, Vol. 10, No. 3, pp. 317–335. DOI: 10.21041/ra.v10i3.432.
24. **Naik, B., Nighojkar, S., Pendharkar, U. (2020).** Design of skirted rectangular combined footing with vertical skirt all around the four edges. International Journal of Creative Research Thoughts (IJCRT), Vol. 8, No. 10, pp. 674–680.
25. **American Concrete Institute (ACI). (2019).** Building code requirements for structural concrete and commentary. New York, U.S.A.

Article received on 17/11/2021; accepted on 25/07/2022. Corresponding author is Arnulfo Luévanos Rojas.

ISSN 2007-9737

Equalization Control of Nonlinear Systems by Discrete Models of the Volterra Operators

Juan Alejandro Vazquez Feijoo[1], Sarahí Morales Pérez[2], Rodrigo Arturo Marquet Rivera[1],
José Navarro Antonio[2], Guillermo Urriolagoitia Sosa[1], Beatriz Romero Angeles[1]

[1] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica,
Mexico

[2] Instituto Politécnico Nacional,
Centro Interdisciplinario de Investigación
para el Desarrollo Integral Regional-Oaxaca,
Mexico

javazquezfeijoo@yahoo.com.mx,
{smorales, rmarquetr, jnavarroa, gurriolagoitias, bromeroa}@ipn.mx

Abstract. The Associated linear equations (ALEs) are parametric models of the Volterra operators. With them, a Volterra inverse is constructed to be used as open loop control of continuous nonlinear systems. However, most of the actual control systems are of discrete nature, this work introduces the novel discrete version of the ALEs. This discrete version is a series of ARX models of the Volterra operators for both the direct and the inverse series. These discrete models of the ALEs are used for an equalization strategy to create an open loop control on a reported simulated Duffing Oscillator and a physical Duffing system constructed by analog circuits.

Keywords. Associated linear equations, Volterra inverse, ARX models, duffing oscillator, nonlinear systems, control systems.

1 Symbology

The following notation is used in the paper:

ω – Frequency,
H(ω) – FRF system output,
H – System operator,
K – Inverse Volterra operator,
W – Overall equalization system operator,
ω_n – Natural frequency,
D – System input amplification,
t – Time,
y(t) – The output signal,
y_j(t) – Output signal from the *j-th* Volterra operator of the system,
x(t) – Input signal,
u(t) – System output acceleration,
U(t) – The FRF of the output acceleration,
z(t) – Output signal from the j-th inverse Volterra operator of the system,
w(t) – Output signal from the j-th Volterra operator of the equalization system,
i – square root of minus one,
r_j, p_j, q_j, $u_{j1,\ j2,\ j3}$.... – ARX lags in time for y, x, errors, and powers of the output,
k_j, c_j, g_j, A_j – Modal parameters.

2 Introduction

In previous work [1], an open-loop control was proposed for time-continuous nonlinear systems. This open-loop control is carried on by a Volterra inverse [2], whose operators are parametric models named Associated Linear Equations (ALE) [3]. They are the counterpart o de FRFs in the frequency domain as can be seen in [4, 5].

The principal characteristic of this method is that the whole system is transformed into an element of unitary gain and therefore losses are compensated, nonlinearities are eliminated and at least in theory, the array has infinite bandwidth.

doi: 10.13053/CyS-26-4-3780

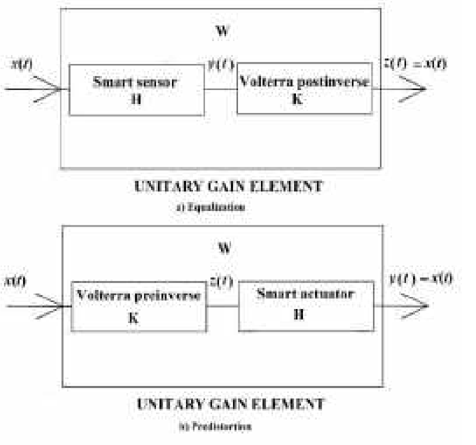


Fig. 1. Volterra inverse strategies for open-loop control

The configuration for sensor systems (equalization strategy) can be seen in Figure 1. The Volterra inverse is connected in tandem with the sensor system.

To obtain a parametric model of a system is in the majority of practical cases very difficult if not impossible. Without a parametric model, it is not possible to obtain the ALEs.

A worthwhile option is the use of discrete models of the system such as the Nonlinear Autoregressive Moving Average with eXogenous inputs (NARMAX). A good introduction to this kind of model can be found in [6] and [7].

Some work on this kind of model and the Volterra series can be found in [6], where these models were applied to structural integrity. Recent applications have been done on ground vehicle control as in [8], though the present work offers a more explicit and simple means of control.

Other uses that can be mentioned are the simulation for system analysis with the aid of other methods of identification such as neural networks [9]. Several works focus on improving the model quality, including accounting for small changes in the system, e.g. [10, 11].

The main objective of this work is to present a novel construction of the inverse Volterra by discrete versions (AutoRegresive with eXogenous inputs ARX models) of the ALEs and to implement the control in a practical case.

The open-loop control is going to be implemented on an analogic Duffing oscillator. Some recent works on this kind of system can be found in [12 & 13]. The Duffing oscillator is considered a sensor system and therefore an equalization strategy (Figure 1) is implemented.

A Duffing Oscillator is a very appropriate system for this kind of control as the number of Volterra operators is finite [14]. In practice, a traditional example of this kind of system are those structures that present softening or hardening stiffness, this kind of structures can be found for example in plane wings or helicopter propellers [15].

The work is organized into five sections. The first section is a breve description of the general theory of the ALEs and the inverse Volterra. The second section presents the novelty of this work, the discrete version of the ALEs for both direct and inverse systems. In the third section, the analogic Duffing oscillator is described. In the fourth section, open-loop control is implemented. The last section contains the conclusions.

3 Materials and Methods

3.1 Associated Linear Equations (ALEs)

In [3], it is explained that the most general second-order equation for which the Associated Linear Equations (ALEs) can be applied is:

$$\ddot{y}(t) + \dot{y}(t) + y(t) + \sum_{j=1}^{N} k_j y(t)^j + \sum_{j=1}^{P} c_j x(t)^j + \sum_{s=1}^{Q} g_s(t), \tag{1}$$

$y(t)$ – system output,
k_j and c_j – constant coefficients,
N, P – the maximum order of polynomial terms depending only on the input or output,
Q – the maximum number of $g_s(t)$´s functions,
$g_s(t)$ – some appropriate non-continuous function such as absolute value.

Equation (1) includes two kinds of Volterra systems, the Duffing oscillator, and the Hammerstein. The Volterra series [2] decomposes the signal in an infinite series of operators:

$$y(t)=\sum_{i=1}^{\infty} y_n(t)\cdot \tag{2}$$

Each operator corresponds to each harmonic output signal order. Where each operator is obtained from the Associated Linear Equations (ALEs). A Duffing oscillator is when in equation (1) one has *P*=1 and *Q*=0, e.g., a third-degree Duffing oscillator of is:

$$\ddot{y}(t)+\dot{y}(t)+y(t)+k_3y(t)^3=c_1x(t), \tag{3}$$

where N=3There is no intention to explain in detail how to obtain the ALEs from a parametric model, a detailed development is done in reference [2]. Here, it is only mentioned that those equations are obtained from a variation of the perturbation method, e.g., see [14]. In the same reference [2], it is demonstrated that all even-order Volterra operators of a third-order Duffing oscillator as that represented by equation (3) are null. The lower order non-null ALEs are:

$$\ddot{y_1}(t)+\dot{y_1}(t)+y_1(t)=c_1x(t)\ , \tag{4a}$$

$$\ddot{y_3}(t)+\dot{y_3}(t)+y_3(t)=k_3y_1^3(t), \tag{4b}$$

$$\ddot{y_5}(t)+\dot{y_5}(t)+y_5(t)=3k_3y_3(t)y_1^2(t), \tag{4c}$$

$$\begin{aligned}\ddot{y_7}(t)+\dot{y_7}(t)+y_7(t)&\\ &=3k_3y_1(t)y_3^2(t)\\ &+3k_3y_5(t)y_1^2(t).\end{aligned} \tag{4d}$$

Each equation in (4) represents a linear system that depends only on the nth-harmonic order of the input (as y_i are a function of the nth-order frequencies of the input), which is obtained from products of lower-order operators.

3.2 Postinverse Volterra Operators by ALEs

The open-loop control that is used in this work [1] consists in connecting in tandem the system with its inverse Volterra (Figure 1). The basic theory can be seen in [15].

Let us represent the system operator as $\mathbf{H}$, $\mathbf{K}$ the inverse Volterra operator, and **W** is the operator of the tandem connection.

Being *x*(*t*), *w*(*t*), and *k*(*t*) the respective outputs in the time domain, $H(\omega)$, $W(\omega)$, and $K(\omega)$ are the Frequency Response Functions (FRFs). From [15], the output from the $\mathbf{W}$ operator is:

$$w(t)=x(t)+\sum_{i=1}^{N} w_j(t)=x(t)\ , \tag{5}$$

where w_j are the composed system [3]. In the most general case, there should be a maximum order *N* with significant contribution and therefore this is the maximum where w_j is the composed system [3]. In the most general case, there should be a maximum order N with significant contribution and therefore this is the maximum order operator to use. In the case of Duffing oscillators, the number N is the actual number of operators. It implies that the inverse Volterra operator of a Duffing oscillator of a third degree possesses only three terms (operators).

From [15], the first-order inverse Volterra operators are defined as:

$$\mathrm{K}_1=\mathrm{H}_1^{-1}. \tag{6}$$

The second and third are respectively:

$$\mathrm{K}_2=-\mathrm{K}_1[\mathrm{H}_2[\mathrm{K}_1]]. \tag{7}$$

According to equation (8a), the second-order inverse operator is also null, as all the even direct operators of the Duffing oscillators of this kind (equation (3)) are null. The third-order Volterra inverse remains as:

$$\mathrm{K}_3=-\mathrm{K}_1[\mathrm{H}_3[\mathrm{K}_1]]. \tag{8}$$

Figure 2 shows the block diagram. The output of the global operator $\mathbf{W}$ is from (8b):

$$w(t)=z_1(y_3(t))-z_1(y_3[z_1(y(t))])=x(t)\cdot \tag{9}$$

3.3 AutoRegresive with EXogenous Inputs (ARXs) Inverse Volterra Models

Now *j, m, s* are the lags with maximums of K, P, Q and *e*(*i*) is the model of additive noise. Our simulated system is taken from [8]. It is a cubic Duffing oscillator is modelled as:

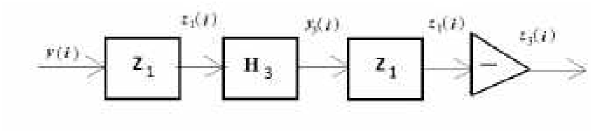


Fig. 2. Block diagram of the third inverse of Volterra

$$y(i)=\sum_{j=1}^{K} r_j y(i-j)+\sum_{m=0}^{P} p_j x(i-m)+\sum_{r=1}^{Q} q_r e(i-s) + \sum_{s1=1}\sum_{s2=1}\sum_{s3=1} u_{s1,s2,s3} y(i-s1)y(i-s2)y(i-s3) \quad (10)$$

$$y(i)=r_1y(i-1)+r_2y(i-2)+p_0x(i)+p_1x(i-1)+p_2x(i-2)+u_{1,1,1}y^3(i-1)+u_{2,2,2}y^3(i-2) +u_{1,1,2}y^2(i-1)y(i-2)+u_{1,2,2}y^2(i-2)y(i-1)+q_1e(i-1)+q_2e(i-2) \quad (11)$$

A discrete Volterra series is:

$$y(i-j)=\sum_{k=1}^{\infty} y_k\,(i-j), \quad (12)$$

substituting this equation into equation (11), we have:

$$\sum_{k=1}^{\infty} y_k\,(i) = r_1 \sum_{k=1}^{\infty} y_k(i-1) + r_2 \sum_{k=1}^{\infty} y_k\,(i-2) + p_0x(i) + p_1x(i-1) + p_2x(i-2) + u_{1,1,1}[\sum_{k=1}^{\infty} y_k(i-1)]^3 + u_{2,2,2}[\sum_{k=1}^{\infty} y_k(i-2)]^3 + u_{1,1,2,}[\sum_{k=1}^{\infty} y_k(i-1)]^2 y_k(i-2) + u_{1,2,2,}y_k(i-1)[\sum_{k=1}^{\infty} y_k(i-2)]^2, \quad (13)$$

where error terms are not included.

The Volterra series possesses a convergence maximum value x_{max}, so the input must comply with the following:

$$x(i) \leq x_{max}.$$

Under this value, it is the guarantee that:

$$y_1(i) \leq y_2(i) \leq y_3(i) \ldots \ldots \leq y_k(i) \ldots \leq y_\infty(i).$$

Therefore, under these circumstances, it is always possible to cut the series up to an order k and then still have the required exactitude.

Taking as an example only the first three nonzero operators of equation (12).

The infinite sum is truncated up to three terms:

$$\left[\sum_{k=1}^{\infty} \mathrm{y_k(i-2)}\right]^3 = [\mathrm{y_1(i-2)+y_3(i-2)+y_5(i-2)}]^3.$$

Developing the cube:

$$\begin{aligned}[y_1(i-2)+y_3(i-2)+y_5(i-2)]^3 &= y_1(i-2)^3 + y_3(i-2)^3 \\ &+ y_5(i-2)^3 \\ &+ 3y_1(i-2)^2y_3(i-2) + \end{aligned} \quad (14)$$

$$\begin{aligned}3y_1(i-2)^2y_5(i-2) + 3y_3(i-2)^2y_5(i-2) \\ + 3y_3(i-2)^2y_1(i-2) \\ + 3y_5(i-2)^2y_1(i-2) + \\ 3y_5(i-2)^2y_3(i-2).\end{aligned}$$

Equation (13) is then:

$$y_1(i)+y_3(i)+y_5(i) = r_1[y_1(i-1)+y_3(i-1)+ y_5(i-1)] + r_2[y_1(i-2)+y_3(i-2)+y_5(i-2)] + p_0x(i)+p_1x(i-1)+p_2x(i-2)+u_{1,1,1}[y_1(i-1)^3+3y_1(i-1)^2y_3(i-1)] + u_{2,2,2}[y_1(i-2)^3 + 3y_1(i-2)^2y_3(i-2)] + u_{1,1,2,}[y_1(i-1)^2y_1(i-2) + y_1(i-1)^2y_3(i-2) + 2y_1(i-1)y_3(i-1)y_1(i-2)] + u_{1,2,2,}[y_1(i-2)^2y_1(i-1) + y_1(i-2)^2y_3(i-1) + 2y_1(i-2)y_3(i-2)y_1(i-1)]. \quad (15)$$

Using Fourier transform, it is possible to express any integrable function as a sum of complex exponentials. When rising this sum of exponential terms to any power generates terms function of harmonic frequencies (harmonic orders), that are the sum of the frequencies of the original terms.

As the equation (15) should remain valid at any time, the right and left sides have to remain the same for each harmonic order. Then the equation (15) can be split into several equations one contains all the harmonics of the same order. Equation (15) is split then in the following equations:

$$\begin{aligned}y_1(i) = r_1y_1(i-1)+r_2y_1(i-2)+p_0x(i) \\ + p_1x(i-1)+p_2x(i-2),\end{aligned} \quad (16a)$$

$$y_3(i) = r_1y_3(i-1)+r_2y_3(i-2)+u_{1,1,1}y_1(i-1)^3 + u_{2,2,2}y_1(i-2)^3 + u_{1,1,2,}y_1(i-1)^2y_1(i-2) + u_{1,2,2,}y_1(i-2)^2y_1(i-1) \qquad (\mathrm{b}.\, y_5(i) = r_1[y_5(i-1)] + r_2[y_5(i-2)] + u_{1,1,1}[3y_1(i-1)^2y_3(i-1)] + u_{2,2,2}[3y_1(i-2)^2y_3(i-2)] + u_{1,1,2,}[y_1(i-1)^2y_3(i-2) + 2y_1(i-1)y_3(i-1)y_1(i-2)] + u_{1,2,2,}[y_1(i-2)^2y_3(i-1) + 2y_1(i-2)y_3(i-2)y_1(i-1)]. \quad (16b)$$

The equations above become the first, third, and fifth discrete (ARX) ALEs.

4 Results and Discussion

4.1 Discrete Volterra Inverse

This section is devoted to obtaining the inverse Volterra. The first inverse operator is easier to obtain in the frequency domain. The z-transform [16] of equation (16a) is:

$$H_1(j) = \frac{p_0 + p_1\varsigma(j)^{-1} + p_2\varsigma(j)^{-2}}{1 + r_1\varsigma(j)^{-1} + r_2\varsigma(j)^{-2}}, \quad (17)$$

where $\varsigma = e^{i\frac{2\pi}{n}j}$ is used to avoid confusion with the inverse Volterra operator output z_i. The output from the first-order operator can be retrieved by the inverse z-transform of the following equation:

$$Y_1(j) = H_1(j)X(j).$$

$H_1(j)$ is then the ratio between the z-transforms of the output and input:

$$H_1(j) = \frac{Y_1(j)}{X(j)}. \quad (18)$$

Considering equation (6), the inverse Volterra operator of the first-order is directly the inverse operator of the first-order Volterra operator of the systems $\mathbf{H}_1$. In the frequency domain, it means that the first-order operators are the multiplicative inverse of each other:

$$K_1(j) = \frac{1}{H_1(j)}. \quad (19)$$

From equations (17) and (19):

$$K(j) = \frac{1 + r_1\varsigma(j)^{-1} + r_2\varsigma(j)^{-2}}{p_0 + p_1\varsigma(j)^{-1} + p_2\varsigma(j)^{-2}}. \quad (20)$$

Substituting equation (18) into (19):

$$K_1(j) = \frac{1}{H_1(j)} = \frac{X(j)}{Y_1(j)}. \quad (21)$$

This means that if the input is the first-order Volterra operator of the system ($y_1(t)$), the output of the inverse Volterra first-order is simply the input $x(t)$. This signal at the same time is the first Volterra operator of the equalization system (system-postinverse) $w_1(t)$ (see Figure 1). It means that:

$$w_1(t) = z_1(t) = x(t).$$

Equation (21) can be also expressed as:

$$K_1(j) = \frac{Z_1(j)}{Y_1(j)}. \quad (22)$$

For completeness in the pre-inverse case (distortion strategy), the input into the equation (22) is $X(j)$, then the equivalent to equation of (22) is:

$$K_1(j) = \frac{Z_1(j)}{X(j)}. \quad (23)$$

Now, equation (20) into equation (23) one obtains:

$$\frac{Z_1(j)}{X(j)} = \frac{1 + r_1\varsigma(j)^{-1} + r_2\varsigma(j)^{-2}}{p_0 + p_1\varsigma(j)^{-1} + p_2\varsigma(j)^{-2}}. \quad (24)$$

The z-inverse transform produces the first-order Volterra postinverse:

$$z_1(j) = -\frac{p_1}{p_0}z_1(j-1) - \frac{p_2}{p_0}z_1(j-2) + \frac{1}{p_0}\big(y_1(j) - r_1y_1(j-1) - r_2y_1(j-2)\big). \quad (25)$$

Something similar is obtained for the pre-inverse case, when the inverse transform on equation (23) gives:

$$z_1(j) = -\frac{p_1}{p_0}z_1(j-1) - \frac{p_2}{p_0}z_1(j-2) + \frac{1}{p_0}\big(x(j) - r_1x(j-1) - r_2x(j-2)\big). \quad (26)$$

About equation (25), in a real system, there are not such signals as $y_1(j)$ or $y_2(j)$, nor any other operator. Rather the system produces only the signal $y(j)$ which is the sum of all these Volterra operators (equation 12). The real input into the first post-inverse operator is $y(j)$. Then equation (25) has to be rewritten as:

$$z_1(j) = -\frac{p_1}{p_0}z_1(j-1) - \frac{p_2}{p_0}z_1(j-2) + \frac{1}{p_0}\big(y(j) - r_1y(j)(j-1) - r_2y(j)(j-2)\big). \quad (27)$$

Lets express equation (11) as:

$$y(i) - r_1y(i-1) - r_2y(i-2) = p_0x(i) + p_1x(i-1) + p_2x(i-2) + u_{1,1,1}y^3(i-1) + u_{2,2,2}y^3(i-2) + u_{1,1,2}y^2(i-1)y(i-2) + u_{1,2,2}y^2(i-2)y(i-1) + q_1e(i-1) + q_2e(i-2) \quad (28)$$

After substitution in equation (27), one has (disregarding e terms):

$$z_1(j) + \frac{p_1}{p_0}z_1(j-1) + \frac{p_2}{p_0}z_1(j-2) = x\big(j\big) + \frac{p_1}{p_0}x(i-1) \ + \frac{p_2}{p_0}x(i-2) \ + \frac{1}{p_0}(u_{1,1,1}y(i-1)^3 + \frac{1}{p_0}u_{2,2,2}y(i-2)^3 + \frac{1}{p_0}u_{1,1,2}y(i-1)^2y(i-2) + \frac{1}{p_0}u_{1,2,2}y(i-1)y(i-2)^2. \quad (29)$$

Observe that in the particular case, in which the input signal is small enough so that only the first-order is significant, equation (25) is simply:

$$z_1(j) + + \frac{p_1}{p_0}z_1(j-1) + \frac{p_2}{p_0}z_1(j-2) = x\big(j\big) + \frac{p_1}{p_0}x(i-1) \ + \frac{p_2}{p_0}x(i-2) \ . \quad (30)$$

Therefore, $z_1(j)=x(j)$.

For the pre-inverse case, the input into the system, i.e., the input into the operator **H** is $z_1(j)$ obtained from equation (26). Theoretically, $\mathbf{H}$ is the sum of all Volterra operators $\mathbf{H}_i$. Equation (11) is now:

$$y(j) - r_1 y(j-1) - r_2 y(j-2) = p_0 z_1(j) + p_1 z_1(j-1) + p_2 z_1(j-2) + (u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2} y(i-1)^2 y(i-2) + u_{1,2,2} y(i-1) y(i-2)^2. \tag{31}$$

Using equation (26) into (31):

$$y(j) - r_1 y(j-1) - r_2 y(j-2) = \big(x(j) - r_1 x(j-1) - r_2 x(j-2)\big) + (u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2} y(i-1)^2 y(i-2) + u_{1,2,2} y(i-1) y(i-2)^2. \tag{32}$$

Again, for a small enough signal, higher orders can be neglected, therefore:

$$y(j) - r_1 y(j-1) - r_2 y(j-2) = x(j) - r_1 x(j-1) - r_2 x(j-2).$$

As expected $z_1(j)=x(j)$. When the signal is not small, equation (32) shows that the output signal $y(j)$ contains higher-order components.

The second-order operator from reference [15] shall be:

$$\mathrm{K}_2 = -\mathrm{K}_1\left[\mathrm{H}_2\left[\mathrm{K}_1[\mathrm{H}]\right]\right]. \tag{32}$$

However, because $\mathbf{H}_2$ is zero, there is no second-order inverse operator.

The third-order inverse operator $\mathbf{K}_3$ is:

$$\mathrm{K}_3 = -\mathrm{K}_1\left[\mathrm{H}_3\left[\mathrm{K}_1[\mathrm{H}]\right]\right]. \tag{33}$$

The input into the third-order operator for the post-inverse case is $\mathbf{H}$ represented by equation (11). The first part of the third-order inverse operator according to equation (33) is $[\mathbf{K}_1[\mathbf{H}]]$, which results is the equation (29).

Following equation (33), the signal goes through the direct operator $\mathbf{H}_3$, represented by equation (16b). Then, from equation (16a), the third-order input in $\mathbf{H}_3$ depends on powers of $y_1(i)$, then using equation (16a):

$$y_1(i) = r_1 y_1(i-1) + r_2 y_1(i-2) + p_0 z_1(i) + p_1 z_1(i-1) + p_2 z_1(i-2). \tag{34}$$

Again remember that the input into $\mathbf{H}_3$ is not $x(i)$ but $z_1(i)$ the substitution of equation (29) into (34):

$$y(i) - r_1 y(i-1) + r_2 y(i-2) = p_0 \mathrm{x(j)} + p_1 x(j-1) + p_2 x(j-2) + (u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2} y(i-1)^2 y(i-2) + u_{1,2,2} y(i-1) y(i-2)^2.$$

That is the same equation as (11). Equation (16b) in the third-order inverse operator is simply:

$$y_3(i) = r_1 y_3(i-1) + r_2 y_3(i-2) + u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2,} y(i-1)^2 y(i-2) + u_{1,2,2,} y(i-2)^2 y(i-1). \tag{35}$$

Up to this point it has been solved up to $\mathbf{H}_3[\mathbf{K}_1[\mathbf{H}]]$, now the signal should go through the operator$-\mathbf{K}_1$ (see equation (33)). Then by the use of equation (25):

$$z_1(j) = -\tfrac{p_1}{p_0} z_1(j-1) - \tfrac{p_2}{p_0} z_1(j-2) + \tfrac{1}{p_0}\big(y_3(j) - r_1 y_3(j-1) - r_2 y_3(j-2)\big). \tag{36}$$

Using equation (35) one obtains:

$$z_1(j) + \tfrac{p_1}{p_0} z_1(j-1) + \tfrac{p_2}{p_0} z_1(j-2) = \tfrac{1}{p_0}\Big(u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2,} y(i-1)^2 y(i-2) + u_{1,2,2,} y(i-2)^2 y(i-1)\Big). \tag{37}$$

Finally, from (30) $z_3(j)$ =- $z_1(j)$ therefore:

$$z_3(j) + \tfrac{p_1}{p_0} z_3(j-1) + \tfrac{p_2}{p_0} z_3(j-2) = -\tfrac{1}{p_0}\Big(u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2,} y(i-1)^2 y(i-2) + u_{1,2,2,} y(i-2)^2 y(i-1)\Big). \tag{38}$$

Adding equations (29) and (38) one obtains:

$$\big(z_1(j) + z_3(j)\big) + \tfrac{p_1}{p_0}\big(z_1(j) + z_3(j)\big) + \tfrac{p_2}{p_0}\big(z_1(j) + z_3(j)\big) = \mathrm{x(j)} + \tfrac{p_1}{p_0} x(i-1) \quad + \tfrac{p_2}{p_0} x(i-2) \quad + \tfrac{1}{p_0}(u_{1,1,1} y(i-1)^3 + \tfrac{1}{p_0} u_{2,2,2} y(i-2)^3 + \tfrac{1}{p_0} u_{1,1,2} y(i-1)^2 y(i-2) + \tfrac{1}{p_0} u_{1,2,2} y(i-1) y(i-2)^2 - \tfrac{1}{p_0}\Big(u_{1,1,1} y(i-1)^3 + u_{2,2,2} y(i-2)^3 + u_{1,1,2,} y(i-1)^2 y(i-2) + u_{1,2,2,} y(i-2)^2 y(i-1)\Big). \tag{39}$$

Then:

$$z_1(j) + z_3(j)\big) + \tfrac{p_1}{p_0}\big(z_1(j) + z_3(j)\big) + \tfrac{p_2}{p_0}\big(z_1(j) + z_3(j)\big) = \mathrm{x(j)} + \tfrac{p_1}{p_0} x(i-1) \quad + \tfrac{p_2}{p_0} x(i-2) \quad . \tag{40}$$

That according with equation (9) gives the expected result:

$$z(j) = z_1(j) + z_3(j) = x(j).$$

Observe that only two inverse operators suffice for obtaining as system exit $x(j)$, no matter the input magnitude. It remains true meanwhile the physical elements of the system remain acting as a Duffing oscillator.

4.2 Inverse Series Convergence

Sometimes the inverse of Volterra presents instability. This section tries to find a suitable criterion to avoid this problem. Let's consider the following example of a simple linear ARX model:

$$y(i)=r_1y(i-1)+r_2y(i-2)+p_0x(i)+p_1x(i-1). \qquad (41)$$

Since it is linear, only the first-order Volterra operator is necessary. The inverse operator is:

$$z_1(i)=\frac{1}{p_0}y(i)-\frac{r_1}{p_0}y(i-1)-\frac{r_2}{p_0}y(i-2)-\frac{p_1}{p_0}z_1(i-1). \qquad (42)$$

Considering the initial condition for the inverse $z_1(1)=0$ one has:

$$z_1(2)=\frac{1}{p_0}y_1(2)-\frac{r_1}{p_0}y_1(1). \qquad (43)$$

where $y_1(1)$ and $y_1(2)$ are system initial conditions.

For i=3 one has:

$$z_1(3)=-\frac{p_1}{p_0}z_1(2)+\frac{1}{p_0}\left(y_1(3)-r_1y_1(2)-r_2y_1(1)\right). \qquad (44)$$

Substitution of equation (43) into equation (44):

$$z_1(3)=\left(\frac{p_1r_1}{p_0^2}-\frac{r_2}{p_0}\right)y_1(1)-\left(\frac{p_1}{p_0^2}+\frac{r_1}{p_0}\right)y_1(2)+\frac{1}{p_0}y_1(3). \qquad (45)$$

The same procedure for the following two instants one has:

$$z_1(4)=\left(-\frac{p_1^2r_1}{p_0^3}+\frac{p_1r_2}{p_0^2}\right)y_1(1)+\left(\frac{p_1^2}{p_0^3}+\frac{p_1r_1}{p_0^2}-\frac{r_2}{p_0}\right)y_1(2)-\left(\frac{p_1}{p_0^2}+\frac{r_1}{p_0}\right)y_1(3)+\frac{1}{p_0}y_1(4)$$

And

$$z_1(5)=\left(\frac{p_1^3r_1}{p_0^4}-\frac{p_1^2r_2}{p_0^3}\right)y_1(1)-\left(\frac{p_1^3}{p_0^4}+\frac{p_1^2r_1}{p_0^3}-\frac{p_1r_2}{p_0^2}\right)y_1(2)+\left(\frac{p_1^2}{p_0^3}+\frac{p_1r_1}{p_0^2}-\frac{r_2}{p_0}\right)y_1(3)-\left(\frac{p_1}{p_0^2}+\frac{r_1}{p_0}\right)y_1(4)+\frac{1}{p_0}y_1(5). \qquad (46)$$

Generalizing for any instant *i*:

$$z_1(i)=\frac{1}{p_0}y_1(i)-\left(\frac{p_1}{p_0^2}+\frac{r_1}{p_0}\right)y_1(i-1)+\sum_{j=2}^{i-1}(-1)^j\left(\frac{p_1^j}{p_0^{j+1}}+\frac{p_1^{j-1}r_1}{p_0^j}-\frac{p_1^{j-2}r_2}{p_0^{j-1}}\right)y_1(i-j). \qquad (47)$$

From equation (47) it can be observed that the inverse operator $z_1(i)$ is the function of powers of the ratio $\frac{p_1}{p_0}$. The higher *i* the higher the power. Therefore, to guarantee convergence it is necessary to have:

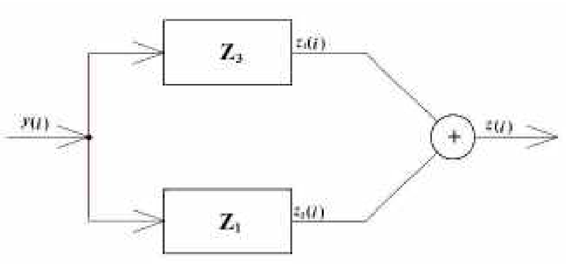


Fig. 3. Signal path through the Volterra postinverse

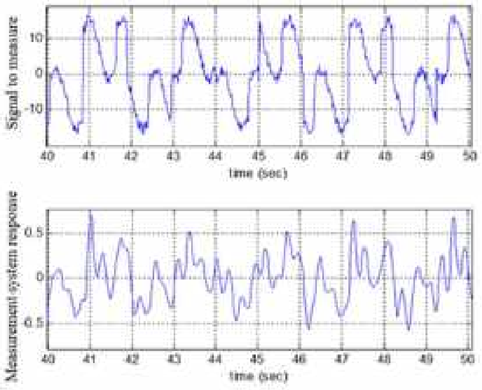


Fig. 4. Signal measured without the postinverse Volterra

$$\frac{p_1}{p_0}<1$$

In general, if there are more input lags, the criterion should be:

$$\frac{p_2}{p_0}<1,\ldots\ldots,\ \frac{p_j}{p_0}<1,\ldots\ldots \text{ and } \frac{p_n}{p_0}<1. \qquad (35)$$

4.3 Duffing Oscillator Model

In [8], Duffing oscillator model was identified and the following equation is obtained:

$$\begin{aligned}y(i)&=1.7639y(i-1)-0.9048y(i-2)+4.034\times10^{-4}x(i)\\&+3.84\times10^{-4}x(i-1)+3.8688\times10^{-4}x(i-2)\\&-0.1998y^3(i-1)+2.15\times10^{-2}y^3(i-2)\\&+0.2089y^2(i-1)y(i-2)-0.1564y^2(i-2)y(i-1)\\&+3.28\times10^{-3}e(i-1)-2.38\times10^{-3}{}_2e(i-2)\end{aligned} \qquad (49)$$

As can be seen, the coefficients of the input lags (p_i) keep the convergence criterion for the Volterra inverse. Let´s consider that this Duffing oscillator represents the dynamics of a sensor system.

The Duffing oscillator does not possess a second power term (equation (40)). It implies that it has only even-order non-zero operators. As it was done in section 2.3, the first three non-zero operators are obtained:

$$\begin{aligned} y_1(i) &= 1.7639 y_1(i-1) - 0.9048 y_1(i-2) \\ &+ 4.034\times10^{-4} x(i) + 3.84\times10^{-4} x(i-1) \\ &+ 3.8688\times10^{-4} x(i-2) \qquad \text{a)} \end{aligned}$$

$$\begin{aligned} y_3(i) &= 1.7639 y_3(i-1) - 0.9048 y_3(i-2) \\ &- 0.1998 y_1^3(i-1) + 2.15\times10^{-2} y_1^3(i-2) \\ &+ 0.2089 y_1^2(i-1) y_1(i-2) \\ &- 0.1564 y_1^2(i-2) y_1(i-1) \qquad \text{b)} \quad (50) \end{aligned}$$

$$\begin{aligned} y_5(i) &= 1.7639 y_5(i-1) - 0.9048 y_5(i-2) \\ &- 0.5994 y_1^2(i-1) y_3(i-1) \\ &+ 6.45\times10^{-2} y_1^2(i-2) y_3(i-2) \\ &+ 0.2089 y_1^2(i-1) y_3(i-2) \\ &+ 0.4178 y_1(i-1) y_3(i-1) y_1(i-2) \\ &- 0.1564 y_1^2(i-2) y_3(i-1) \\ &- 0.3128 y_1(i-2) y_3(i-2) y_1^2(i-1) \qquad \text{c)} \end{aligned}$$

Equations (50) are obtained in the same way as equations (16). The input signal into the Duffing oscillator system is a random signal with an rms of 13.55 units. The system is simulated, and in Figure 4, the comparison between the input and the output signal is shown.

Because of the system inertia, nonlinearity, and noise, the difference between both signals is quite noticeable, i.e., the measurement signal differs greatly from the actual signal to be measured.

Now, let us implement the equalization strategy for controlling the measuring system (the Duffing oscillator equation (50)).

By the use of equation (16) and by the same process done in section 3.1, the first inverse of the Volterra operator is:

$$\begin{aligned} z_1(t) &= 2.4789\times10^3 y(t) \\ &- 4.3726\times10^3 y(t-1) \\ &+ 2.2429\times10^3 y(t-2) \\ &- 0.9519\times10^{-4} z_1(t-1) \\ &- .9142 z_1(t-2) \end{aligned} \qquad (51)$$

By the use of equations (50) and (51) on equation (33) the signal $z_3(i)$ is obtained (see section 3.1), The Volterra inverse $z\ (i)$ is obtained by adding $z_1(i)$ and $z_3(i)$, see Figure 3.

Figure 5 is again compared to the input signal, but this time with the output produced by the equalization control. It is now very difficult to see visually the difference between the input signals $x(i)$ and the output signal $w(i)= z(i)$, i.e., the signal has been measured quite much more accurately. The mse is 12.92% despite the noise present in the signal.

The Frequency Response Function (FRF) of the first order of the Duffing oscillator is shown in Figure 6.

The semilogarithmic graph shows a bandwidth of 30rad/sec. It implies that if the input signal is of a frequency of 200rad/sec the Duffing oscillator is not going to be capable to measure such a signal. This circumstance can be verified in Figure 7, where a signal of a frequency of 200 rad/sec is measured. The Duffing oscillator response is indistinguishable in the graphic.

Now, if the signal is measured with the equalization strategy, the result is shown in Figure 8. What is now indistinguishable is the difference between the signal to be measured x(i) and the measured signal *w(i)*. The mse is only 0.192%.

4.4 Analog Duffing Oscillator System

An analog electronic system is used in this work. It is designed as a third-degree Duffing oscillator, see equation (3). This equation can be rewritten in an integral form as in [9]:

$$\begin{aligned} u(t) + 2\zeta\omega_n \int u(t)dt + \omega_n^2 \int\int u(t)dt \\ + \omega_e^2 \left(\int\int u(t)dt\right)^3 = Ax(t) \end{aligned} \qquad (52)$$

The design of integral blocks is reported to have more stability than a design with derivative elements. Figure 9 shows a schematic of the system. The Fourier transform of equation (30) for the first-order is:

$$U(\omega) = AX(\omega) - 2\zeta\omega_n \frac{U(\omega)}{i\omega} - \omega_n^2 \frac{U(\omega)^3}{(i\omega)^2}\cdot \qquad (53)$$

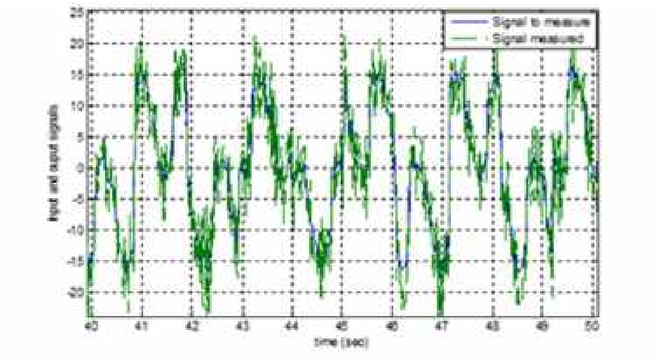


Fig. 5. Output of the equalization system (Duffing oscillator-Post-inverse Volterra) compared with the system input (signal to measure)

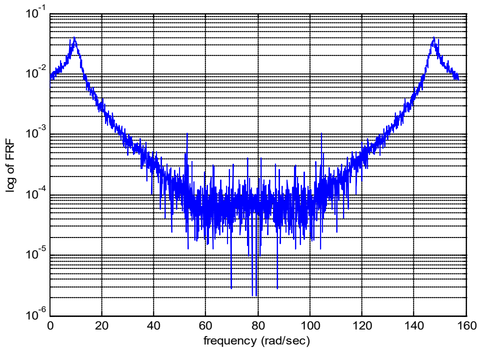


Fig. 6. First-order FRF of the Duffing Oscillator

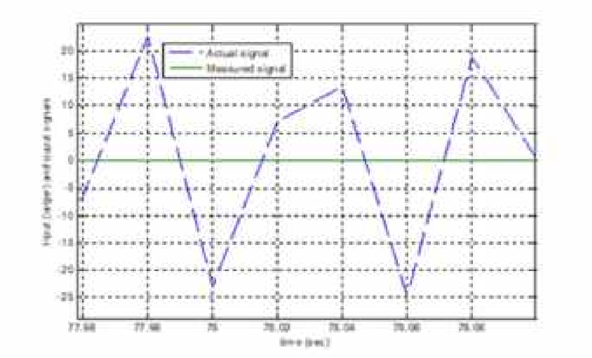


Fig. 7. For very high frequencies, the Duffing oscillator used as a measurement system has very low response

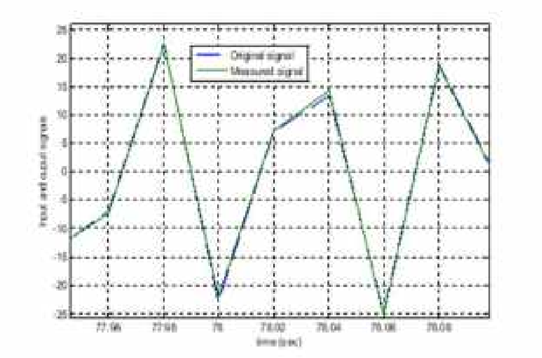


Fig. 8. The measurement Duffing system with post-inverse for a very high frequency

The integration blocks are constructed traditionally by Op-amps (Figure 10). The cubic operator is done by connecting in cascade three multiplicative circuits AD633JN/AN. The electronic design can be seen in Figure 11.

4.4.1 Identification

The input into the system is a three-harmonic signal shown in Figure 12. The system response is shown in Figure 13 The input frequencies are 6, 12, and 15 rad/sec. The output is filtered; this is to eliminate frequencies bigger than 50rad/sec, to prevent noise contamination. The data is taken at an interval of 0.02 sec.

The NARX model proposed for this system is:

$$\begin{aligned} y(i) &= r_1 y(i-1) + r_2 y(i-2) + p_1 x(i) \\ &+ u_{1,1} y^2(i-1) + u_{1,1,1} y^3(i-1) \end{aligned}$$

Applying a NARX system identification procedure [12], the model found is:

$$\begin{aligned} y(i) &= 0.9804\, y(i-1) + 7.321 \times 10^{-6} y(i-2) \\ &+ 1.9 \times 10^{-3} x(i) + 6.269 \times 10^{-4} y^2(i-1) \\ &+ 2.4 \times 10^{-3} y^3(i-1) \end{aligned} \tag{54}$$

Figure 14 compares the simulation output from the NARX model (equation (54)) and the real system response, the mse is 5.143%. The analog system presents certain instabilities both in frequency and in amplitude that implies that any model in equation (45) can be used only up to an input amplitude of 13 V and a frequency of 60rs/sec.

This limitation is not associated with predistortion control, but with the change in the response behavior of the elements of the analog system.

Just as equations (16) were obtained, the first three non-zero ALE's are: Only the first three ALEs are obtained as the inverse Volterra is composed only for the first three order inverse operators.

It is possible to compare the ARX obtained, i.e., the ALE´s with the real corresponding response of the system. The whole signal is supposed to contain all the nth-order signals.

If equation (55a) is subtracted from the output signal, the results are very close to the sum of the harmonic signals of the second harmonic order onwards.

The adequate amplitude of the input signal gives a signal dominated by the second harmonic order signal.

Now, subtracting the equation (55b) from this signal, the signal that results is mainly the third harmonic order.

It is possible then to compare each harmonic signal of the output system with the corresponding ALE (equations (55)) see Figure 14.

Figure 15b shows that the curves differ from each other. This is because the third-order signal is of significant magnitude when compared with the second-order.

However, the remaining signal is quite close to the third-order ALE and therefore the ALEs are correct:

$$\begin{aligned}
&y_1(i)=0.9804y_1(i-1)\\
&+7.321\times10^{-6}y_1(i-2)\\
&+1.9\times10^{-3}x(i)\\[4pt]
&y_2(i)=0.9804y_2(i-1)\\
&+7.321\times10^{-6}y_2(i-2)\\
&+6.269\times10^{-4}y_1^2(i-1)\\[4pt]
&y_3(i)=0.9804y_3(i-1)\\
&+7.321\times10^{-6}y_3(i-2)\\
&+2.4\times10^{-3}y_1^3(i-1)\\
&+1.254\times10^{-3}y_1(i-1)y_2(i-1)
\end{aligned} \tag{55}$$

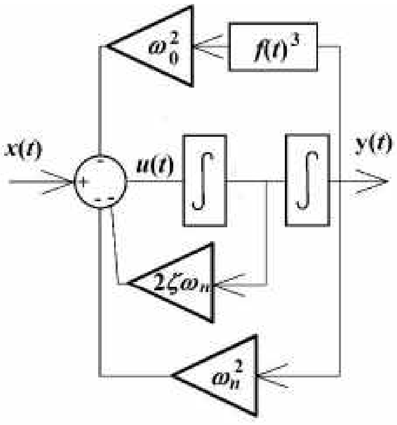


Fig. 9. Block Diagram of the analog amp Duffing oscillator system (equation (52)

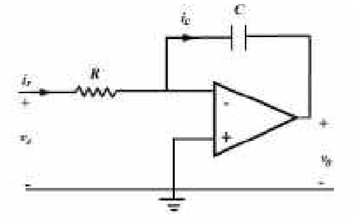
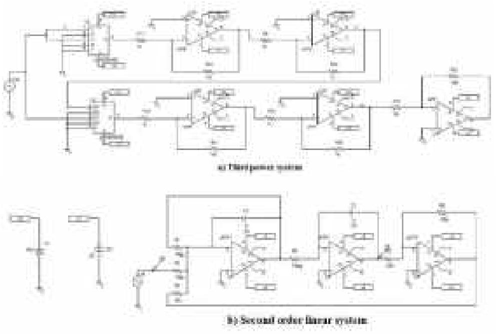


Fig. 10. Integrator circuit with an Op 720


a) Third power system

b) Second order linear system

Fig. 11. The third degree Analog Duffing oscillator

4.5 inverse Volterra Model for the Analog Duffing Oscillator System

From equation (20) and the system model (54), the first inverse Volterra frequency response function model is obtained by the z-transform as:

$$K(j)=\frac{1-0.9804\ z^{-1}-7.321\ \times10^{-6}z^{-2}}{1.9\times10^{-3}}. \tag{56}$$

The corresponding ARX model is.

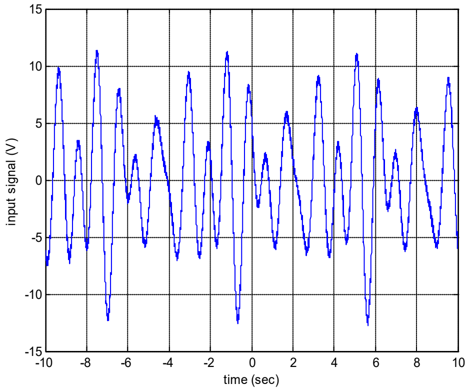


Fig. 12. Input signal into the analog Duffing oscillator

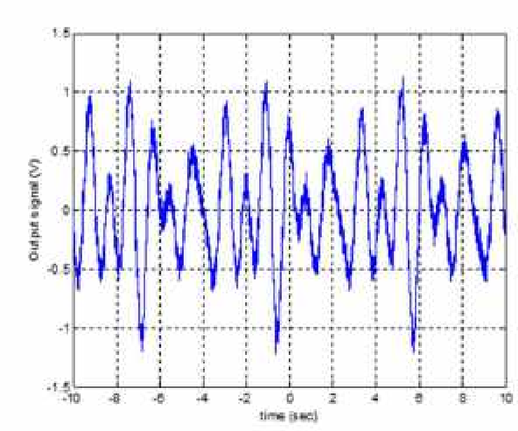


Fig. 13. Output signal from the analog Duffing

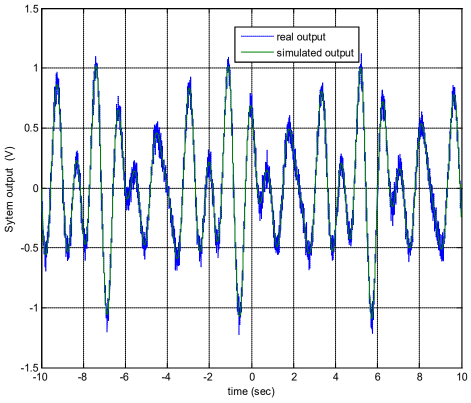


Fig. 14. Comparison between the output of the Duffing oscillator system signal and that obtained by the NARMAX model

$$z_1(j)=\frac{1}{1.9\times10^{-3}}y_1(j)-\frac{0.9804}{1.9\times10^{-3}}y_1(j-1)-\frac{7.321\times10^{-6}}{1.9\times10^{-3}}y_1(j-2) \qquad (57)$$

The second-order operator is according to [2] (see equation (31)):

$$z_2(t)=-\mathrm{K}_1(\mathrm{H}_2[z_1(t)])\cdot \qquad (58)$$

The third-order inverse Volterra operator is now:

$$z_3(t)=-\mathrm{K}_1(\mathrm{H}_3[z_1(t)])-\mathrm{K}_1(\mathrm{H}_3(z_1(t),z_2(t))). \qquad (59)$$

That differs from equation (33) because the second-order operator is not zero in this case. Figure 16 shows the plot of the second and third-order Volterra inverse operators. The equalization strategy is implemented and then the signal that results is shown in Figure 17. The mean square error (mse) between these two signals is only 3.0%. The output signal is quite the same as the signal to measure.

5 Conclusions

Two main objectives have been pursued in this work. The first one is to apply for the first time the equalization strategy on a practical system and on the other hand to present the discrete version of the Associated Linear Equations (ALEs) for both direct and inverse Volterra operators. Two systems were considered: a Duffing oscillator NARMAX model with additive noise and an analog system.

Duffing oscillator system is considered to follow the dynamics of a sensor system.

The equalization strategy of control is implemented for both systems using the discrete version of the inverse Volterra operators. In both cases, the original signal (the signal to be measured) is accurately obtained.

For the simulated system, it was possible to verify an infinite band wide. In the analog system is not possible to do the same, because the elements of the system are not stable at all frequencies and then the Duffing oscillator response is lost.

The use of the Volterra inverse seems to be very promising for systems where feedback is not an alternative, such as measuring and actuator systems.

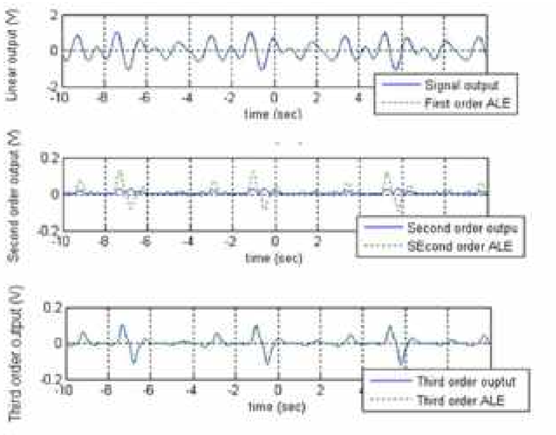


Fig.15. Plot of the ALEs against the corresponding order from the Duffing oscillator system

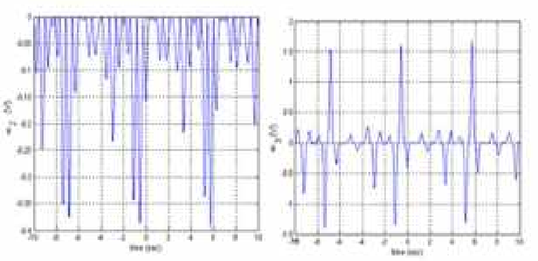


Fig.16. The inverse Volterra operator of second and third-order

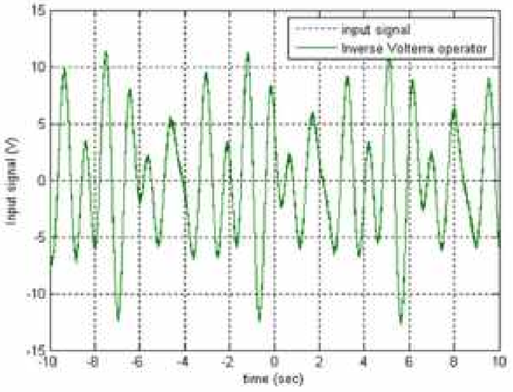


Fig. 17. The system input against the inverse Volterra output

In both cases, the nonlinearities and the inertia of both systems are eliminated and both systems act as elements of unitary gain just as expected.

References

1. **Vazquez-Feijoo, J. A., Worden, K., Stanway, R, Juarez-Rodriguez, N. (2007).** Transformation of a sensor o actuator system into a unitary gain element. Mechanical Systems and Signal Processing, Vol. 21. No. 8, pp. 3088–3107. DOI: 10.1016/j.ymssp.2007.03.008.
2. **Balasubramaniam, B Muthukumar, P., Ratnavelu, K. (2015).** Theoretical and practical applications of fuzzy fractional integral sliding mode control for fractional-order dynamical system. Nonlinear Dyn 80, pp. 249–267. DOI: 10.1007/s11071-014-1865-4.
3. **Vazquez Feijoo, J. A., Worden, K., Stanway, R. (2005).** Associated linear equations for Volterra operators. Mechanical systems and signal Processing, Vol. 19, No. 1, pp. 57–69. DOI: 10.1016/j.ymssp.2004.03.003.
4. **Pedapati, P. K., Pradhan, S. K., Kumar, S. (2019).** Nonlinear adaptive control of an autonomous ground vehicle in obstacle rich environment: some experimental results. TENCON´19 IEEE Region 10 Conference Kochi, India, pp. 614–619. DOI: 10.1109/TENCON.2019.8929625.
5. **Peyton-Jones, J. C., Yaser, K. S. A. (2019).** Computation of the MIMO Volterra frequency response functions of nonlinear systems. Mechanical Systems and Signal Processing, Vol. 134, No. 1, pp. 106323. DOI: 10.1016/j.ymssp.2019.106323.
6. **Chan, S., Billings, S. A., Cowan, C. F. N., Grant, P. M. (1990).** Practical identification of NARMAX models using radial basis functions. International Journal of control, Vol. 52, No. 6, pp. 1327–1350. DOI: 10.1080/00207179008953599.
7. **Jang, H. K., Kim, K. J. (1994).** Identification of cubic stiffness nonlinearity by linearity-conserved NARMAX modeling. Journal of Mechanical Science and Technology, Vol. 8, pp. 332–3428. DOI: 10.1007/BF02953362.
8. **Zhu, Y.P., Lang, Z. Q. (2020).** A new convergence analysis for the Volterra series representation of nonlinear systems.

Automatica, Vol. 111, DOI: 10.1016/j.automatica.2019.108599.

9. **Huang, H., Mao, H., Mao, H., Zheng, W., Huang, Z., Li, X., Wang, X. (2017).** Study of cumulative fatigue damage detection for used parts, with nonlinear output frequency response functions based on NARMAX modelling. Journal of Sound and Vibration, Vol. 411, pp. 75–87. DOI: 10.1016/j.jsv.2017.08.023.
10. **Wootton, A. J., Butcher, J. B., Kyriacou, T., Day, C. R., Haycock, P. H. (2017).** Structural health monitoring of a footbridge using Echo State Networks and NARMAX. Engineering Applications of Artificial Intelligence, Vol. 64, pp. 152–163. DOI: 10.1016/j.engappai.2017.05.014.
11. **Nikiforov, V. O., Voronov, K. V. (2001).** Nonlinear adaptive controller with integral action. IEEE Transactions on Automatic Control, Vol. 46, No. 12, pp. 2035–2037. DOI: 10.1109/9.975516.
12. **Yan, J., Deller-Jr, J. R. (2016).** NARMAX modeling identification using a set-theoretic evolutionary approach. Signal processing, Vol. 123, pp. 30–41. DOI: 10.1016/j.sigpro.2015.12.001.
13. **Starossek, U. (2016).** Exact analytical solutions for forced undamped Duffing oscillator, exact analytical solutions for forced undamped Duffing oscillator. International Journal of Non-Linear Mechanics, Vol. 85, pp. 197-206, DOI: 10.1016/j.ijnonlinmec.2016.06.008.
14. **Luongo, D. Z. A. (2016).** Control of primary and subharmonic resonances of a Duffing oscillator via non-linear energy sink, International Journal of Non-Linear Mechanics, Vol. 80, pp. 170–182. DOI: 0.1016/j.ijnonlinmec.2015.08.014.
15. **Vazquez-Feijoo, J. A., Worden, K., Stanway, R. (2006).** Analysis of time-invariant systems in the time and frequency domain by associated linear equations (ALEs). Mechanical Systems and Signal Processing, Vol. 20, No. 4, pp. 896–919. DOI: /10.1016/j.ymssp.2005.03.004.
16. **Strganac, T. W., Ko, J., Thompson, D. E. (2000).** Identification and control of limit cycle oscillations in aeroelastic systems. Journal of Guidance, Control, and Dynamics, Vol. 23, No. 6, pp. 1127–1133. DOI: 10.2514/2.4664.
17. http://www.ucl.ac.uk/~ucahhwi/LTCC/section2-3-perturb-regula.pdf
18. **Schetzen, M. (1980).** The Volterra and Wiener theories of nonlinear systems. New York: Wiley.
19. http://mathworld.wolfram.com/Z-Transform.html, Weisstain Z-Transform from MathWorld-A Wolfram Web resource.

Article received on 14/10/2020; accepted on 18/08/2022. Corresponding author is Juan Alejandro Vazquez Feijoo.

ISSN 2007-9737

Mapeo sistemático sobre estudios empíricos realizados con colecciones de proyectos software

Juan Andrés Carruthers[1], Jorge Andrés Diaz-Pace[2],
Emanuel Agustín Irrazábal[1]

[1] Universidad Nacional del Nordeste,
Departamento de Informática, Corrientes,
Argentina

[2] Universidad Nacional del Centro de la Provincia de Buenos Aires,
Instituto Superior De Ingeniería Del Software, Buenos Aires,
Argentina

{jacarruthers, eairrazabal}@exa.unne.edu.ar, adiaz@exa.unicen.edu.ar

Resumen. Contexto: los proyectos software son insumos comunes en los experimentos de la Ingeniería del Software, aunque estos muchas veces sean seleccionados sin seguir una estrategia específica, lo cual disminuye la representatividad y replicación de los resultados. Una opción es el uso de colecciones preservadas de proyectos software, pero estas deben ser vigentes y con reglas explícitas que aseguren su actualización a lo largo del tiempo. Objetivo: realizar un estudio secundario sistematizado sobre las estrategias de selección de los proyectos software en estudios empíricos para conocer las reglas tenidas en cuenta, el grado de uso de colecciones de proyectos, los meta-datos extraídos y los análisis estadísticos posteriores realizados. Método: se utilizó un mapeo sistemático para identificar estudios publicados desde enero de 2013 a diciembre de 2020. Resultados: se identificaron 122 estudios de los cuales el 72% utilizó reglas propias para la selección de proyectos y un 27% usó colecciones de proyectos existentes. Asimismo, no se encontraron evidencias de un marco estandarizado para la selección de proyectos, ni la aplicación de métodos estadísticos que se relacionen con la estrategia de recolección de las muestras.

Palabras clave. Colecciones, proyectos software, experimentación, ingeniería del software basada en evidencia.

A Systematic Mapping Study of Empirical Studies Performed with Collections of Software Projects

Abstract. Context: software projects are common resources in Software Engineering experiments, although these are often selected without following a specific strategy, which reduces the representativeness and replication of the results. An option is the use of preserved collections of software projects, but these must be current, with explicit guidelines that guarantee their updating over a long period of time. Goal: to carry out a systematic secondary study about the strategies to select software projects in empirical studies to discover the guidelines taken into account, the degree of use of project collections, the meta-data extracted and the subsequent statistical analysis conducted. Method: A systematic mapping study to identify studies published from January 2013 to December 2020. Results: 122 studies were identified, of which the 72% used their own guidelines for project selection and the 27% used existent project collections. Likewise, there was no evidence of a standardized framework for the project selection process, nor the application of statistical methods that relates with the sample collection strategy.

Keywords. Collections, software projects, experimentation, evidence based software engineering.

1. Introducción

El desarrollo software actual trabaja con la construcción de aplicaciones multi versión [50] que crecen, tanto en complejidad como en funcionalidad, siendo necesario conservar su calidad actual [39].

Por ello, es necesario obtener métodos empíricos para demostrar la calidad del producto software [37] y utilizar evidencia directamente relacionada con el producto software resultante a

doi: 10.13053/CyS-26-4-3981

partir de mediciones que se vinculen con los atributos de calidad del código fuente [19].

En este sentido, la Ingeniería del Software Basada en Evidencia (ISBE) proporciona los medios para que la evidencia actual de la investigación pueda integrarse con la experiencia práctica y los valores humanos en el proceso de toma de decisiones para el desarrollo y mantenimiento de software [32].

En el caso de los experimentos cuyo objeto de estudio es el código fuente, una colección *ad hoc* de proyectos software muchas veces no es suficiente para lograr representatividad o replicación en los resultados. Con la intención de mejorar los resultados surgen las colecciones preservadas de proyectos software, que se utilizan para reducir el costo de recopilar los proyectos software y para que los estudios sean replicables y comparables [60].

Estas colecciones son un insumo para los grupos de trabajo y sirven como mecanismo de comparación para los experimentos en la disciplina de Ingeniería del Software. Existen distintas colecciones, como las realizadas por Barone y Sennrich [5], Allamanis y Sutton [2] o Keivanloo [29], que se diferencian por la cantidad, calidad de proyectos que lo componen o los criterios y métodos de agrupamiento. Por ejemplo, Zerouali y Mens [66] investigaron el uso de las bibliotecas y frameworks de pruebas en Java mientras que Goeminne y Mens [21] trabajaron con frameworks de bases de datos.

En este y otros ejemplos las reglas para la selección de los proyectos muchas veces no son explícitas, se corresponden con selecciones anteriores o son al azar. La definición de un modelo de procedimientos a partir de reglas estándar y herramientas es fundamental para garantizar la capacidad de preservar sistemáticamente la colección a lo largo del tiempo por integrantes del mismo equipo de trabajo o externos, es decir, que la conservación de la misma sea independiente del grupo que la construyó.

Así, por ejemplo, la última versión del Qualitas es del año 2013 [60], lo que hace necesaria la revisión de los proyectos y sus versiones y la generación de los meta-datos necesarios. En otras colecciones, con mayor cantidad de proyectos [2], su diseño no ha tenido en cuenta la extracción de meta-datos necesarios para el estudio de la calidad del producto software.

Por lo antes indicado, el objetivo de este trabajo es determinar los criterios de selección de los proyectos software que son insumos de estudios empíricos, las características de estos proyectos, qué meta-datos se extraen, qué herramientas se utilizan para obtener los meta-datos y qué análisis estadísticos se realizan con los meta-datos recopilados. Se eligió el enfoque de mapeo sistemático para obtener una visión general del desarrollo técnico actual o el nivel de práctica de un área de investigación [51].

Este estudio pretende brindar una visión general sobre las prácticas seguidas por los grupos de investigación para la experimentación con colecciones de proyecto, exponiendo los problemas encontrados que comprometen a la representatividad de las muestras, la replicación de los experimentos y la generación de los resultados, y de esta manera evitar que siga sucediendo en futuros trabajos.

Además de esta introducción, el trabajo se encuentra organizado de la siguiente manera. La sección 2 describe los estudios relacionados con la temática. En la sección 3 se detalla la metodología empleada para el estudio y se presentan las preguntas de investigación. En la sección 4 se reportan las actividades llevadas a cabo. En la sección 5 se discuten y se responden las preguntas de investigación. Finalmente, en la sección 6 se incluye la conclusión del mapeo sistemático desarrollado.

2. Trabajos relacionados

El uso de proyectos software para realización de estudios empíricos no resulta una práctica novedosa en la Ingeniería de Software. En 1971, Knuth publicó su artículo [38] en el cual recolectó aleatoriamente programas en FORTRAN de la Universidad de Stanford y de la compañía Lockheed Missiles and Space. Chidamber y Kemerer en [7] desarrollaron un trabajo para validar las métricas orientadas a objetos creadas por ellos mismos tres años antes [8], una parte del mismo buscaba demostrar la factibilidad de estas métricas en dos sistemas comerciales: uno codificado en el lenguaje C++ y el otro en Smalltalk.

Harrison et al. [22] también condujeron un estudio con cinco proyectos software en tecnología C++ para evaluar y comparar la métrica acoplamiento entre objetos y la métrica número de asociaciones entre una clase y sus pares.

El advenimiento de plataformas para compartir código como SourceForge, aumentó la accesibilidad a proyectos de fuente abierta [60]. Con estos nuevos repositorios públicos fue posible el acceso al código fuente versionado de proyectos software con distintos tamaños, tecnología o perfiles de trabajo de los equipos.

Esto hizo más necesaria la construcción de colecciones de proyectos software que aumente la replicabilidad de los experimentos, se agreguen los resultados, se reduzcan los costos y se mejore la representatividad de las muestras. En la literatura existe una gran variedad de ejemplos de estas colecciones. Barone y Sennrich [5] crearon una colección de más de 100K funciones en Python con sus respectivos cuerpos y descripciones para estudios.

Allamanis y Sutton [2] construyeron el Github Java Corpus con todos los proyectos Java en Github Archive que no fuesen repositorios duplicados. Similar al anterior, Keivanloo et al. [29] incluyeron 24824 proyectos Java, alojados en SourceForge y Google Code.

Zerouali y Mens [66] actualizaron el Github Java Corpus y analizaron el uso de bibliotecas y frameworks de pruebas como JUnit, Spring o TestNG. En esta actualización agregaron los repositorios creados en Github no presentes en la colección original y descartaron aquellos que no estuvieran disponibles y los que no utilizaron la herramienta Maven para gestionar las dependencias, teniendo como resultado una colección de 4532 proyectos. De igual manera, Goeminne y Mens [21] realizaron cambios al Github Java Corpus y estudiaron cinco frameworks de base de datos.

Tempero et al. [60] en el año 2013 constituyeron la colección Qualitas Corpus donde incluyeron sistemas desarrollados en Java con sus archivos fuente y binarios disponibles públicamente en formato ".jar". El objetivo del Qualitas es reducir sustancialmente el costo de los equipos de investigación para desarrollar grandes estudios empíricos del código fuente.

En algunas colecciones, además de la documentación, código fuente y binarios del proyecto, también se proporcionan meta-datos relacionados. Por ejemplo, FLOSSmole [26] brinda los nombres de los proyectos, lenguajes de programación, plataformas, tipo de licencia, sistema operativo y datos de los desarrolladores involucrados. FLOSSMetrics [24] calcula, extrae y almacena información de sistemas de control de versiones, sistemas de rastreo de defectos, archivos de listas de correos y métricas de código. Qualitas.class [61] por su parte, contiene medidas relativas a los proyectos tales como métricas de tamaño (cantidad de líneas de código, número de paquetes, clases e interfaces) y métricas de diseño.

Finalmente, mapeos sistemáticos como los de Falessi et al. [15] y Cosentino et al. [9] estudiaron los conjuntos de datos de proyectos usados en la Ingeniería del Software, debido a la gran diversidad de criterios considerados para seleccionar los proyectos y los meta-datos obtenidos. En ambos trabajos reportaron un bajo nivel replicabilidad de las metodologías empleadas para su recolección.

3. Metodología

Se llevó a cabo un estudio de mapeo sistemático siguiendo las pautas identificadas en [52] para obtener una visión general del uso de colecciones de proyectos en la ISBE. Se seleccionó esta técnica por centrarse en la "clasificación y análisis temático de un tema de la Ingeniería del Software" [31].

En este caso, aunque la recolección de proyectos sea una práctica estándar para la experimentación en ISBE, los criterios de selección de los proyectos software, sus características, los meta-datos que se extraen de estos proyectos, qué herramientas se utilizan para obtener los meta-datos, así como también qué análisis estadísticos se realizan con los meta-datos recopilados no han sido abordados ampliamente por otros estudios secundarios.

Por lo tanto, es necesario identificar, categorizar y analizar la investigación disponible

Tabla 1. Preguntas de investigación

Pregunta de investigación	Motivación
RQ1: ¿Cuáles son los criterios de selección de los proyectos software objeto de estudios empíricos?	Identificar cuáles son los criterios tenidos en cuenta por los investigadores para la selección de proyectos en la realización de estudios empíricos.
RQ2: ¿Con qué tipo de proyectos trabajan los grupos de investigación para realizar estudios empíricos?	Conocer qué características tienen los proyectos seleccionados en términos del tipo de software y lenguaje de programación.
RQ3: ¿Cuáles son los datos/meta-datos que se extraen de estos proyectos?	Determinar los meta-datos y métricas que son extraídos de cada uno de estos proyectos para su posterior experimentación.
RQ4: ¿Qué herramientas se utilizan para obtener estos datos/meta-datos?	Determinar cuáles son las herramientas usadas en los diferentes estudios para recolectar los meta-datos.
RQ5: ¿Qué análisis estadísticos se realizan con los datos/meta-datos recopilados?	Definir los tipos de análisis estadísticos que se desarrollan sobre los experimentos hechos con los proyectos y sus respectivos meta-datos para interpretar los resultados obtenidos.

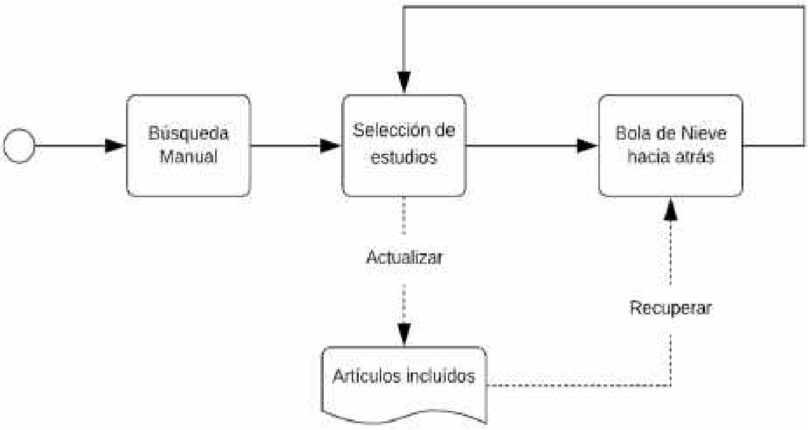


Fig. 1. Proceso de búsqueda y selección de artículos

sobre el tema para describir las prácticas y obtener una visión general de su estado de arte.

Las preguntas de investigación que guiaron el desarrollo del estudio se presentan en la Tabla 1.

3.1. Selección de artículos

Para reunir los estudios primarios relevantes se llevó a cabo una búsqueda y selección iterativa en tres fases tal y como se indica en la Fig. 1.

Búsqueda manual: se realizó una búsqueda manual en las principales conferencias y revistas enfocadas en ISBE. Se seleccionó la revista Empirical Software Engineering (EMSE) y las conferencias Empirical Software Engineering and Measurement (ESEM) e International Conference on Evaluation and Assessment in Software Engineering (EASE) por ser representativas del área de investigación y haber sido utilizadas en otros estudios, como en [67, 47].

De acuerdo a estrategias planteadas en trabajos como [59, 67] y las buenas prácticas de búsquedas manuales de [52], en primera instancia se seleccionó el período de tiempo comprendido entre el 1 de enero del 2013 hasta el 30 de noviembre del 2018 de ESE y ESEM, de forma análoga a [67].

Posteriormente se complementó el rango de búsqueda hasta el 31 de diciembre del 2020 conforme se fueron publicando nuevos números de la revista y el congreso. Después se incorporaron las ediciones del congreso EASE como en [47] considerando el mismo período de tiempo. De esta manera se tomaron las publicaciones de EMSE, ESEM y EASE de los últimos 8 años.

En particular, el instrumento de recolección fue una hoja de cálculo, donde se escribieron los nombres y autores de los artículos de la revista y las conferencias.

Selección de estudios: esta fase consistió en una revisión dual que se realiza de forma iterativa. Los artículos candidatos, después de ser recopilados se incluyeron o excluyeron de acuerdo con los criterios presentes en la Tabla 2.

Los trabajos fueron analizados considerando resumen, introducción, metodología, resultados y conclusión. En cada iteración, se seleccionaron 15 estudios al azar que fueron revisados por dos investigadores.

Los mismos anotaron sus decisiones de incluir o excluir cada estudio, junto con el CI o CE en que se basó la decisión.

Para medir el grado de acuerdo entre los investigadores se utilizó el estadístico Kappa de Cohen, como sugieren [2, 33]. El valor registrado de Kappa de Cohen fue 0.77, lo que demuestra un nivel de acuerdo alto.

Muestreo “bola de nieve”: se complementó la búsqueda manual con el método de bola de nieve hacia atrás con un muestreo de los artículos incluidos. Las referencias proporcionaron artículos relacionados o similares. Aquellos artículos recopilados durante esta etapa también se agregaron a la lista de candidatos y se seleccionaron de acuerdo con los criterios descritos anteriormente. Este proceso se realizó en dos iteraciones.

Evaluación de la calidad: en este trabajo se decidió no realizar una evaluación de calidad de los estudios primarios, al contrario de lo que se propone en las guías de buenas prácticas de revisiones sistemáticas [25, 34].

A diferencia de las revisiones sistemáticas, en los mapeos no es necesario determinar el rigor y la relevancia de los estudios primarios porque su objetivo es proporcionar una visión general del alcance del tema investigado [52]. Finalmente, los artículos seleccionados fueron importados y procesados con la herramienta de gestión de referencias bibliográficas Mendeley[1].

Tabla 2. Criterios de inclusión y exclusión de estudios

Id.	Descripción
CI1	El artículo pertenece a ESE, EASE y ESEM.
CI2	Es un artículo completo.
CI3	En el artículo se realizan estudios empíricos.
CI4	Los estudios empíricos se realizan en base a la selección de un conjunto de proyectos software.
CE1	Artículos no técnicos (guías, artículos de introducción, editoriales, etc.)
CE2	Artículos duplicados.

3.2. Extracción de datos

Se utilizó un cuestionario de extracción de datos basado en las preguntas de investigación para recopilar la información relevante de los estudios primarios. La extracción fue realizada por dos investigadores y revisada por un tercero, comprobando la información presente en el formulario con cada uno de los artículos para verificar la consistencia.

Los datos recolectados incluyeron información general (título, autores, año de publicación y fuente) e información relativa a las preguntas de investigación (RQ1 – RQ5), como se ilustra en la Tabla 3.

La información se extrajo exactamente como los autores la mencionan en los artículos y los conflictos se discutieron y resolvieron internamente por los investigadores. Se adoptó este enfoque para evitar la subjetividad y facilitar la replicación del estudio.

Para dar soporte a este proceso se utilizó la herramienta FRAMEndeley [18], una extensión que permite codificar con tablas el texto resaltado en el gestor de referencias Mendeley valiéndose de los colores de subrayado para distinguir y extraer fragmentos de texto relevantes. En este caso se le asignó un color identificativo a cada RQ.

Previo a la extracción, se realizó una prueba piloto con 15 artículos para calibrar el instrumento de extracción, refinar la estrategia y evitar diferencias entre los investigadores.

[1] https://www.mendeley.com

Tabla 3. Formulario de extracción de datos

	ID	Item	Descripción
General	D1	Título	Título del estudio primario
	D2	Autores	Autores del estudio primario
	D3	Año de Publicación	Año de publicación del estudio primario
	D4	Fuente	Fuente donde se publicó el estudio primario
RQ1	D5	Criterio	Criterio de selección de los proyectos utilizados en el estudio primario
RQ2	D6	Lenguaje	Lenguaje de programación utilizado en los proyectos seleccionados
	D7	Tipo de proyecto	Tipo de proyecto de acuerdo a su funcionalidad
RQ3	D8	Meta-datos	Meta-datos extraídos de los proyectos seleccionados en los estudios primarios
RQ4	D9	Herramientas	Herramientas de recolección de meta-datos
RQ5	D10	Análisis	Tipo de análisis estadísticos que se realiza con los meta-datos extraídos

Tabla 4. Resultados de búsqueda

Año	EMSE	ESEM	EASE
2013	37	65	31
2014	61	72	61
2015	55	41	30
2016	74	58	30
2017	91	70	50
2018	105	58	26
2019	119	54	43
2020	146	43	69
Total	688	461	340

3.3. Análisis de los datos

Las respuestas a las RQ fueron analizadas de forma cualitativa mediante la codificación abierta de los textos encontrados en los estudios primarios recolectados [54]. La selección de esta estrategia se basó en la realizada por Janssen y Van der Voort en [27].

El primer y segundo autor realizaron el proceso de forma separada e identificaron los desacuerdos, que se resolvieron sumando un tercer investigador que visitaba nuevamente la

Tabla 5. Detalle de cada fase de selección de estudios primarios

Fase	Descripción	Incluidos	Excluidos
1	Búsqueda manual	1489	-
2	Selección por criterios de exclusión	1396	93
3	Selección por criterios de inclusión	111	1285
4	Bola de Nieve hacia atrás	11	-

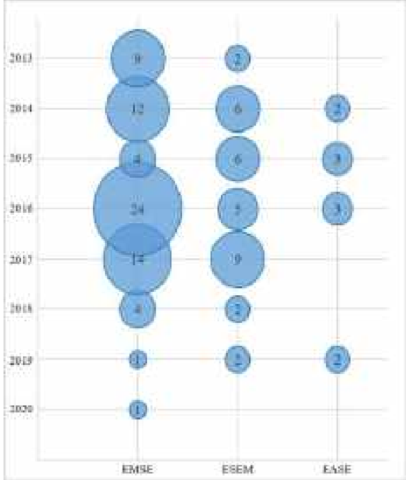


Fig. 2. Distribución de los artículos seleccionados de cada revista por año

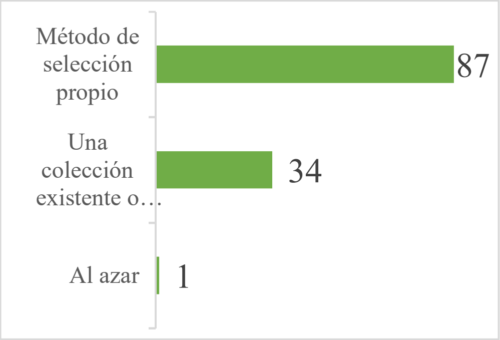


Fig. 3. Método utilizado para elegir los proyectos

fuente de información y tenía en cuenta las justificaciones dadas por los dos integrantes originales.

Como siguiente paso se utilizó la codificación cerrada [10] para identificar y resignificar las respuestas a las taxonomías encontradas. Nuevamente los dos primeros autores del estudio realizaron la codificación inicial y los desacuerdos fueron resueltos mediante la presentación a un tercer investigador.

En la sección 4 de análisis y resultados se pueden encontrar las descripciones de las clasificaciones empleadas.

4. Análisis y resultados

En esta sección, se presentan los datos recopilados y se responde a las preguntas de investigación con los datos extraídos. Los mismos se organizaron y resumieron en tablas y gráficos para una mejor visualización. Para tener un mayor nivel de detalle, en el Anexo[2] están disponibles las planillas originales con los datos extraídos.

4.1. Selección de artículos

Se obtuvo un total de 122 estudios primarios aplicando la estrategia descrita en la Sección 3.1.

La búsqueda se efectuó en la revista ESE y las conferencias EASE y ESEM empleando título, resumen y palabras claves indexadas. Los resultados se observan en la Tabla 4.

En la búsqueda manual fueron recolectados 1489 artículos, de los cuales se incluyeron 1396 y excluyeron 93 según los criterios de exclusión. De los 1396 incluidos en la fase 2, 1285 fueron rechazados por no cumplir alguno de los criterios de inclusión. A los 111 artículos aceptados se le sumaron 11 en la fase 4, quedando finalmente la suma de 122. Los resultados se observan en la Tabla 5.

En la Fig. 2 se puede observar un gráfico de burbujas con la cantidad de estudios seleccionados por año en EMSE, ESEM y EASE. La revista EMSE aportó la mayor cantidad de artículos con un total de 69, seguido por ESEM con 32 y EASE con 10.

[2] http://bit.ly/AnexoTablaCompleta

Desde el año 2013 al 2018 se seleccionaron en promedio casi 20 artículos por año registrándose el punto máximo en el 2016 con 32. Desde el año 2018 en adelante este promedio se reduce a 4 artículos por año siendo el punto mínimo en el 2020 con un solo artículo.

4.2. RQ1: ¿Cuáles son los criterios de selección de los proyectos software objeto de estudios empíricos?

Los estudios recolectados fueron clasificados de acuerdo a la estrategia de selección escogida teniendo en cuenta dos enfoques, uno general y otro particular. En la primera clasificación (ver Fig. 3), que abarca 3 categorías, el 72% de los artículos optaron por un método de selección propio, el 27% utilizaron una colección existente o subgrupo de esta y el 1% los recolectaron al azar.

En la Tabla 6 se encuentran las colecciones que fueron utilizadas en los 33 estudios que implementaron esta estrategia. Algunos ejemplos son: los datasets de PROMISE, el SourceForge 100 (SF100), el Qualitas Corpus y el dataset del Grupo de Estándares y Benchmarking Internacional (ISBSG), entre otros.

Para la segunda clasificación (ver Tabla 7) se definieron 13 categorías. La clasificación fue elaborada en función de los datos recopilados tal y como se indica en el punto 3.3. En total, se extrajeron datos de 94 estudios primarios.

De esta manera, el 36% de los estudios describieron criterios específicos del caso de estudio por el cual se realizaba el experimento. Por ejemplo, repositorios cuyo primer commit no pareciera una migración en [P86], proyectos que incluyan una matriz que indique las fallas que cubren los tests en [P110], sistemas de fuente abierta que sean similares a soluciones industriales en [P22], entre otros.

El 34% de los estudios construyeron la colección de proyectos seleccionando aquellos que estuvieran disponibles públicamente con sus meta-datos asociados. Se mencionaron sitios de alojamiento de proyectos software como: SourceForge en [P16, P25, P94, P107, P119]; ohloh.net en [P86]; Squeaksource en [P23]; Google Play en [P7, P73]; Github en [P1, P9], [P14, P18, P21, P29, P54, P64, P72, P102, P104, P122] o Apple Store en [P73]; entre otros.

Tabla 6. Colecciones de proyectos

#	Nombre de la colección	Artículos
17	PROMISE [1, 4, 6, 12, 28, 30, 35, 42, 44, 55, 56]	P20 P31 P37 P41 P49 P61 P65 P80 P81 P85 P88 P89 P91 P92 P97 P101 P118
2	Qualitas Corpus [60]	P12 P115
2	SF100 [P43]	P42 P99
2	ISBSG [40]	P61 P93
1	FlossMole [26]	P4
1	Vasilescu, et al [63]	P14
1	Centro de desarrollo de un banco de China	P27
1	Tukutuku [43]	P31
1	Yu et al. [65]	P40
1	D’Ambros et al [11]	P41
1	Sistemas de defensa de Corea del Sur	P66
1	Dataset de Finlandia	P68
1	Qualitas Corpus de aplicaciones en Python [49]	P70
1	CVS-Vintage [48]	P71
1	Mockus [45]	P75
1	Mkaouer et al. [45]	P78
1	Departamento de defensa de Estados Unidos	P100
1	Hamasaki et al [22]	P111

El 29% consideraron la actividad del proyecto a lo largo del tiempo como factor de selección. La actividad puede ser medida por cantidad de años de desarrollo o cantidad de commits realizados.

Así se encuentran casos como en [P94] que selecciona proyectos con 3 o más años, en [P58] recolecta proyectos con más de 10K commits, o en [P119] que excluye aquellos sistemas que tengan menos de 32 commits y un año de desarrollo.

El 28% eligieron la popularidad en repositorios públicos como criterio de recolección. Dependiendo del trabajo se tomaron enfoques diferentes para cuantificarla. Por ejemplo, la

Tabla 7. Criterios tenidos en cuenta para seleccionar los proyectos

#	Criterios	Artículos
34	Específicas del caso de estudio del articulo	P2 P8 P9 P14 P18 P19 P22 P30 P34 P51 P55 P58 P61 P62 P64 P65 P66 P68 P72 P74 P79 P83 P86 P88 P90 P98 P100 P102 P105 P108 P110 P114 P116 P119 P122
32	Proyectos y meta-datos disponibles públicamente	P1 P6 P7 P9 P13 P14 P16 P18 P21 P23 P25 P26 P29 P30 P54 P64 P71 P72 P73 P76 P83 P86 P94 P95 P102 P104 P107 P113 P114 P119 P120 P122
27	Actividad en el proyecto a lo largo del tiempo	P3 P26 P44 P51 P53 P54 P55 P57 P58 P60 P64 P72 P75 P78 P79 P87 P90 P93 P94 P95 P100 P104 P106 P109 P111 P117 P119 P121
26	Popularidad	P1 P8 P10 P16 P19 P21 P26 P29 P34 P45 P48 P54 P56 P63 P64 P73 P76 P79 P86 P102 P104 P106 P109 P114 P116 P121
24	Dominio o tipo de software	P1 P4 P6 P8 P10 P12 P22 P27 P31 P32 P33 P50 P53 P55 P56 P57 P67 P73 P77 P84 P86 P99 P106 P121
24	Tamaño	P1 P2 P3 P4 P6 P16 P18 P33 P35 P55 P56 P57 P61 P68 P72 P78 P79 P85 P87 P88 P102 P106 P113 P120
16	Usan herramientas que dan soporte a procesos	P2 P3 P10 P17 P18 P44 P48 P58 P64 P69 P74 P75 P102 P107 P111 P117
12	Actividad reciente en el momento de recolección	P2 P14 P21 P25 P54 P75 P76 P78 P83 P94 P104 P117
10	Disponibilidad Información de defectos	P13 P34 P78 P90 P94 P95 P107 P114 P119 P120
8	Equipo de Desarrollo	P16 P26 P51 P72 P87 P104 P113 P114
8	Calidad	P9 P15 P16 P61 P68 P105 P108 P114
7	Disponibilidad datos históricos	P26 P31 P44 P51 P94 P106 P121
5	Documentación	P14 P16 P79 P106 P114

popularidad se expresó términos de: el número de descargas [P16]; la cantidad de usuarios como [P8, P26, P106, P109, P121] el número de visitas [P29]; el número de duplicaciones del repositorio [P104]; o la cantidad de reseñas de usuarios [P73], entre otras.

En el 26% de los estudios primarios consideraron como criterio el dominio o tipo de software, es decir, la funcionalidad o ámbito de aplicación del mismo. En 21 artículos ([P1, P4, P6, P10, P12, P22, P31, P32, P33, P50, P55, P56, P57, P67, P73, P77, P84, P86, P99, P106, P121]) mencionan que los sistemas seleccionados deben provenir de múltiples dominios de aplicación.

En los tres casos restantes recolectaron proyectos de usos menos generales, como bases de datos en [P53], aplicaciones de propósito general en [P8], y ámbitos más específicos como sistemas de un banco comercial en China en [P27].

El 26% tuvieron en cuenta el tamaño de los proyectos cuantificado en términos de clases como en [P6, P16 P57, P102]; módulos en [P35]; líneas de código [P18, P55, P87, P88, P113]; puntos de función IFPUG en [P61] o puntos de función FiSMA en [P68].

En el 17% de los casos una condición fue el uso de herramientas para dar soporte a procesos dentro del desarrollo del proyecto. Esto incluyó procesos tales como: gestión de dependencias en [P2, P3, P18, P48, P64, P75, P102]; gestión de versiones en [P10, P17, P44, P58 , P69 P107, P117] o revisión de código en [P74, P111].

El 13% buscaron proyectos en los que se haya registrado actividad reciente al momento de recolección, es decir, que los mismos sigan siendo mantenidos o actualizados por sus colaboradores. La actividad suele ser medida en base a la cantidad de commits realizados en un periodo de tiempo [P14].

El 11% de los estudios establecieron como criterio de selección que hubiera disponibilidad de información de defectos en sistemas de rastreo de defectos. En el 9% de los artículos recolectaron proyectos según la cantidad de participantes en el equipo de desarrollo o si existiera una comunidad que ofreciera soporte.

También con un 9% fue considerada la calidad del proyecto, donde se utilizaron diversos métodos para establecerla. En [P108] tomaron proyectos orientados a objetos bien diseñados con evidencias de prácticas agiles; en [P9, P105, P114] filtraron los proyectos de baja calidad con la herramienta Reaper; en [P61, P68] evaluaron la calidad según los métodos de IFPUG y FiSMA respectivamente.

El 7% buscaron aquellos proyectos en los que existiera datos históricos del proceso de desarrollo, en algunos casos para recuperar información evolutiva del sistema. Y en el 5% de los casos, la documentación del proyecto fue un factor determinante, en medios como los comentarios en el código fuente ([P14, P16, P79]), documentación de desarrollo en [P106] y licencia del software en [P114].

4.3. RQ2: ¿Con qué tipo de proyectos trabajan los grupos de investigación para realizar estudios empíricos?

Cada uno de los proyectos software en las colecciones utilizados en los estudios seleccionados fueron clasificados según el lenguaje de programación principal (Fig. 4) y el tipo de software (Fig. 5). En total se registraron 110 artículos que describieron los lenguajes de programación principales de los proyectos. En un 79% fueron construidos en Java, seguido por C con un 27% y C++ con un 24%.

Para identificar los tipos de software se buscaron los nombres y descripciones de proyectos mencionadas en los estudios. En los casos donde los datos extraídos no eran suficientes para determinar el tipo de software utilizado, se realizó una búsqueda y consulta manuales.

Cada proyecto fue clasificado según la taxonomía publicada en [16], de donde se emplearon cinco categorías de segundo nivel: diseño y construcción de software, servidor, redes y comunicaciones, sistema operativo y sistema embebido (ver Fig. 6).

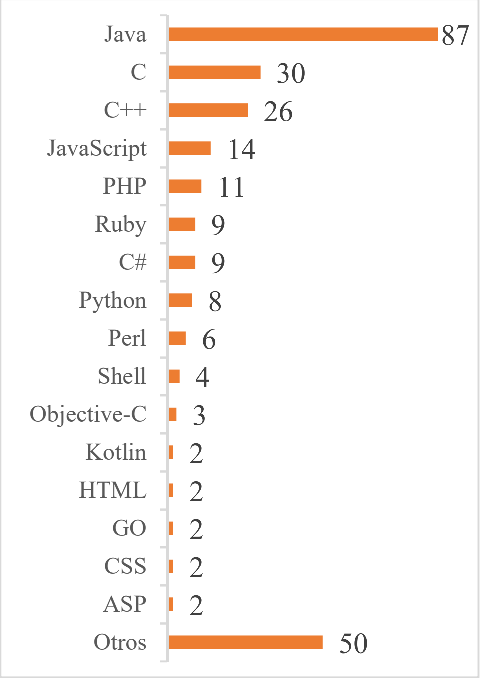


Fig. 4. Lenguajes de programación de los proyectos

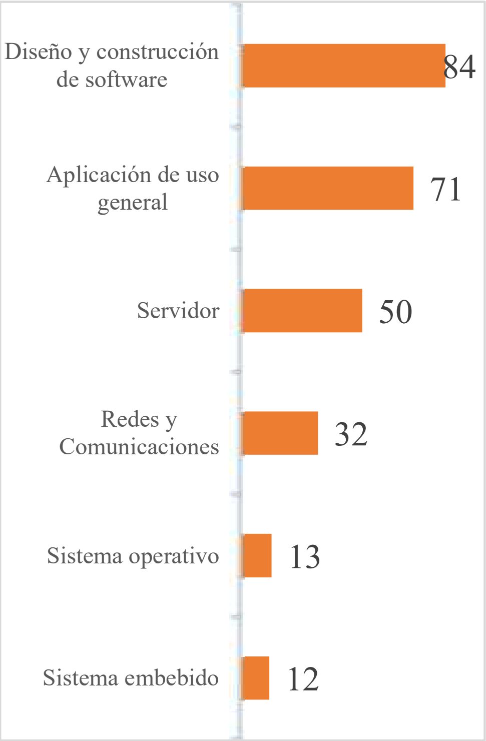


Fig. 5. Tipos de software

Para comprender las demás categorías dentro de la taxonomía original que abarcan al software guiado por datos, orientado al consumidor y de propósito general se usó el término “aplicación de uso general”.

De los estudios primarios seleccionados 96 aportaron esta información.

El 88% de los artículos trabajaron con proyectos software para el diseño y construcción de sistemas.

En este sentido se consideran compiladores como: Clang en [P47], frameworks de desarrollo como Spring Framework en [P86], entornos de desarrollo integrado como Eclipse en [P36], interfaces de programación de aplicaciones como Apache iBatis en [P43], herramientas de construcción como Ant en [P86], herramientas para el modelado del sistema como ArgoUML en [P43] o librerías para la generación de casos de prueba como jUnit en [P59], entre otros.

El 74% experimentaron con aplicaciones de uso general. En esta clasificación entran: procesadores de texto como jEdit en [P94], videojuegos como Freecol en [P12], navegadores como Firefox en [P109], aplicaciones cliente de mensajería como Pidgin en [P94], sitios web como phpMyAdmin en [P94], aplicaciones android en [P30] o reproductores de música como aTunes en [P103], entre otras.

El 52% se identificó con servidores, es decir, sistemas preparados para la recepción de solicitudes de clientes. Por ejemplo, servidores web como Apache en [P43], de base de datos como MySQL en [P53], de transferencia de archivos como FileZilla en [P94], para computación distribuida como Hadoop en [P26] o de instancias de virtualización como Apache Cloudstack en [P58].

En el 33% utilizaron software de sistema para tareas de redes y comunicaciones entre aplicaciones y dispositivos, por ejemplo: Apache Synapse en [P85], dnsjava en [P71], Quagga en [P47], ActiveMQ en [P58]. El 14% trabajaron con sistemas operativos, tales como Linux en [P109] y Android en [P75].

En el 13% estudiaron sistemas embebidos, es decir, software embebido en un dispositivo mecánico o eléctrico diseñado para realizar una función específica.

Tabla 8. Categorías de meta-datos

#	Categorías de Meta-datos	Artículos
36	Tamaño Código Fuente	P6 P9 P10 P11 P12 P18 P20 P21 P25 P26 P29 P38 P41 P46 P49 P50 P55 P56 P58 P65 P66 P69 P72 P74 P80 P81 P83 P85 P86 P89 P96 P97 P100 P105 P114 P119
34	Diseño	P6 P9 P12 P15 P16 P18 P20 P21 P25 P29 P37 P41 P46 P49 P50 P55 P59 P65 P69 P74 P78 P80 P81 P84 P85 P96 P97 P102 P105 P113 P115 P117 P118 P119
17	Code Smells	P2 P3 P7 P10 P12 P18 P19 P21 P29 P37 P44 P56 P63 P78 P79 P85 P105

Por ejemplo, en [P66] utilizaron proyectos software armamentísticos de Corea del Sur, y en [P101] sistemas controladores de una compañía de Turquía. En [P55] sistemas industriales embebidos como controladores de motores de combustión o soluciones de procesamiento de audio.

4.4. RQ3: ¿Cuáles son los datos/meta-datos que se extraen de estos proyectos?

Dependiendo del objeto de estudio del artículo se recolectaron diferentes meta-datos del código de cada uno de los proyectos.

En la Tabla 8 se agruparon los meta-datos siguiendo la taxonomía elaborada en [14] debido a la gran variedad existente.

Las categorías consideradas de esta taxonomía son aquellas que abarcan métricas obtenibles por medio del análisis estático del código.

De los 54 estudios que extrajeron meta-datos, las métricas presentes son de tamaño del código fuente en un 67%, por ejemplo, líneas de código (LOC) en [P6], volumen de Halstead en [P46] o líneas de comentarios también en [P46].

En el 63% de artículos se utilizan métricas de diseño que miden propiedades inherentes del software y se puede disgregar en otras 7 subcategorías: complejidad, acoplamiento, diagrama de clases, herencia, cohesión, encapsulamiento y polimorfismo (ver Tabla 9).

Las métricas de complejidad calculan la cantidad de caminos existentes en el flujo de control del programa; como la complejidad ciclomática de McCabe en [P25], o el peso de métodos por clase en [P85].

Las métricas de acoplamiento cuantifican la asociación o interdependencia entre los módulos, como el acoplamiento entre objetos en [P85] o la respuesta por clase en [P6].

Las medidas de diagrama de clases representan la estructura estática del sistema y pueden clasificarse en términos de clases, métodos y atributos (Tabla 10), por ejemplo: el número de métodos en [P6], el número de clases en [P12] o el número de atributos en [P6].

Las métricas de herencia miden los atributos y métodos compartidos entre clases en una relación jerárquica, dentro de las cuales están el número de hijos en [P85], la profundidad de árbol de herencia en [P25].

Las métricas de cohesión establecen el grado en que métodos y atributos de una misma clase están conectados, como la falta de cohesión en los métodos, la cohesión de clases ajustada o la cohesión de clases suelta todas en [P6].

Las medidas de encapsulamiento calculan los datos y funciones empaquetadas en una sola unidad, por ejemplo, el número de métodos accessors en [P12], la métrica de acceso a datos en [P78].

Y las medidas de polimorfismo se usan para cuantificar la habilidad de un componente de tomar múltiples formas, como el número de métodos polimórficos en [P78].

La tercera categoría es code smells. Los code smells son estructuras en el código fuente que a menudo indican la existencia de un problema de calidad o estructural, y plantean la necesidad de realizar una refactorización. Es una forma disciplinada de limpiar el código que reduce las chances de insertar defectos [17].

Si bien esta categoría no se encuentra presente en la taxonomía original [14], es razonable incluirla en la misma porque se ha establecido como un método efectivo para descubrir problemas en el código fuente y por medio de la refactorización, mejorar la calidad y mantenimiento del software [P12]. Dicho esto, el 32% de los estudios recolectan code smells.

Tabla 9. Subcategorías de meta-datos de diseño

#	Subcategorías de Diseño	Artículos
23	Complejidad	P9 P12 P18 P20 P21 P25 P29 P37 P41 P46 P49 P50 P55 P69 P74 P80 P81 P85 P97 P105 P117 P118 P119
19	Acoplamiento	P6 P9 P12 P15 P20 P25 P29 P37 P49 P59 P65 P69 P78 P80 P81 P85 P96 P118 P119
18	Diagrama de Clases	P6 P9 P12 P20 P25 P29 P49 P50 P65 P69 P78 P80 P81 P96 P102 P113 P118 P119
15	Herencia	P9 P12 P16 P20 P25 P37 P49 P69 P78 P80 P85 P105 P115 P118 P119
15	Cohesión	P6 P9 P20 P25 P37 P49 P65 P69 P78 P80 P81 P84 P85 P118 P119
9	Encapsulamiento	P12 P20 P49 P65 P69 P78 P80 P96 P118
1	Polimorfismo	P78

Tabla 10. Subcategorías de meta-datos de diagrama de clases

#	Subcategorías de Diagrama de Clases	Artículos
16	Métodos	P6 P9 P12 P20 P25 P29 P49 P50 P65 P69 P78 P80 P81 P113 P118 P119
9	Atributos	P6 P12 P20 P49 P69 P78 P80 P81 P118
5	Clases	P9 P12 P78 P96 P102

4.5. RQ4: ¿Qué herramientas se utilizan para obtener estos datos/meta-datos?

Es común en estos estudios hallar diferentes herramientas para dar soporte a tareas específicas para lograr precisión y repetitividad [41]. Dada la variedad de herramientas encontradas, se realizó una clasificación en cinco categorías en base a las funciones que las mismas desempeñan.

En la Fig. 6 se puede observar un gráfico de burbujas con la clasificación antes mencionada (eje vertical), donde están descripta la cantidad de herramientas encontradas, cuántas de ellas siguen vigentes, es decir, si la última versión estable sigue en funcionamiento y cuantas han sido actualizadas después del 1º de enero del 2020 (eje horizontal). Dentro de las herramietas halladas se encuentran: Understand en [P25], CKJM en [P69] o PMD en [P85].

4.6. RQ5: ¿Qué análisis estadísticos se realizan con los datos/meta-datos recopilados?

Los investigadores necesariamente realizan un análisis previo a los datos recabados para exponer los hallazgos encontrados. Este proceso permite identificar las características que posee la muestra y a su vez seleccionar los estudios apropiados para generar los resultados. De los artículos recolectados se extrajeron los análisis estadísticos utilizados y se clasificaron en las dos áreas generales de la estadística: estadística inferencial y estadística descriptiva (ver Tabla 11), tomando como referencia las clasificaciones de [57]. En total 88 estudios informaron los procedimientos estadísticos realizados.

La estadística inferencial emplea los datos para sacar conclusiones o hacer predicciones. En el 78% de los artículos seleccionaron procedimientos inferenciales que se clasificaron en 2 categorías, test no paramétrico y test paramétrico (ver Tabla 12).

Una de las diferencias entre estos grupos es que los test paramétricos hacen suposiciones específicas con respecto a uno o más parámetros de la población que caracterizan la distribución subyacente para la cual el test está siendo empleado. Ejemplos de test paramétricos son: test T en [P23], test chi cuadrado de Wald en [P74], y test de análisis de varianza (ANOVA) en [P94].

Tabla 12. Procedimientos estadísticos inferenciales

#	Procedimientos Inferenciales	Artículos
61	Test No Paramétrico	P1 P2 P4 P5 P9 P13 P15 P16 P19 P20 P22 P23 P25 P26 P31 P32 P33 P34 P35 P36 P37 P42 P49 P50 P51 P54 P55 P56 P57 P59 P60 P61 P62 P67 P68 P69 P73 P74 P75 P77 P78 P79 P80 P83 P86 P90 P91 P94 P96 P99 P101 P103 P110 P111 P113 P114 P116 P118 P119 P120 P121
22	Test Paramétrico	P3 P9 P13 P23 P25 P33 P39 P40 P60 P65 P74 P75 P80 P81 P85 P90 P94 P97 P99 P103 P111 P121

Tabla 13. Procedimientos estadísticos descriptivos

#	Procedimientos Descriptivos	Artículos
51	Tamaño del efecto	P2 P3 P4 P6 P10 P14 P19 P20 P23 P26 P29 P32 P33 P35 P37 P41 P42 P54 P56 P57 P58 P60 P64 P71 P74 P76 P77 P79 P80 P83 P84 P86 P87 P90 P92 P93 P95 P96 P99 P101 P103 P105 P106 P110 P111 P113 P114 P116 P118 P119 P120
26	Medida de Variabilidad	P2 P3 P6 P13 P22 P23 P24 P25 P31 P32 P35 P36 P37 P43 P60 P64 P68 P69 P72 P74 P77 P80 P91 P94 P103 P111
25	Medida de Tendencia Central	P2 P3 P6 P13 P22 P23 P24 P25 P31 P32 P35 P36 P37 P43 P60 P64 P68 P69 P72 P74 P77 P80 P94 P103 P111
3	Medida de Asimetría	P13 P43 P61
2	Medida de Curtosis	P13 P43

Dentro de los no paramétricos hay ejemplos como test Mann-Whitney en [P36], test de rangos con signo de Wilcoxon en [P31], y test de normalidad Shapiro-Wilk en [P103].

Por otra parte, el 73% de los artículos trabaja con procedimientos descriptivos utilizados para presentar y resumir los datos. Estos procedimientos fueron clasificados en cinco categorías: tamaño del efecto, medida de variabilidad, medida de tendencia central, medida de asimetría y medida de curtosis (ver Tabla 13).

El tamaño de efecto mide la magnitud de fuerza de un fenómeno o efecto. En Kitchenham et al. [36] remarcan su utilidad porque proporcionan una medida objetiva de la importancia que tiene un fenómeno en un experimento, independientemente de la significación estadística de la prueba de hipótesis conducida.

Además, le afecta menos el tamaño de la muestra que a la significación estadística. Por ejemplo, podemos mencionar: el coeficiente de correlación Spearman en [P86], el tamaño de efecto de Cliff en [P103] o el coeficiente de correlación de Pearson en [P29].

Las medidas de variabilidad y las de tendencia central son los estadísticos más básicos empleados tanto en la estadística inferencial como en la descriptiva. En esta clasificación fueron incluidos solamente dentro de los procedimientos descriptivos porque en cada caso los usaron con el fin de describir la muestra recolectada. Como ejemplos de medidas de variabilidad se encuentran la desviación estándar en [P13], los cuartiles en [P6] y la varianza en [P24]. En el caso de medidas de tendencia central están la media en [P74], la mediana en [P43] y la moda en [P24].

5. Discusión

En esta sección se discuten los resultados obtenidos y como se relacionan con las preguntas de investigación identificadas en la sección 2. Sin embargo, es posible que no se hayan localizado todos los estudios relevantes, por esa razón el proceso fue desarrollado metodológicamente siguiendo un protocolo bien definido y múltiples investigadores revisaron la calidad de la información extraída. El mapeo comprendió 122 artículos publicados en la revista EMSE y las conferencias ESEM y EASE durante el período 2013 – 2020.

5.1. RQ1: ¿Cuáles son los criterios de selección de los proyectos software objeto de estudios empíricos?

Para abordar esta pregunta se buscaron evidencias de la selección de proyectos software con dos enfoques, uno a nivel general y otro más específico. A nivel general, en la mayoría de los casos (72%) los investigadores siguen lineamientos propios para seleccionar sus proyectos y en menor medida (27%) utilizan una colección de proyectos existentes.

De los 94 estudios que reportaron información al respecto, los criterios de selección predominantes son: los específicos del caso de estudio (36%), junto con la disponibilidad de los proyectos y meta-datos (34%), el período de actividad en el proyecto (29%), la popularidad (28%), el dominio de aplicación (26%) el cual muchas veces no se encuentra descrito con precisión, el tamaño del proyecto (26%) que tiene en cuenta diferentes dimensiones, desde la cantidad de líneas de código hasta medidas de punto función.

La Tabla 7 evidencia la diversidad de estrategias existentes para recolectar las muestras. En particular, algunos estudios optan por automatizar el proceso utilizando herramientas para explorar repositorios públicos como Github o SourceForge.

Así, en [P9, P83, P105] aplican un framework, que permite seleccionar repositorios Github que contengan “proyectos que aprovechan las prácticas sólidas de ingeniería de software en una o más de sus dimensiones, como la documentación, pruebas y gestión de proyectos” [P83].

O en [P24, P107] que utilizan un lenguaje de programación de dominio específico para el análisis y minado de repositorios software de gran escala [13].

En resumen, se evidencias estrategias diversas, pocos descritas o justificadas y en ocasiones sesgadas para los estudios.

5.2. Q2: ¿Con qué tipo de proyectos trabajan los grupos de investigación para realizar estudios empíricos?

Para caracterizar la muestra de los proyectos software recolectados por los grupos de investigación se describieron el lenguaje de programación principal y el tipo de actividad o función que desempeña. De 110 artículos que reportan el lenguaje de programación principal de los proyectos, la mayoría (79%) experimentan con Java y en menor grado (31%) los lenguajes C o C++. Esto puede tener una relación directa con la cantidad de proyectos en los repositorios de ambos lenguajes [20].

Otra razón es la gran cantidad de herramientas de análisis estático del código fuente compatibles disponibles para el caso del lenguaje Java. Aún así, trabajos como [P31, P60, P66, P72, P75, P83, P104, P114, P119, P121] recolectaron proyectos con 5 o más lenguajes.

Finalmente, de los 96 artículos que fue posible determinar el tipo de software empleado la mayoría era para el diseño y construcción de sistemas (88%) y, en segundo lugar, aplicaciones de uso general (74%).

5.3. RQ3: ¿Cuáles son los datos/meta-datos que se extraen de estos proyectos?

Para responder esta pregunta se buscaron los meta-datos extraídos de los proyectos software para la realización de los experimentos. Así se encontraron 54 estudios primarios que reúnen meta-datos del código en forma de métricas, indicadores o medidas.

De esta manera en este conjunto de artículos se pueden encontrar métricas que cuantifican el tamaño (67%), diseño (63%) y los code smells (32%). Esto evidencia el uso de meta-datos básicos en primera instancia y la posibilidad de incluir interpretaciones en una segunda instancia.

El resto de los trabajos se valieron de otras fuentes de información. Por ejemplo, en 13 artículos utilizaron los reportes de defectos para calcular su tiempo de resolución en [P13, P26, P94]; clasificarlos en [P52, P106, P109, P112] o detectar reportes duplicados en [P95]; entre otros estudios.

En 6 artículos recolectaron la información de la ejecución del programa (análisis dinámico) para localizar funciones específicas en [P36], clasificar los procesos en [P27], estudiar la asignación de memoria en [P28], analizar cómo trabajan las excepciones en [P30] o las dependencias del sistema en ejecución en [P4].

En 50 artículos extrajeron los datos contenidos en el repositorio del proyecto para medir el acoplamiento y dependencias en la evolución del sistema en [P16, P51, P59]; clasificar los commits en [P38, P53]; predecir defectos en el software en [P101]; estudiar el uso de patrones en la historia del proyecto en [P57] o el aporte de los desarrolladores [P74], entre otros.

Para estos casos es necesario que los proyectos tengan un conjunto de meta-datos registrados a lo largo de su tiempo de desarrollo. Esto se consigue muchas veces en la gestión del proyecto con una herramienta de control de versiones y desarrollo colaborativo.

5.4. RQ4: ¿Qué herramientas se utilizan para obtener estos datos?

El objetivo de esta pregunta es conocer las herramientas utilizadas en la Ingeniería del Software para la recolección de meta-datos de proyectos. Es notable que solamente 61 artículos mencionan de manera explícita los nombres de las herramientas utilizadas. De los cuales 44 estudios informan herramientas que estrictamente generen meta-datos de los proyectos, siendo este un requisito indispensable para la replicabilidad de los experimentos.

En muchas ocasiones se informa el modelo, técnica, procedimiento o algoritmo utilizado, pero no así la herramienta utilizada. Por ejemplo, en [P36] utilizan un modelo de fusión de datos compuesto por técnicas de recuperación de información, análisis dinámicos y minado Web para la localización de métodos que desarrollan funciones específicas del sistema.

En [P31] implementan un algoritmo de búsqueda meta-heurístico en un modelo de aprendizaje automatizado para estimar el esfuerzo de desarrollo. En [P27] declararon el uso de un “método propio” para la refactorización de artefactos software.

En [P6] trabajaron con una herramienta con la que calculan 29 métricas de código fuente pero no reportan su nombre, ni como acceder a la misma.

Dicho esto, en los 44 estudios (ver Tabla 15) usaron herramientas para el cálculo de métricas de software (57%), análisis de la estructura y dependencias del código fuente (37%), detección de code smells o vulnerabilidades (37%), refactorización de código (9%), generación automática de tests (9%) y detección patrones de diseño (5%). En la Tabla 14 y en la Tabla 15 se encuentran las cuatro herramientas más repetidas en los artículos seleccionados.

Existen herramientas que se ubican en más de una categoría de la clasificación presentada en la Fig. 6 y Tabla 14. Understand, Alitheia Core y Analizo calculan métricas de software y analizan la estructura y dependencias del código fuente. PMD, iPlasma, SonarQube, Designite e InCode también calculan métricas y además detectan code smells y vulnerabilidades del código.

5.5. RQ5: ¿Qué análisis estadísticos se realizan con los datos recopilados?

Esta pregunta pretende determinar qué técnicas o procedimientos estadísticos son elegidos por los investigadores para validar los resultados de los experimentos realizados.

En este sentido, la selección y aplicación apropiada de los métodos de análisis es una de las recomendaciones de Wohlin y Rainer en [63] para garantizar que la evidencia generada se presente correctamente y evitar que existan interpretaciones erróneas de los resultados.

De los 88 artículos que se registraron procedimientos estadísticos el 78% son inferenciales, de los cuales el 25% contiene pruebas o tests estadísticos paramétricos para el análisis de los datos recopilados.

Esto implica supuestos, como que los datos obtenidos sigan una distribución normal, lo que podría no coincidir con la forma de selección de los proyectos o los mecanismos de extracción de los datos. Así, por ejemplo, muchos estudios utilizan reglas basadas en la disponibilidad del proyecto o su popularidad, pero no en la representatividad de los meta-datos con respecto a la población.

Tabla 14. Tipo de herramientas

#	Tipo de Herramientas	Artículos
25	Cálculo métricas de software	P2 P3 P4 P9 P12 P17 P18 P20 P21 P25 P26 P46 P50 P66 P67 P69 P74 P78 P83 P85 P105 P107 P113 P116 P119
17	Análisis de la estructura y dependencias del código fuente	P2 P4 P8 P9 P16 P21 P25 P35 P46 P50 P55 P57 P64 P74 P90 P116 P119
17	Detección de code smells o vulnerabilidades	P2 P3 P12 P18 P19 P20 P29 P37 P44 P46 P56 P57 P63 P67 P78 P85 P105
4	Refactorización de código	P9 P21 P54 P84
4	Generación automática de tests	P42 P43 P62 P99
2	Detección patrones de diseño	P57 P78

Tabla 15. Herramientas más utilizadas

#	Herramientas	Artículos
8	Understand[3]	P2 P4 P9 P21 P25 P74 P116 P119
4	Evo Suite[4]	P42 P43 P62 P99
3	CKJM[5]	P20 P25 P69
3	PMD[6]	P12 P67 P85

[3] https://www.scitools.com
[4] https://www.evosuite.org
[5] https://www.spinellis.gr/sw/ckjm
[6] https://pmd.github.io

Finalmente, en 51 estudios incorporaron un análisis de tamaño del efecto, siendo esta una recomendación indicada para los casos en que las muestras sean poco representativas y se utilicen técnicas paramétricas [36].

5.6. Amenazas a la validez

A continuación, se discuten las amenazas a la validez del estudio siguiendo el enfoque propuesto por [53]. Se consideraron cinco aspectos de validez.

Validez de constructo. La validez del constructo refleja hasta qué punto la metodología de la investigación representa a la estrategia del investigador y lo que se busca estudiar en las preguntas de investigación. Esta amenaza está presente al diseñar el instrumento de extracción de datos. La misma disminuyó implementando una prueba piloto de la tabla, tomando artículos al azar para completar una primera versión, y modificándola iterativamente según sea necesario hasta lograr la versión final.

Validez interna. Este aspecto de validez analiza los riesgos cuando se estudian relaciones causales [58]. Elegir los artículos y el período de tiempo adecuados son factores importantes que afectan la validez interna. La cantidad de artículos relacionados con la ISBE ha crecido mucho recientemente y las revistas de Ingeniería del Software tienen diferentes grados de aceptación para investigaciones empíricas. Para mitigar esta amenaza, se seleccionaron artículos de revistas y conferencias ampliamente aceptados en el ámbito de la Ingeniería del Software en términos de alcance y reputación. La subjetividad en la selección disminuyó mediante el proceso descrito en la Sección 3.1, realizando tantas iteraciones como fueron necesarias hasta obtener un grado de acuerdo alto.

Validez externa. La validez externa se refiere hasta qué punto es posible generalizar los resultados más allá del estudio. Los hallazgos aquí presentados se basan en las publicaciones pertenecientes a EMSE, EASE y ESEM. Si bien se desconoce si los resultados se pueden generalizar a artículos de otras revistas o conferencias, la investigación se basó en el análisis de 122 artículos y, por tanto, puede considerarse representativa.

Fiabilidad. La fiabilidad evidencia hasta qué punto los resultados de la investigación son independientes de los investigadores. Es decir, si otro autor llevase a cabo el mismo estudio, los resultados deberían ser iguales o similares [53]. En este artículo, los métodos y procesos de investigación se describen en detalle para garantizar su reproducibilidad y en el anexo se incluyen las planillas originales con los datos extraídos.

Sesgo de publicación. El sesgo de publicación se refiere al "problema de que es más probable que se publiquen los resultados positivos de la investigación que los negativos" [62]. Este problema ocurre en cualquier revisión de literatura o estudio de mapeo. Sin embargo, en este caso, su efecto fue moderado porque nuestro estudio no pretende comparar resultados de investigación.

6. Conclusiones

En este mapeo sistemático se identificaron artículos en los que se realizaron estudios empíricos con colecciones de proyectos. Se abordaron cinco preguntas de investigación de estos estudios, como los criterios de selección de proyectos, su caracterización, los meta-datos recolectados en los estudios empíricos, las herramientas utilizadas para generar u obtener los meta-datos y los análisis estadísticos desarrollados con ellos.

Por medio de una búsqueda manual inicial en la revista EMSE y las conferencias ESEM y EASE se obtuvieron 1496 artículos entre el 1 de enero del 2013 al 31 de diciembre del 2020. De los cuales 122 estudios fueron seleccionados después de aplicar los criterios de inclusión y exclusión definidos. A continuación, se presentan las respuestas a las preguntas de investigación.

Respecto de los criterios de selección del conjunto de proyectos, las practicas más comunes realizadas por los investigadores en este sentido es seguir lineamientos propios para seleccionar los proyectos y utilizar una colección de proyectos existentes. No se ha evidenciado un marco unificado o automatizado para la selección de proyectos debido a la gran diversidad de aspectos considerados en los 94 estudios que reportaron criterios con un mayor nivel de detalle.

En 35 estudios se informaron criterios específicos del caso de estudio y solo en 5 casos se reportaron el uso de herramientas para automatizar el proceso de selección en base a las reglas establecidas.

Con respecto a las características de los proyectos recolectados, el lenguaje de programación principal de los proyectos software con un gran margen de diferencia es Java (79%), el segundo y tercero son C y C++ (31%) de 110 artículos que se registraron respuestas.

Los proyectos software utilizados fueron específicos con un 74% de aplicaciones de uso general y 96% de proyectos en el dominio del diseño y construcción de sistemas, servidores, redes, sistemas operativos o sistemas embebidos de los 96 artículos que fue posible determinar el tipo de software empleado.

Con respecto a los meta-datos de los proyectos, las fuentes de información de donde se extraen varían desde el mismo código fuente, la información en tiempo de ejecución, los reportes de defectos o el repositorio del proyecto, entre otras.

En particular se encontraron 54 estudios primarios que reúnen meta-datos extraídos del código de los proyectos software en forma de métricas. Estas miden aspectos como el tamaño, diseño y code smells del proyecto.

Respecto de las herramientas para obtener los meta-datos, en 44 estudios primarios se utilizan herramientas para la recolección de meta-datos de proyectos obtenidos por medio del análisis estático del código.

Realizan tareas como el cálculo de métricas de software, el análisis de la estructura y dependencias del código fuente, la detección de code smells o vulnerabilidades, la refactorización de código, la generación automática de tests y la detección patrones de diseño. Las herramientas que más se repitieron fueron Understand (8), Evo Suite (4), CKJM (3) y PMD (3).

De los análisis estadísticos desarrollados sobre los meta-datos, en 88 artículos se registraron tanto procedimientos estadísticos inferenciales como descriptivos. La mayoría de los métodos inferenciales fueron pruebas estadísticas no paramétricas (69%) y en menor medida paramétricas (25%). En el caso de los métodos descriptivos, se utilizaron tamaño de efecto, medidas de variabilidad y medidas de tendencia central.

Dicho esto, en un gran número de casos no se observa que se tome en consideración la forma de selección de las muestras en los análisis estadísticos practicados, ignorando la posibilidad de que las mismas no sean representativas de la población.

Finalmente, como aporte de este trabajo se identificaron algunas pautas que ayudan a sistematizar la selección de proyectos en la construcción de colecciones con fines de investigación. Las principales reglas son: código fuente libre tanto su acceso como distribución, la vigencia del proyecto, su popularidad en repositorios y en lenguaje Java.

La colección resultante debería conservar el código fuente de los proyectos, sus métricas obtenidas del análisis estático y los valores de estadísticos descriptivos que caractericen la muestra.

Como trabajo futuro se analizarán las colecciones creadas con fines de investigación presentes en la Tabla 6 y en artículos relacionados, con el objetivo de reconocer cuales fueron los criterios considerados en cada caso, el propósito de su construcción, y la vigencia de estos.

Agradecimientos

Este artículo cuenta con el apoyo de la Universidad Nacional del Nordeste (UNNE) a través del proyecto PI-17F018 “Metodologías y herramientas emergentes para contribuir con la calidad del software”, acreditado por la Secretaría de Ciencia y Técnica de la misma entidad para el periodo 2018-2021, y del Consejo Nacional de Investigaciones Científicas y Técnicas de la Argentina (CONICET) por medio de una beca interna doctoral otorgada por RESOL-2021-154-APN-DIR#CONICET para el período 2021-2025.

Referencias

1. **Albrecht, A. J., Gaffney, J. E. (1983).** Software function, source lines of code, and development effort prediction: A software

science validation. IEEE Transactions on Software Engineering, Vol. 9, No. 6, pp. 639–648. DOI: 10.1109/TSE.1983.235271.

2. **Ali, N. Bin, Petersen, K. (2014).** Evaluating strategies for study selection in systematic literature studies. International Symposium on Empirical Software Engineering and Measurement, pp. 1–4. DOI: 10.1145/2652 524.2652557.

3. **Allamanis, M., Sutton, C. (2013).** Mining source code repositories at massive scale using language modeling. 10th Working Conference on Mining Software Repositories MSR´13, pp. 207–216

4. **Bakır, A., Turhan, B., Bener, A. B. (2010).** A new perspective on data homogeneity in software cost estimation: A study in the embedded systems domain. Software Quality Journal, Vol. 18, No. 1, pp. 57–80. DOI: 10.10 07/s11219-009-9081-z.

5. **Barone, A. V. M., Sennrich, R. (2017).** A parallel corpus of Python functions and documentation strings for automated code documentation and code generation. http://ar xiv.org/abs/1707.02275.

6. **Boehm, B. W. (1981).** Software Engineering Economics. Springer Berlin Heidelberg.

7. **Chidamber, S. R., Kemerer, C. F. (1994).** A metrics suite for object oriented design. IEEE Transactions on Software Engineering, Vol. 20, No. 6

8. **Chidamber, S. R., Kemerer, C. F. (1991).** Towards a metrics suite for object oriented design. Conference Proceedings on Object-Oriented Programming Systems, Languages, and Applications OOPSLA'91, pp. 197–211. DOI: 10.1145/117954.117970.

9. **Cosentino, V., Luis, J., Izquierdo, C., Cabot, J. (2016).** Findings from GitHub: methods, datasets and limitations. Proceedings 13th Working Conference on Mining Software Repositories, MSR´16, pp. 137–141. DOI: 10.1145/2901739 .2901776.

10. **Crabtree, B. F., Miller, W. L. (1999).** Doing qualitative research 2nd ed. SAGE Publications, https://us.sagepub.com/enus/ nam/doing-qualitative -research/book9279

11. **D'Ambros, M., Lanza, M., Robbes, R. (2012).** Evaluating defect prediction approaches: A benchmark and an extensive comparison. Empirical Software Engineering, Vol. 17, No. 4–5, pp. 531–577. DOI: 10.1007/ s10664-011-9173-9.

12. **Desharnais, J. M. (1989).** Analyse statistique de la productivite des projets de developpement en informatique a partir de la technique des points de fonction. Masters Thesis University of Montreal

13. **Dyer, R., Nguyen, H. A., Rajan, H., Nguyen, T. N. (2015).** Boa: Ultra-large-scale software repository and source-code mining. ACM Transactions on Software Engineering and Methodology, Vol. 25, No. 1. DOI: 10.1145/ 2803171.

14. **Elmidaoui, S., Cheikhi, L., Idri, A. (2019).** Towards a taxonomy of software maintainability predictors. Advances in Intelligent Systems and Computing, Vol. 930, pp. 823–832. DOI: 10.1007/978-3-030-16181-1_77.

15. **Falessi, D., Smith, W., Serebrenik, A. (2017).** STRESS: A semi-automated, fully replicable approach for project selection. International Symposium on Empirical Software Engineering and Measurement, 2017-November, pp. 151–156. DOI: 10.1109/ ESEM.2017.22.

16. **Forward, A., Lethbridge, T. C. (2008).** A taxonomy of software types to facilitate search and evidence-based software engineering. Proceedings of the Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds, No. 14, pp. 179–191. DOI: 10.1145/1463788.1463807.

17. **Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D. (2002).** Refactoring: Improving the design of existing code. pp. 1–337

18. **FRAMEndeley. (2021).** Chrome web store. https://chrome.google.com/webstore/detail/fr amendeley/decpeaebklmmgfhnnhggeikfhhlbc jpf?hl=es.

19. **Garvin, D. A. (1984).** What does "Product quality" really mean? MIT Sloan Management Review, pp. 25–43. https://sloanreview.mit. edu/article/what-does-product-quality-really-mean/.

20. **Githut 2.0: A Small Place To Discover Languages In GITHUB. (2021).** GitHut. https://madnight.github.io/githut.

21. **Goeminne, M., Mens, T. (2015).** Towards a survival analysis of database framework usage in Java projects. IEEE International Conference on Software Maintenance and Evolution (ICSME), pp. 551–555. DOI: 10.1109/ICSM.2015.7332512.

22. **Hamasaki, K., Gaikovina Kula, R., Yoshida, N., Camargo Cruz, A. E., Fujiwara, K., Naist, H. I. (2013).** Who Does What during a Code Review? Datasets of OSS Peer Review Repositories. 10th Working Conference on Mining Software Repositories (MSR). http://source.android.com/

23. **Harrison, R., Counsell, S., Nithi, R. (1998).** Coupling metrics for object-oriented design. Proceedings Fifth International Software Metrics Symposium. Metrics (Cat. No.98TB100262), DOI: 10.1109/METRIC. 1998.731240.

24. **Herraiz, I., Izquierdo-Cortazar, D., Rivas-Hernandez, F., Gonzalez-Barahona, J., Robles, G., Dueñas-Domínguez, S., Garcia-Campos, C., Gato, J. F., Tovar, L. (2009).** Flossmetrics: Free / libre / open source software metrics. Proceedings of the European Conference on Software Maintenance and Reengineering, CSMR, pp. 281–284. DOI: 10.1109/CSMR.2009.43.

25. **Higgins, J. P. T., Thomas, J., Chandler, J., Cumpston, M., Li, T., Page, M. J., Welch, V. A. (2019).** Cochrane Handbook for systematic reviews of interventions. J. P. T. Higgins, J. Thomas, J. Chandler, M. Cumpston, T. Li, M. J. Page, & V. A. Welch (Eds.), Cochrane Handbook for Systematic Reviews of Interventions. Wiley. DOI: 10.1002/9781 119536604.

26. **Howison, J., Conklin, M., Crowston, K. (2006).** FLOSSmole: A collaborative repository for FLOSS research data and analyses. International Journal of Information Technology and Web Engineering, Vol. 1, No. 3, pp. 17–26. DOI: 10.4018/jitwe.200 6070102.

27. **Janssen, M., van der Voort, H. (2020).** Agile and adaptive governance in crisis response: Lessons from the COVID-19 pandemic. International Journal of Information Management, Vol. 55, pp. 102180. DOI: 10.1016/j.ijinfomgt.2020.102180.

28. **Jureczko, M., Madeyski, L. (2010).** Towards identifying software project clusters with regard to defect prediction. Promise '10, Towards Identifying Software Project Clusters With Regard to Defect Prediction, No. 9, pp. 1–10. DOI: 10.1145/1868328. 1868342.

29. **Keivanloo, I., Rilling, J., Zou, Y. (2014).** Spotting working code examples. Proceedings International Conference on Software Engineering, Vol. 1, pp. 664–675. DOI: 10.1145/2568225.2568292.

30. **Kemerer, C. F. (1987).** An empirical validation of software cost estimation models. Communications of the ACM, Vol. 30, No. 5, pp. 416–429. DOI: 10.1145/22899.22906.

31. **Kitchenham, B. A., Budgen, D., Brereton, P. O. (2011).** Using mapping studies as the basis for further research - A participant-observer case study. Information and Software Technology, Vol. 53, No. 6, pp. 638–651. DOI: 10.1016/j.infsof.2010.12.011.

32. **Kitchenham, B. A., Dybå, T., Jørgensen, M. (2004).** Evidence-based software engineering. Proceedings of the 26th International Conference on Software Engineering, pp. 273–281. www.evidencenetwork.org.

33. **Kitchenham, B., Brereton, P. (2013).** A systematic review of systematic review process research in software engineering. Information and Software Technology, Elsevier B.V., Vol. 55, No. 12, pp. 2049–2075. DOI: 10.1016/j.infsof. 2013.07.010.

34. **Kitchenham, B., Charters, S., (2007).** Guidelines for performing Systematic Literature Reviews in Software Engineering.pp. 1–57.

35. **Kitchenham, B., Kansala, K. (1993).** Inter-item correlations among function points. Proceedings First International Software Metrics Symposium, pp. 11–14. DOI: 10.1109/ METRIC.1993.263805.

36. **Kitchenham, B., Madeyski, L., Budgen, D., Keung, J., Brereton, P., Charters, S.,**

Gibbs, S., Pohthong, A. (2017). Robust statistical methods for empirical software engineering. Empirical Software Engineering, Vol. 22, No. 2, pp. 579–630. DOI: 10.1007/s10 664-016-9437-5.

37. **Kitchenham, B., Pfleeger, S. L. (1996).** Software quality: the elusive target. IEEE Software, Vol. 13, No. 1, pp. 12–21. DOI: 10.1 109/52.476281.

38. **Knuth, D. E. (1971).** An empirical study of FORTRAN programs. Software: Practice and Experience, Vol. 1, No. 2, pp. 105–133. DOI: 10.1002/spe.4380010203.

39. **Lehman, M. M. (1996).** Laws of software evolution revisited. In: Montangero, C. (eds) Software Process Technology. EWSPT 1996. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg. Vol 1149. pp. 108–124. DOI: 10.1007/BFb0017737.

40. **Lokan, C., Wright, T., Hill, P. R., Stringer, M. (2001).** Organizational benchmarking using the ISBSG data repository. IEEE Software, Vol. 18, No. 5, pp. 26–32. DOI: 10. 1109/52.951491.

41. **Maibaum, T., Wassyng, A. (2008).** A product-focused approach to software certification. Computer, Vol. 41, No. 2, pp. 91–93. DOI: 10.1109/MC.2008.37.

42. **Maxwell, K. (2002).** Applied Statistics for Software Managers.

43. **Mendes, E., Di Martino, S., Ferrucci, F., Gravino, C. (2008).** Cross-company vs. single-company web effort models using the Tukutuku database: An extended study. Journal of Systems and Software, Vol. 81, No. 5, pp. 673–690. DOI: 10.1016/j.jss.20 07.07.044.

44. **Miyazaki, Y., Terakado, M., Ozaki, K., Nozaki, H. (1994).** Robust regression for developing software estimation models. The Journal of Systems and Software, Vol. 27, No. 1, pp. 3–16. DOI: 10.1016/01641212(94) 90110-4.

45. **Mkaouer, W., Kessentini, M., Bechikh, S., Deb, K., Cinnéide, M. Ó. (2014).** High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III. GECCO´14 In: Proceedings of the 2014 Genetic and Evolutionary Computation Conference, pp. 1263–1270. DOI: 10.1145/2576768.2598366.

46. **Mockus, A. (2009).** Amassing and indexing a large sample of version control systems: Towards the census of public source code history. Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR´09, pp. 11–20. DOI: 10.1109/MSR.2009. 5069476.

47. **Molléri, J. S., Petersen, K., Mendes, E. (2016).** Survey guidelines in software engineering: An annotated review. Proceedings of the 10th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, No. 58, pp. 1–6. DOI: 10.1145/2961111.2962619

48. **Monperrus, M., Martinez, M. (2012).** CVS-Vintage: A Dataset of 14 CVS Repositories of Java Software. https://hal.archivesouver tes.fr/hal-0076 9121

49. **Orrú, M., Tempero, E., Marchesi, M., Tonelli, R., Destefanis, G. (2015).** A curated benchmark collection of python systems for empirical studies on software engineering. ACM International Conference Proceeding Series, pp. 1–4. DOI: 10.1 145/2810146. 2810148.

50. **Parnas, D. L. (2001).** Some software engineering principles. Software fundamentals: collected papers by David L. Parnas, pp. 257–266. DOI: 10.5555/376 584.376632.

51. **Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M. (2008).** Systematic Mapping Studies in Software Engineering. Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 68–77. www.splc.net.

52. **Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015).** Guidelines for conducting systematic mapping studies in software engineering: An update. Information and Software Technology, Vol. 64, pp. 1–18. DOI: 10.1016 /j.infsof.2015.03.007.

53. **Runeson, P., Höst, M. (2009).** Guidelines for conducting and reporting case study research in software engineering. Empirical Software

Engineering, Vol. 14, No. 2, pp. 131–164. DOI: 10.1007/s10664-008-9102-8.

54. **Saldaña, J. (2015).** The coding manual for qualitative researchers. SAGE Publications Ltd. https://us.sagepub.com/enus/nam/book/coding-manual-qualitative-researchers-1.

55. **Shepperd, M., Schofield, C. (1997).** Estimating software project effort using analogies. IEEE Transactions on Software Engineering, Vol. 23, No. 11, pp. 736–743. DOI: 10.1109/32.637387.

56. **Shepperd, M., Song, Q., Sun, Z., Mair, C. (2013).** Data quality: Some comments on the NASA software defect datasets. IEEE Transactions on Software Engineering, Vol. 39, No. 9, pp. 1208–1215. DOI: 10.1109/TSE.2013.11.

57. **Sheskin, D. J. (2000).** Parametric and nonparametric statistical procedures second Edition. www.crcpress.com

58. **Siegmund, J., Siegmund, N., Apel, S. (2015).** Views on internal and external validity in empirical software engineering. Proceedings - International Conference on Software Engineering, Vol. 1, pp. 9–19. DOI: 10.1109/ICSE.2015.24.

59. **Storey, M. A., Ernst, N. A., Williams, C., Kalliamvakou, E. (2020).** The who, what, how of software engineering research: a socio-technical framework. Empirical Software Engineering, Vol. 25, No. 5, pp. 4097–4129. DOI: 10.1007/s10664-020-09858-z.

60. **Tempero, E., Anslow, C., Dietrich, J., Han, T., Li, J., Lumpe, M., Melton, H., Noble, J. (2010).** The qualitas corpus: A curated collection of Java code for empirical studies. Proceedings - Asia-Pacific Software Engineering Conference, APSEC, pp. 336–345. DOI: 10.1109/APSEC.2010.46.

61. **Terra, R., Miranda, L. F., Valente, M. T., Bigonha, R. S. (2013).** Qualitas.class corpus. ACM SIGSOFT Software Engineering Notes, Vol. 38, No. 5, pp. 1–4. DOI: 10.1145/2507288.2507314.

62. **Unterkalmsteiner, M., Gorschek, T., Islam, A. K. M. M., Cheng, C. K., Permadi, R. B., Feldt, R. (2012).** Evaluation and measurement of software process improvement-A systematic literature review. IEEE Transactions on Software Engineering Vol. 38, No. 2, pp. 398–424. DOI: 10.1109/TSE.2011.26.

63. **Vasilescu, B., Serebrenik, A., Filkov, V. (2015).** A Data Set for Social Diversity Studies of GitHub Teams. MSR ’15: Proceedings of the 12th Working Conference on Mining Software Repositories. https://github.com/bvasiles/ght.

64. **Wohlin, C., Rainer, A. (2021).** Challenges and recommendations to publishing and using credible evidence in software engineering. Information and Software Technology, Vol. 134, pp. 106555. DOI: 10.1016/j.infsof.2021.106555.

65. **Yu, Y., Wang, H., Filkov, V., Devanbu, P., Vasilescu, B. (2015).** Wait for It: Determinants of pull request evaluation latency on GitHub. IEEE International Working Conference on Mining Software Repositories, pp. 367–371. DOI. 10.1109/MSR.2015.42.

66. **Zerouali, A., Mens, T. (2017).** Analyzing the Evolution of Testing Library Usage in Open Source Java Projects. IEEE 24th International Conference on Software Analysis, Evolution and Reengineering SANER, pp. 417–421.

67. **Zhang, L., Tian, J. H., Jiang, J., Liu, Y. J., Pu, M. Y., Yue, T. (2018).** Empirical research in software engineering — A literature survey. Journal of Computer Science and Technology, Vol. 33, No. 5, pp. 876–899. DOI: 10.1007/s11390-018-1864-x.

Referencias estudios primarios

P1. **Accioly, P., Borba, P., Cavalcanti, G. (2018).** Understanding semi-structured merge conflict characteristics in open-source Java projects. Empirical Software Engineering, Vol. 23, No. 4, pp. 2051–2085. DOI: 10.1007/s10664-017-9586-1.

P2. **Ahmed, I., Brindescu, C., Mannan, U. A., Jensen, C., Sarma, A. (2017).** An empirical examination of the relationship between code smells and merge conflicts. International Symposium on Empirical

Software Engineering and Measurement, pp. 58–67. DOI: 10.1109/ESEM.2017.12.

P3. **Ahmed, I., Mannan, U. A., Gopinath, R., Jensen, C. (2015).** An empirical study of design degradation: How software projects get worse over time. International Symposium on Empirical Software Engineering and Measurement, pp. 31–40. DOI: 10.1109/ESEM.2015. 7321186.

P4. **Ajienka, N., Capiluppi, A., Counsell, S. (2017).** Managing hidden dependencies in oo software: A study based on open source projects. International Symposium on Empirical Software Engineering and Measurement, pp. 141–150. DOI: 10.1109/ ESEM. 2017.21.

P5. **Al Alam, S. M. D., Shahnewaz, S. M., Pfahl, D., Ruhe, G. (2014).** Monitoring bottlenecks in achieving release readiness. International Symposium on Empirical Software Engineering and Measurement, pp. 1–4. DOI: 10.1145/2652 524.2652549.

P6. **Al Dallal, J., Morasca, S. (2014).** Predicting object-oriented class reuse-proneness using internal quality attributes. Empirical Software Engineering, Vol. 19, No. 4, pp. 775–821. DOI: 10.1007/s10664-0129239-3.

P7. **Allix, K., Bissyandé, T. F., Jérome, Q., Klein, J., State, R., Le Traon, Y. (2016).** Empirical assessment of machine learning-based malware detectors for Android: Measuring the gap between in-the-lab and in-the-wild validation scenarios. Empirical Software Engineering, Vol. 21, No. 1, pp. 183–211. DOI: 10.1007/s10664-014-9352- 6.

P8. **Alnaeli, S. M., Maletic, J. I., Collard, M. L. (2016).** An empirical examination of the prevalence of inhibitors to the parallelizability of open source software systems. Empirical Software Engineering, Vol. 21, No. 3, pp. 1272–1301. DOI: 10.1007/s10664-015-9385-5.

P9. **AlOmar, E. A., Mkaouer, M. W., Ouni, A., Kessentini, M. (2019).** On the impact of refactoring on the relationship between quality attributes and design metrics. International Symposium on Empirical Software Engineering and Measurement.

P10. **Aman, H., Amasaki, S., Sasaki, T., Kawahara, M. (2015).** Empirical analysis of change-proneness in methods having local variables with long names and comments. International Symposium on Empirical Software Engineering and Measurement, pp. 50–53. DOI: 10.1109/ESEM.2015. 7321197.

P11. **Aman, H., Sasaki, T., Amasaki, S., Kawahara, M. (2014).** Empirical analysis of comments and fault-proneness in methods: Can comments point to faulty methods? International Symposium on Empirical Software Engineering and Measurement.

P12. **Arcelli Fontana, F., Mäntylä, M. V., Zanoni, M., Marino, A. (2016).** Comparing and experimenting machine learning techniques for code smell detection. Empirical Software Engineering, Vol. 21, No. 3, pp. 1143–1191. DOI: 10.1007/s10664-015-9378-4.

P13. **Assar, S., Borg, M., Pfahl, D. (2016).** Using text clustering to predict defect resolution time: a conceptual replication and an evaluation of prediction accuracy. Empirical Software Engineering, Vol. 21, No. 4, pp. 1437–1475. DOI: 10.1007/s10664-015-9391-7.

P14. **Aué, J., Haisma, M., Tómasdóttir, K. F., Bacchelli, A. (2016).** Social diversity and growth levels of open source software projects on GitHub. International Symposium on Empirical Software Engineering and Measurement, pp. 1–6. DOI: 10.1145/2961111.2962633.

P15. **Bavota, G., De Lucia, A., Marcus, A., Oliveto, R. (2014).** Automating extract class refactoring: an improved method and its evaluation. Empirical Software Engineering, Vol. 19, No. 6, pp. 1617–1664. DOI: 10.1007/s10664-013-9256-x.

P16. **Beck, F., Diehl, S. (2013).** On the impact of software evolution on software clustering. Empirical Software Engineering, Vol. 18, No. 5, 970–1004. DOI: 10.1007/s10664-012-9225-9.

P17. **Behnamghader, P., Le, D. M., Garcia, J., Link, D., Shahbazian, A., Medvidovic, N. (2017).** A large-scale study of architectural evolution in open-source software systems. Empirical Software Engineering, Vol. 22, No. 3, 1146–1193. DOI: 10.1007/s10664-016-9466-0.

P18. **Behnamghader, P., Meemeng, P., Fostiropoulos, I., Huang, D., Srisopha, K., Boehm, B. (2018).** A scalable and efficient approach for compiling and analyzing commit history. International Symposium on Empirical Software Engineering and Measurement, No. 27, p.p. 1–10. DOI: 10.1145/3239235.3239237

P19. **Beller, M., Zaidman, A., Karpov, A., Zwaan, R. A. (2017).** The last line effect explained. Empirical Software Engineering, Vol. 22, No. 3, 1508–1536. DOI: 10.1007/ s10664-016-9489-6.

P20. **Bennin, K. E., Keung, J., Monden, A., Phannachitta, P., Mensah, S. (2017).** The significant effects of data sampling approaches on software defect prioritization and classification. International Symposium on Empirical Software Engineering and Measurement, pp. 364–373. DOI: 10.1109/ ESEM.2017.50.

P21. **Bibiano, A. C., Fernandes, E., Oliveira, D., Garcia, A., Kalinowski, M., Fonseca, B., Oliveira, R., Oliveira, A., Cedrim, D. (2019).** A Quantitative study on characteristics and effect of batch refactoring on code smells. International Symposium on Empirical Software Engineering and Measurement, pp. 1–11. DOI: 10.1109/ESEM.2019.8870183.

P22. **Biggers, L. R., Bocovich, C., Capshaw, R., Eddy, B. P., Etzkorn, L. H., Kraft, N. A. (2014).** Configuring latent Dirichlet allocation based feature location. Empirical Software Engineering, Vol. 19, No. 3, pp. 465–500. DOI: 10.1007/s106 64-012-9224- x.

P23. **Callaú, O., Robbes, R., Tanter, É., Röthlisberger, D. (2013).** How (and why) developers use the dynamic features of programming languages: The case of smalltalk. Empirical Software Engineering, Vol. 18, No. 6, pp. 1156–1194. DOI: 10.1007/s10664-012-9203-2.

P24. **Campos, E. C., Maia, M. D. A. (2017).** Common Bug-Fix Patterns: A large-scale observational study. International Symposium on Empirical Software Engineering and Measurement, pp. 404–413. DOI: 10.1109/ESEM.2017.55.

P25. **Ceccato, M., Capiluppi, A., Falcarin, P., Boldyreff, C. (2015).** A large study on the effect of code obfuscation on the quality of java code. Empirical Software Engineering, Vol. 20, No. 6, pp. 1486–1524. DOI: 10.10 07/s10664-014-9321-0.

P26. **Chen, B., Jack-Jiang, Z. M. (2017).** Characterizing logging practices in Java-based open source software projects – a replication study in Apache Software Foundation. Empirical Software Engineering, Vol. 22, No. 1, pp. 330–374. DOI: 10.1007/s10664-016-9429-5.

P27. **Chen, N., Hoi, S. C. H., Xiao, X. (2014).** Software process evaluation: a machine learning framework with application to defect management process. Empirical Software Engineering, Vol. 19, No. 6, pp. 1531–1564. DOI: 10.1007/s10664-013-9254-z.

P28. **Chen, X., Slowinska, A., Bos, H. (2016).** On the detection of custom memory allocators in C binaries. Empirical Software Engineering, Vol. 21, No. 3, pp. 753–777. DOI: 10.1007/s10664-015-9362-z.

P29. **Cheung, W. T., Ryu, S., Kim, S. (2016).** Development nature matters: An empirical study of code clones in JavaScript applications. Empirical Software Engineering, Vol. 21, No. 2, pp. 517–564. DOI: 10.1007/s10664-015-9368-6.

P30. **Coelho, R., Almeida, L., Gousios, G., van Deursen, A., Treude, C. (2017).** Exception handling bug hazards in Android: Results from a mining study and an exploratory survey. Empirical Software Engineering, Vol. 22, No. 3, pp. 1264–1304. DOI: 10.10 07/s10664-016-9443-7.

P31. **Corazza, A., Di Martino, S., Ferrucci, F., Gravino, C., Sarro, F., Mendes, E. (2013).** Using tabu search to configure support vector regression for effort estimation. Empirical Software Engineering, Vol. 18, No.

3, pp. 506–546. DOI: 10.1007/s10664-011-9187-3.

P32. Corazza, Anna, Di Martino, S., Maggio, V., Scanniello, G. (2016). Weighing lexical information for software clustering in the context of architecture recovery. Empirical Software Engineering, Vol. 21, No. 1, pp. 72–103. DOI: 10.1007/s10664-014-9347-3.

P33. da Costa, D. A., McIntosh, S., Kulesza, U., Hassan, A. E., Abebe, S. L. (2018). An empirical study of the integration time of fixed issues. Empirical Software Engineering, Vol. 23, No. 1, pp. 334–383. DOI: 10.1007/s10664-017-9520-6.

P34. Dashevskyi, S., Brucker, A. D., Massacci, F. (2019). A screening test for disclosed vulnerabilities in FOSS components. IEEE Transactions on Software Engineering, Vol. 45, No. 10, pp. 945–966. DOI: 10.1109/TSE. 2018.2816033.

P35. De O. Barros, M. (2014). An experimental evaluation of the importance of randomness in hill climbing searches applied to software engineering problems. Empirical Software Engineering, Vol. 19, No. 5, pp. 1423–1465. DOI: 10.1007/s10664-013-9294-4.

P36. Dit, B., Revelle, M., Poshyvanyk, D. (2013). Integrating information retrieval, execution and link analysis algorithms to improve feature location in software. Empirical Software Engineering, Vol. 18, No. 2, pp. 277–309. DOI: 10.1007/s10664-011-9194-4.

P37. Elish, M. O., Al-Ghamdi, Y. (2015). Fault density analysis of object-oriented classes in presence of code clones. International Conference on Evaluation and Assessment in Software Engineering, pp. 1–7. DOI: 10.1 145/2745802. 2745811.

P38. Eyolfson, J., Tan, L., Lam, P. (2014). Correlations between bugginess and time-based commit characteristics. Empirical Software Engineering, Vol. 19, No. 4, pp. 1009–1039. DOI: 10.1007/s10664-013-9245-0.

P39. Fagerholm, F., Guinea, A. S., Münch, J., Borenstein, J. (2014). The role of mentoring and project characteristics for onboarding in open source software projects. International Symposium on Empirical Software Engineering and Measurement, No. 55, pp. 1–10. DOI: 10.1145/2652524.2652540.

P40. Fan, Q., Yu, Y., Yin, G., Wang, T., Wang, H. (2017). Where Is the road for issue reports classification based on text mining? International Symposium on Empirical Software Engineering and Measurement, pp. 121–130. DOI: 10.1109/ESEM.2017.19.

P41. Foucault, M., Falleri, J. R., Blanc, X. (2014). Code ownership in open-source software. International Conference on Evaluation and Assessment in Software Engineering, pp. 1–9. DOI: 10.1145/26012 48.2601283.

P42. Fraser, G., Arcuri, A. (2015). 1600 faults in 100 projects: automatically finding faults while achieving high coverage with evosuite. Empirical Software Engineering, Vol. 20, No. 3, pp. 611–639. DOI: 10.1007/s10664-013-9288-2.

P43. Fraser, G., Arcuri, A. (2012). Sound empirical evidence in software testing. Proceedings International Conference on Software Engineering, pp. 178–188. DOI: 10.1109/IC SE.2012.6227195.

P44. Fu, S., Shen, B. (2015). Code bad smell detection through evolutionary data mining. International Symposium on Empirical Software Engineering and Measurement, pp. 1–9. DOI: 10.1109/ESEM.2015. 7321194.

P45. Gallaba, K., Mesbah, A., Beschastnikh, I. (2015). Don’t call us, we’ll call you: characterizing Callbacks in Javascript. International Symposium on Empirical Software Engineering and Measurement, pp. 1–10. DOI: 10.1109/ESEM.2015. 7321196.

P46. Gousios, G., Spinellis, D. (2014). Conducting quantitative software engineering studies with Alitheia Core. Empirical Software Engineering, Vol. 19, No. 4, pp. 885–925. DOI: 10.1007/ s10664-013-9242-3.

P47. **Haller, I., Slowinska, A., Bos, H. (2016**). Scalable data structure detection and classification for C/C++ binaries. Empirical Software Engineering, Vol. 21, No. 3, pp. 778–810. DOI: 10.1007/s10664-015-9363- y.

P48. **Hassan, F., Mostafa, S., Lam, E. S. L., Wang, X. (2017).** Automatic building of Java projects in software repositories: A study on feasibility and challenges. International Symposium on Empirical Software Engineering and Measurement, pp. 38–47. DOI: 10.1109/ESEM .2017.11.

P49. **He, Z., Peters, F., Menzies, T., Yang, Y. (2013).** Learning from open-source projects: An empirical study on defect prediction. International Symposium on Empirical Software Engineering and Measurement, pp. 45–54. DOI: 10.1109 /ESEM.2013.20.

P50. **Hebig, R., Derehag, J., Chaudron, M. R. V. (2015).** Identifying metrics' biases when measuring or approximating size in heterogeneous languages. International Symposium on Empirical Software Engineering and Measurement, pp. 1–4. DOI: 10.1109/ ESEM.2015.7321201.

P51. **Herzig, K., Just, S., Zeller, A. (2016).** The impact of tangled code changes on defect prediction models. Empirical Software Engineering, Vol. 21, No. 2, pp. 303–336. DOI: 10.1007/s10664-015-9376-6.

P52. **Hindle, A., Alipour, A., Stroulia, E. (2016).** A contextual approach towards more accurate duplicate bug report detection and ranking. Empirical Software Engineering, Vol. 21, No. 2, pp. 368–410. DOI: 10.1007/s10664-015-9387-3.

P53. **Hindle, A., Ernst, N. A., Godfrey, M. W., Mylopoulos, J. (2011).** Automated topic naming to support cross-project analysis of software maintenance activities. Proceedings of the 8th Working Conference on Mining Software Repositories, pp. 163–172. DOI: 10.1145/1985 441.1985466.

P54. **Hora, A., Robbes, R. (2020).** Characteristics of method extractions in Java: a large scale empirical study. Empirical Software Engineering, Vol. 25, pp. 1798–1833. DOI: 10.1007/s10664-020-09809-8.

P55. **Hunsen, C., Zhang, B., Siegmund, J., Kästner, C., Leßenich, O., Becker, M., Apel, S. (2016).** Preprocessor-based variability in open-source and industrial software systems: An empirical study. Empirical Software Engineering, Vol. 21, No. 2, pp. 449–482. DOI: 10.1007/s10664-015-9360-1.

P56. **Islam, M. R., Zibran, M. F., Nagpal, A. (2017).** Security vulnerabilities in categories of clones and non-cloned code: an empirical study. International Symposium on Empirical Software Engineering and Measurement, pp. 20–29. DOI: 10.1109/ES EM.2017.9.

P57. **Jaafar, F., Guéhéneuc, Y. G., Hamel, S., Khomh, F., Zulkernine, M. (2016).** Evaluating the impact of design pattern and anti-pattern dependencies on changes and faults. Empirical Software Engineering, Vol. 21, No. 3, pp. 896–931. DOI: 10.1007/s106 64-015-9361-0.

P58. **Kabinna, S., Bezemer, C. P., Shang, W., Syer, M. D., Hassan, A. E. (2018).** Examining the stability of logging statements. Empirical Software Engineering, Vol. 23, No. 1, pp. 290–333. DOI: 10.1007/s10664-017-9518-0.

P59. **Kagdi, H., Gethers, M., Poshyvanyk, D. (2013).** Integrating conceptual and logical couplings for change impact analysis in software. Empirical Software Engineering, Vol. 18, No. 5, pp. 933–969. DOI: 10.1007/s 10664-012-9233-9.

P60. **Kamei, Y., Fukushima, T., McIntosh, S., Yamashita, K., Ubayashi, N., Hassan, A. E. (2016).** Studying just-in-time defect prediction using cross-project models. Empirical Software Engineering, Voo. 21, No. 5, pp. 2072–2106. DOI: 10.1007/s106 64-015-9400-x.

P61. **Khatibi Bardsiri, V., Jawawi, D. N. A., Hashim, S. Z. M., Khatibi, E. (2014).** A flexible method to estimate the software development effort based on the classification of projects and localization of comparisons. Empirical Software

Engineering, Vol. 19, No. 4, pp. 857–884. DOI: 10.1007/ s10664-013-9241-4.

P62. **Kifetew, F. M., Tiella, R., Tonella, P. (2017).** Generating valid grammar-based test inputs by means of genetic programming and annotated grammars. Empirical Software Engineering, Vol. 22, No. 2, pp. 928–961. DOI: 10.1007/s10664-015-9422-4.

P63. **Kim, S., Kim, D. (2016).** Automatic identifier inconsistency detection using code dictionary. Empirical Software Engineering, Vol. 21, No. 2, pp. 565–604. DOI: 10.1007/s 10664-015-9369-5.

P64. **Kula, R. G., German, D. M., Ouni, A., Ishio, T., Inoue, K. (2018).** Do developers update their library dependencies? An empirical study on the impact of security advisories on library migration. Empirical Software Engineering, Vol. 23, No. 1, pp. 384–417. DOI: 10.1007/s10664-017-9521-5.

P65. **Lavazza, L., Morasca, S. (2016).** Identifying thresholds for software faultiness via optimistic and pessimistic estimations. International Symposium on Empirical Software Engineering and Measurement, pp. 1–10. DOI: 10.1145/2961 111.2962595.

P66. **Lee, T., Gu, T., Baik, J. (2014).** MND-SCEMP: An empirical study of a software cost estimation modeling process in the defense domain. Empirical Software Engineering, Vol. 19, No. 1, pp. 213–240. DOI: 10.1007/s10664-012-9220-1.

P67. **Linares-Vásquez, M., McMillan, C., Poshyvanyk, D., Grechanik, M. (2014).** On using machine learning to automatically classify software applications into domain categories. Empirical Software Engineering, Vol. 19, No. 3, 582–618. DOI: 10.1007/s 10664-012-9230-z.

P68. **Lokan, C., Mendes, E. (2017).** Investigating the use of moving windows to improve software effort prediction: a replicated study. Empirical Software Engineering, Vol. 22, No. 2, pp, 716–767. DOI: 10.1007/s10664-016-9446-4.

P69. **Malhotra, R., Khanna, M. (2017).** An empirical study for software change prediction using imbalanced data. Empirical Software Engineering, Vol. 22, No. 6, pp. 2806–2851. DOI: 10.1007/s10664-016-9488-7.

P70. **Malloy, B. A., Power, J. F. (2017).** Quantifying the transition from python 2 to 3: An empirical study of python applications. International Symposium on Empirical Software Engineering and Measurement, pp. 314–323. DOI: 10.1109/ ESEM.2017.45.

P71. **Martinez, M., Monperrus, M. (2013).** Mining software repair models for reasoning on the search space of automated program fixing. Empirical Software Engineering, Vol. 20, No. 1, pp. 176–205. DOI: 10.1007/s106 64-013-9282-8.

P72. **Mayer, P., Bauer, A. (2015).** An empirical analysis of the utilization of multiple programming languages in open source projects. International Conference on Evaluation and Assessment in Software Engineering, No. 4, pp. 1–10. DOI: 10.1145/ 2745802.2745805.

P73. **McIlroy, S., Ali, N., Khalid, H., E. Hassan, A. (2016).** Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews. Empirical Software Engineering, Vol. 21, No. 3, pp. 1067–1106. DOI: 10.1007/s10664-015-9375-7.

P74. **McIntosh, S., Kamei, Y., Adams, B., Hassan, A. E. (2016).** An empirical study of the impact of modern code review practices on software quality. Empirical Software Engineering, Vol. 21, No. 5, pp. 2146–2189. DOI: 10.1007/s10664-015-9381-9.

P75. **McIntosh, S., Nagappan, M., Adams, B., Mockus, A., Hassan, A. E. (2015).** A large-scale empirical study of the relationship between build technology and build maintenance. Empirical Software Engineering, Vol. 20, No. 6, pp. 1587–1633. DOI: 10.1007/s10664-014-9324-x.

P76. **Medeiros, F., Lima, G., Amaral, G., Apel, S., Kästner, C., Ribeiro, M., Gheyi, R. (2019).** An investigation of misunderstanding code patterns in C open-source software projects. Empirical Software Engineering, Vol. 24, No. 4, pp.

1693–1726. DOI: 10.1007/s10664-018-9666-x.

P77. **Misirli, A. T., Shihab, E., Kamei, Y. (2016).** Studying high impact fix-inducing changes. Empirical Software Engineering, Vol. 21, No. 2, pp. 605–641. DOI: 10.1007/s10664-015-9370-z.

P78. **Mkaouer, M. W., Kessentini, M., Bechikh, S., Ó Cinnéide, M., Deb, K. (2016).** On the use of many quality attributes for software refactoring: a many-objective search-based software engineering approach. Empirical Software Engineering, Vol. 21, No. 6, pp. 2503–2545. DOI: 10.1007/s10664-015-9414-4.

P79. **Mkaouer, M. W., Kessentini, M., Cinnéide, M., Hayashi, S., Deb, K. (2017).** A robust multi-objective approach to balance severity and importance of refactoring opportunities. Empirical Software Engineering, Vol. 22, No. 2, pp. 894–927. DOI: 10.1007/s10664-016-9426-8.

P80. **Morasca, S., Lavazza, L. (2019).** Comparing the effectiveness of using design and code measures in software faultiness estimation. International Conference on Evaluation and Assessment in Software Engineering, pp. 112–121. DOI: 10.114 5/3319008.3319026.

P81. **Morasca, S., Lavazza, L. (2016).** Slope-based fault-proneness thresholds for software engineering measures. International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10. DOI: 10.1145/29 15970.2915997.

P82. **Mori, T., Tamura, S., Kakui, S. (2013).** Incremental estimation of project failure risk with Naïve bayes classifier. International Symposium on Empirical Software Engineering and Measurement, pp. 283–286. DOI: 10.1109/ES EM.2013.40.

P83. **Munaiah, N., Kroh, S., Cabrey, C., Nagappan, M. (2017).** Curating GitHub for engineered software projects. Empirical Software Engineering, Vol. 22, No. 6, pp. 3219–3253. DOI: 10.1007/s10664-017-9512-6.

P84. **Cinnéide, M. O., Hemati-Moghadam, I., Harman, M., Counsell, S., Tratt, L. (2017).** An experimental search-based approach to cohesion metric evaluation. Empirical Software Engineering, Vol. 22, No. 1, pp. 292–329. DOI: 10.1007/s10664-016-9427- 7.

P85. **Okutan, A., Yıldız, O. T. (2014).** Software defect prediction using Bayesian networks. Empirical Software Engineering, Vol. 19, No. 1, pp. 154–181. DOI: 10.1007/s10664-012-9218-8.

P86. **Parnin, C., Bird, C., Murphy-Hill, E. (2013).** Adoption and use of Java generics. Empirical Software Engineering, Vol. 18, No. 6, pp. 1047–1089. DOI: 10.1007/s10664-012-9236-6.

P87. **Parsai, A., Murgia, A., Demeyer, S. (2016).** Evaluating random mutant selection at class-level in projects with non-adequate test suites. International Conference on Evaluation and Assessment in Software Engineering, No. 11, pp. 1–10. DOI: 10.1145 /2915970.2915992.

P88. **Petrić, J., Bowes, D., Hall, T., Christianson, B., Baddoo, N. (2016).** Building an ensemble for software defect prediction based on diversity selection. International Symposium on Empirical Software Engineering and Measurement, 08-09-Sept, No. 46, pp. 1–10. DOI: 10.1145/ 2961111. 2962610.

P89. **Petrić, J., Bowes, D., Hall, T., Christianson, B., Baddoo, N. (2016).** The Jinx on the NASA software defect data sets. International Conference on Evaluation and Assessment in Software Engineering, No. 13, pp. 1–5. DOI: 10.1145/2915970.2 916007.

P90. **Petrić, J., Galinac Grbac, T. (2014).** Software structure evolution and relation to system defectiveness. International Conference on Evaluation and Assessment in Software Engineering, No. 34, pp. 1–10. DOI: 10.1145/2 601248.2601287.

P91. **Phannachitta, P., Keung, J., Monden, A., Matsumoto, K. (2017).** A stability assessment of solution adaptation techniques for analogy-based software effort estimation. Empirical Software Engineering, Vol. 22, No. 1, pp. 474–504. DOI: 10.1007/s10664-016-9434-8.

P92. **Phannachittay, P., Mondeny, A., Keungz, J., Matsumotoy, K. (2015).** Case consistency: A necessary data quality property for software engineering data sets. International Conference on Evaluation and Assessment in Software Engineering, No. 19, pp. 1–10. DOI: 10.1145/274580 2.2745820.

P93. **Quesada-López, C., Jenkins, M. (2014).** Function point structure and applicability validation using the ISBSG dataset: A replicated study. International Symposium on Empirical Software Engineering and Measurement, No. 66, pp.1. DOI: 10.1145/2652524.2652595.

P94. **Raja, U. (2013).** All complaints are not created equal: Text analysis of open source software defect reports. Empirical Software Engineering, Vol. 18, No. 1, pp. 117–138. DOI: 10.1007/s10 664-012-9197-9.

P95. **Rakha, M. S., Shang, W., Hassan, A. E. (2016).** Studying the needed effort for identifying duplicate issues. Empirical Software Engineering, Vol. 21, No. 5, pp. 1960–1989. DOI: 10.1007/s10664-015-9404-6.

P96. **Ramírez, A., Romero, J. R., Ventura, S. (2016).** A comparative study of many-objective evolutionary algorithms for the discovery of software architectures. Empirical Software Engineering, Vol. 21, No. 6, pp. 2546–2600. DOI: 1007/s10664-015-9399-z.

P97. **Rodriguez, D., Herraiz, I., Harrison, R., Dolado, J., Riquelme, J. C. (2014).** Preliminary comparison of techniques for dealing with imbalance in software defect prediction. International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10. DOI: 10.1145/2601248.2601294.

P98. **Rodriguez, I., Wang, X. (2017).** An empirical study of open source virtual reality software projects. In: International Symposium on Empirical Software Engineering and Measurement, pp. 474–475. DOI: 10.1109/ESEM.2017.65.

P99. **Rojas, J. M., Vivanti, M., Arcuri, A., Fraser, G. (2017).** A detailed investigation of the effectiveness of whole test suite generation. Empirical Software Engineering, Vol. 22, No. 2, pp. 852–893. DOI: 10.1007 /s10664-015-9424-2.

P100. **Rosa, W., Madachy, R., Boehm, B., Clark, B. (2014).** Simple empirical software effort estimation model. International Symposium on Empirical Software Engineering and Measurement, No. 43, pp. 1–4. DOI: 10.1145 /2652524.2652558.

P101. **Ryu, D., Choi, O., Baik, J. (2016**). Value-cognitive boosting with a support vector machine for cross-project defect prediction. Empirical Software Engineering, Vol. 21, No. 1, pp. 43–71. DOI: 10.1007/s10664014 9346-4.

P102. **Sawant, A. A., Bacchelli, A. (2017).** fine-GRAPE: fine-grained APi usage extractor – an approach and dataset to investigate API usage. Empirical Software Engineering, Vol. 22, No. 3, pp. 1348–1371. DOI: 10.1007/s10 664-016-9444-6.

P103. **Scanniello, G., Marcus, A., Pascale, D. (2015).** Link analysis algorithms for static concept location: an empirical assessment. Empirical Software Engineering, Vol. 20, No. 6, pp. 1666–1720. DOI: 10.1007/s10664-014-9327-7.

P104. **Scholtes, I., Mavrodiev, P., Schweitzer, F. (2016).** From Aristotle to Ringelmann: a large-scale analysis of team productivity and coordination in Open Source Software projects. Empirical Software Engineering, Vol. 21, No. 2, pp. 642–683. DOI: 10.1007/ s10664-015-9406-4.

P105. **Sharma, T., Fragkoulis, M., Spinellis, D. (2017).** House of cards: code smells in open-source c# repositories. International Symposium on Empirical Software Engineering and Measurement, pp. 424–429. DOI: 10.1109/ ESEM.2017.57.

P106. **Shihab, E., Ihara, A., Kamei, Y., Ibrahim, W. M., Ohira, M., Adams, B., Hassan, A. E., Matsumoto, K. I. (2013).** Studying reopened bugs in open source software. Empirical Software Engineering, Vol. 18, No. 5, PP. 1005–1042. DOI: 10.1007/s10664-012-9228-6.

P107. Shippey, T., Hall, T., Counsell, S., Bowes, D. (2016). So you need more method level datasets for your software defect prediction? Voilà! International Symposium on Empirical Software Engineering and Measurement, pp. 1–6. DOI: 10.1145/29611 11.2962620.

P108. Soetens, Q. D., Demeyer, S., Zaidman, A., Pérez, J. (2016). Change-based test selection: an empirical evaluation. Empirical Software Engineering, Vol. 21, No. 5, pp. 1990–2032. DOI: 10.1007/s10664-015-9405-5.

P109. Tan, L., Liu, C., Li, Z., Wang, X., Zhou, Y., & Zhai, C. (2014). Bug characteristics in open source software. Empirical Software Engineering, Vol. 19, No. 6, pp. 1665–1705. DOI: 10.1007/s10664-013-9258-8.

P110. Thomas, S. W., Hemmati, H., Hassan, A. E., Blostein, D. (2014). Static test case prioritization using topic models. Empirical Software Engineering, Vol. 19, No. 1, pp. 182–212. DOI: 10.1007/s10664-012-9219- 7.

P111. Thongtanunam, P., McIntosh, S., Hassan, A. E., Iida, H. (2017). Review participation in modern code review: An empirical study of the android, Qt, and OpenStack projects. Empirical Software Engineering, Vol. 22, No. 2, pp. 768–817. DOI: 10.1007/s10664-016-9452-6.

P112. Tian, Y., Ali, N., Lo, D., Hassan, A. E. (2016). On the unreliability of bug severity data. Empirical Software Engineering, Vol. 21, No. 6, pp. 2298–2323. DOI: 10.1007/ s10664-015-9409-1.

P113. Vidal, S. A., Bergel, A., Marcos, C., Díaz-Pace, J. A. (2016). Understanding and addressing exhibitionism in Java empirical research about method accessibility. Empirical Software Engineering, Vol. 21, No. 2, pp. 483–516. DOI: 10.1007/s10664-015-9365-9.

P114. Walkinshaw, N., Minku, L. (2018). Are 20% of files responsible for 80% of defects? International Symposium on Empirical Software Engineering and Measurement. No. 2, pp. 1–10. DOI: 10.1145/3239235.3 239244.

P115. Wood, M. I., Ivanov, L., Lamprou, Z. (2019). An analysis of inheritance hierarchy evolution. International Conference on Evaluation and Assessment in Software Engineering, pp. 24–33. DOI: 10.1145/3319 008.3319023.

P116. Wu, D., Chen, L., Zhou, Y., Xu, B. (2015). An empirical study on C++ concurrency constructs. International Symposium on Empirical Software Engineering and Measurement, pp. 257–266. DOI: 10.1109/ ESEM.2015.7321187.

P117. Xia, X., Shihab, E., Kamei, Y., Lo, D., Wang, X. (2016). Predicting crashing releases of mobile applications. International Symposium on Empirical Software Engineering and Measurement, pp. 1–10. DOI: 10.1145/29611 11.2962606.

P118. Yan, M., Fang, Y., Lo, D., Xia, X., Zhang, X. (2017). File-level defect prediction: unsupervised vs. Supervised models. International Symposium on Empirical Software Engineering and Measurement, pp. 344–353. DOI: 10.1109/ESEM.2017.48.

P119. Zhang, F., Mockus, A., Keivanloo, I., Zou, Y. (2016). Towards building a universal defect prediction model with rank transformed predictors. Empirical Software Engineering, Vol. 21, No. 5, pp. 2107–2145. DOI: 10.1007/s10664-015-9396-2.

P120. Zhao, Y., Zhang, F., Shihab, E., Zou, Y., Hassan, A. E. (2016). How are discussions associated with bug reworking? An empirical study on open source projects. International Symposium on Empirical Software Engineering and Measurement, pp. 1–10. DOI: 10.1145/296 1111.2962591.

P121. Zhou, B., Neamtiu, I., Gupta, R. (2015). A cross-platform analysis of bugs and bug-fixing in open source projects: Desktop vs. Android vs. iOS. In: International Conference on Evaluation and Assessment in Software Engineering, pp. 1–10. DOI: 10.1145/2745802.2745808.

P122. Zhu, J., Zhou, M., Mockus, A. (2014). Patterns of folder use and project popularity: A case study of github repositories. International Symposium on Empirical Software Engineering and Measurement. No. 30, pp. 1–4. DOI: 10.11 45/2652524.2 652564.

Article received on 31/05/2021; accepted on 18/08/2022. Corresponding author is Juan Andrés Carruther.

ISSN 2007-9737

Big Medical Image Analysis: Alzheimer’s Disease Classification Using Convolutional Autoencoder

Padmini Mansingh[1], Binod Kumar Pattanayak[1], Bibudhendu Pati[2]

[1] Deemed to be University,
Department of Computer Science and Engineering,
Institute of Technical Education and Research Siksha 'O' Anusandhan,
Bhubaneswar, Odisha,
India

[2] Ramadevi Women's University,
Department of Computer Science,
Bhubaneswar, Odisha,
India

padminimansingh, patibibudhendu @gmail.com,
binodpattanayak@soa.ac.in

Abstract. Deep learning-based analysis is a noticeable topic in recent years. The enormous success of deep learning is now combined with big data analytics to provide an open platform to the healthcare industry for a better diagnosis of any disease. In this paper, we described the convolutional autoencoder technique that reduces the complexity of radiologists through a brief study of Alzheimer's MRI data, which led to a rise in data-driven medical research for a better diagnosis. In this research, we have compared the effects of two techniques: convolutional autoencoder (CANN) and independent component analysis (ICA), and discovered that CANN has a higher accuracy of 99.42% and outperforms ICA models in terms of convergence speed.

Keywords. Deep learning, big data analytics, CANN, ICA, healthcare, machine learning.

1 Introduction

The vast amount of data (big data) produced within the healthcare industry is analyzed to provide better healthcare services to patients. The analytical system of big data is designed with very efficient integrated technology. Healthcare is one of the many big data applications that use both big data analytics and a relatively new well-known technology called deep learning to provide high-quality treatment to patients. Data gathered for healthcare informatics comes from different modalities MRI, fMRI, SPECT, PET, CT, and DTI. Using various deep learning algorithms with large amounts of healthcare data allows the healthcare industry to make more informed and faster decisions.

Recently MRI has been extensively utilized for the diagnosis of different diseases. This study aims to find the link between deep learning and the diagnosis of Alzheimer's patients.

Alzheimer's disease (AD) is the most common cause of dementia, and it is one of the currently leading diseases in most countries.

According to NIA, it will exceed the rate by 16 million in 2050 [1]. The unexpected growth of AD creates a huge economic breakdown for countries like USA and UK. According to the Alzheimer Association, 5.5 million Americans suffered from AD at the age of 65 or above.

Alzheimer's disease is named after Dr.Alois Alzheimer in1906 [2]. It is not a genetic disease but it happens due to two abnormal clumps called Amyloid plaques and tangles called tau protein. In the mid-60s, doctors found certain aspects of cognition, such as weak thinking power, behavioral changes, poor visuospatial ability, etc in an AD patient. By performing brain scans, such

doi: 10.13053/CyS-26-4-4090

as MRI through effective radioglaciology approaches, a huge amount of imaging data is generated because each patient has thousands of medical images.

Due to this Big Data analytics was introduced with different characteristics. Here volume is one of the characteristics of big data managed and analyzed by different tools of both big data analytics and deep learning with its great advantages.

Here Fig.1 shows a sagittal view of Alzheimer's pre-processed scan by Keras. In this paper, we proposed a novel method for the classification of Alzheimer's (AD) vs mild cognitive impairment in a prodromal stage of AD vs (MCI) vs healthy control (HC) by deep learning architecture. Deep learning tools have been applied to medical images for computer-aided pathology detection, classification, and prediction.

In recent, deep learning tools have been shown highly effective for a broad range of big healthcare analytics, multiple hidden layers are placed in between input and output layers for better analysis in diagnosing AD patients. For feature extraction, several hidden layers are placed, and each next layer contains the feature of the previous layer.

This paper pays close attention to the convolutional autoencoder for the diagnosis of AD. Combining both the approaches of CNN and Autoencoder for feature extraction of AD is our main motive behind this research. In this paper, we proposed a convolutional autoencoder deep learning for unsupervised feature learning to improve accuracy.

The remaining section is organized as follows. Section 2 contains an elaborative description of related works. Section 3 describes the convolutional neural network. Section 4 describes the proposed convolutional autoencoder (CANN) model. Section 5 elaborates on the comparisons between the two techniques of convolutional autoencoder (CANN) and Independent component analysis (ICA). Section 6 presents the experiment and result in part. Section 7 elaborates the classification part. Section 8 gives the discussion part and the last section contributes the conclusion part.

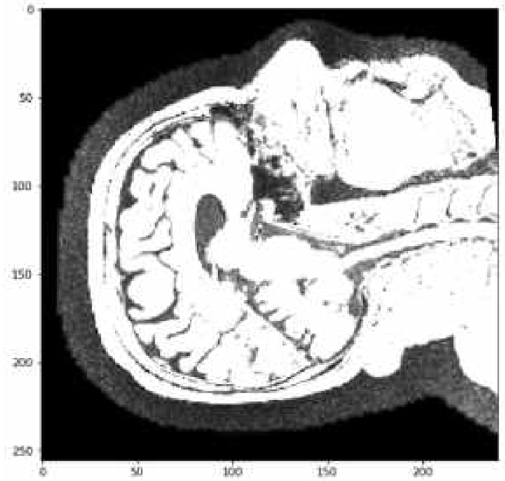


Fig. 1. Sagittal view of Alzheimer's pre-processed scans

2 Related Work

To classify patterns, selecting a subset of an input variable from a whole set of the learning algorithm is the motivation for feature extraction:

$\{f_1, f_2, f_3, fu.......f_m\} \rightarrow \{f_{ij}...f_{uv},..,f_{un}\},$

$uv \in \{1,..., m\},$

$v=1,..., n.$

Feature selection makes more accuracy a model if the feature is very large. By overcoming traditional methods of feature selection algorithms like filter method, wrapper method, and embedded methods, deep learning approaches were introduced to get a higher accuracy through big healthcare data.

The manual approaches of radiologists are more complex to degenerate features for better detection. A large amount of unlabeled data is used to learn the features in unsupervised learning approaches.

Convolutional neural network (CNN) has been creating good research on text analysis, image detection, speech recognition, etc. in the field of deep learning. Many researchers have shown that CNN is better to extract features as compared to the traditional method like SVM [3] finding the classification of skin concern with the deep neural network.

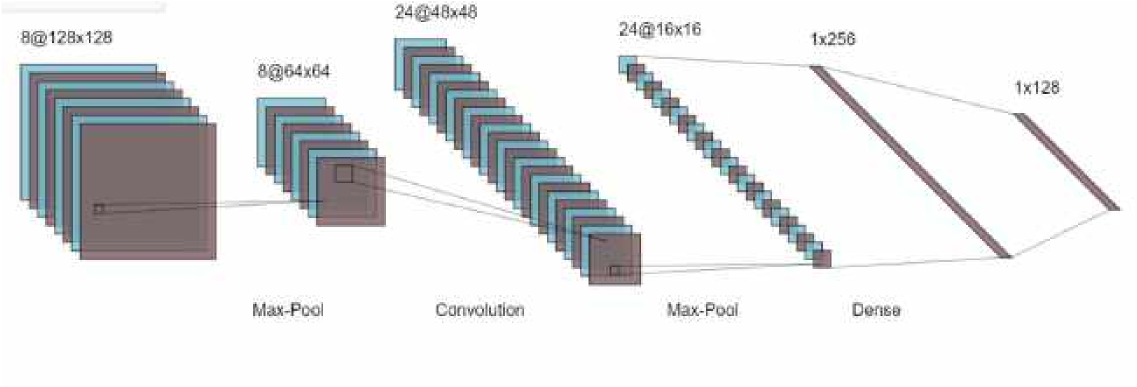


Fig. 2. The brief convolutional architecture

Unsupervised methods are an effective research area for using SAEs to classify AD & MCI [4]. For classification of AD using a whole-brain hierarchical network [5] was very well defined in their research, paper.AD classified the AD via a deep convolutional neural network using MRI & fMRI data [6] is also a landmark for feature study.

3 Convolutional Neural Network (CNN)

Image segmentation is becoming a critical/different challenge for clinical researchers using machine-learning technologies. As a result, various distinct works in the medical area are currently giving a location for research to overcome the difficulty of high dimensionality data through segmentation.

Despotovic et al. [7] provided a substantial review of the different segmentation techniques. All proposed segmentation technique deploys on evolutionary algorithms. Diverse strategies are presented to briefly define the task to increase performance based on the different anatomy and functions of medical images.

In addition, using deep learning to combine high-dimensional data, such as massive medical data, different feature selection/extraction or to generate disease characteristics for a better diagnosis is possible.

The training model for medical imaging classification typically takes one or more images as inputs and outputs yes or no.

CNN was first used in image analysis in 2012 as a faster and more accurate model. The real definition of convolution, in general, is the integral of the product of two signals:

$$\mathrm{h(t)*k(t)}=\int_{-\infty}^{\infty} h(\tau) * k(t-\tau)d\tau.$$

The different layer of CNN is the convolutional layer, rectified linear unit (RELU) layer, and pooling layer are described in Fig 1.

To extract text characteristics or procure the features of any disease is simpler for researchers by using CNN techniques.

Here Fig. 2 briefly describe the different layer with the proper equation.

3.1 Convolutional Layer

The VGG (visual geometry Group) has established a track for classification. The whole-brain image is fed into CNN, which uses hidden layers to extract useful characteristics. The first function contains input values as pixel values, whereas the second function contains the kernel as a filter.

The dot product among two functions gives output as another function, and the filter is shifted over the image from one position to another is called stride length for detecting real features.

The feature map (activation) is captured by shifting the operation by covering the entire image. The CNN has a limited connection to the next layer for meaningful feature detection. The '*' is denoting the convolutional operation. Other notation is as following.

Output feature map=I(t).

I(t)=h(t)*k(t)=(h*k) (t):

I: denoted as output or feature map.

t: denoted as integer.

h: denoted as input.

k: denoted filter or kernel.

*: convolutional operation.

Discretized convolutional formula:

I(t)=$\sum_b i(b) * k(t-b)$.

For two dimensions:

I(t)=$\sum_a \sum_b i(a,b) * k(u-a, v-b)$.

3.2 RELU Layer

In the Relu layer, replace the - ve values with 0 for avoiding the vanishing gradient problem.

P(x)=max (0, x).

x→input to the neuron.

Among all the activation functions, Relu is one of them to maintain the CNN operation.

3.3 Pooling Layer

For extracting dominant features, it is used by reducing the parameters and computation in the network. There are two types of pooling here: max pooling and avg pooling.

3.4 Fully Connected Layer

The last layer fully connected is a feed-forward neural network. Here, every neuron from the preceding layer is connected to the next layer. The fully connected layer is created by learning possibly nonlinear functions of high-level features.

4 Proposed Convolutional Autoencoder Model

To build a better model, MRI images are very useful for the classification of medical data. From a supervised point of view, weights are updated through forwarding and backward propagation. Standard autoencoder is the artificial neural network and for the unsupervised convolutional filter, convolutional autoencoder (CAE)is a tool that learns to encode input and tries to reconstruct the input by decoding layer.

The raw AD MRI patches are input to the convolutional autoencoder CANN for feature learning and the explicit parameters are determined by autoencoder's unsupervised learning. CNN is then classified by classification technique.

CAE is the type of CNN, where filters are learning and aim to classify the inputs. At first, it is the autoencoder (AE) that uses both the convolutional and pooling layer to extract the hidden pattern of input features. In the decoding phase, deconvolutional and unpooling for reconstructing the features from the hidden pattern. Specifically, the different patches of AD MRI images can be denoted as:

U∈ U.

U⊂ R^{n*l*l}.

n→ no. of input channels.

l*l→ size of input images.

unlabeled dataset=UD--= {u | u∈ U}.

4.1 Autoencoder

Specifically, the standard autoencoder is based on a supervised learning approach. If we explained the autoencoder technique, our aim of this paper is fully solved.

The Autoencoder technique is a type of artificial neural network (ANN), which is used for encoding the data for dimensionality reduction in an unsupervised and supervised manner.

In the supervised approach, we updated the connection weight through forwarding and backward propagation theory.

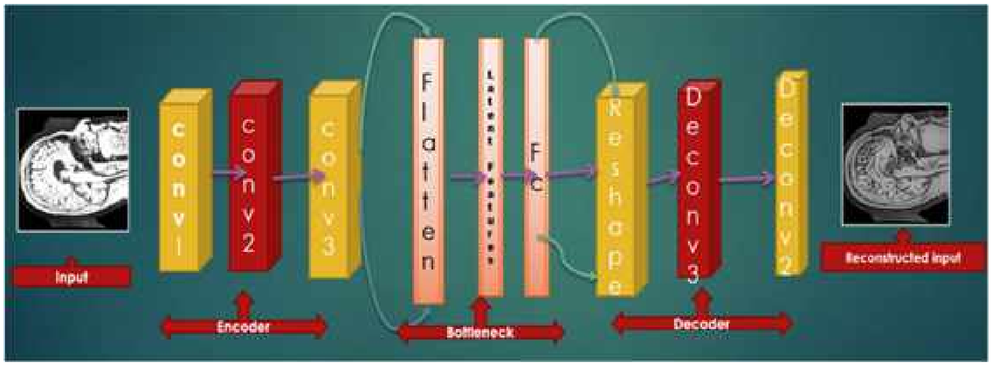


Fig. 3. Convolutional autoencoder architecture

For learning generative models, the unsupervised approaches are accepted with unlabeled data as input to a model. Reconstructing input data from execrating features of the output data is the process of the autoencoder. Here, the output layer has the same no of neurons as compared to the input layer. An autoencoder consists of two parts:

1 Encoder,

2 Decoder.

There is a convolutional filter for every convolutional layer with depth D. The cost function reaches its optimality, after many times of iterations:

k→ input data.

$k \in R^n$.

n→dimension vector.

n→output data .

$h \in R^m$.

m→dimension vector.

Image extraction, compression, denoising, and dimensionality reduction are the application of autoencoder techniques.

4.1.1 Encoding Stage

By converting input k into h of the hidden layer by:

$h=f(k)=\alpha(wk+b)$.

h→ referred to as code or latent representation.

A→activation function such as sigmoid function on a rectified linear unit.

b→bias vector.

w→weight matrix.

4.1.2 Decoding Stage

The autoencoder maps h to reconstruct h^1 of the same shape as h.

$h^1=f^1(h)=\alpha^1(w^1\,h+b^1)$

α^1→activation function.

Autoencoders are trained to minimize the squared root error:

$L\,(k, h^1) =||k\text{-}h^1||^2=||k\text{-}\alpha^1(w^1(\alpha(wk+b)) +b^1||^2$.

Reconstruction error minimization or loss is trained by an autoencoder.

4.2 Convolutional Autoencoder

In an autoencoder, the neural network of input is the same as the output. As such, it is a part of unsupervised learning. Here, conversion from feature maps input to output or reconstructing the original input from the lower dimension is called a decoder. In addition, transforming the input to an encoding form is called an encoder. Therefore, the model shows how to compress the data by minimizing information loss. The step-by-step layer of the convolutional autoencoder is described in the following mathematical formulas:

f (*)→convolutional encoder operation.

f^1(*)→convolutional decoder operation.

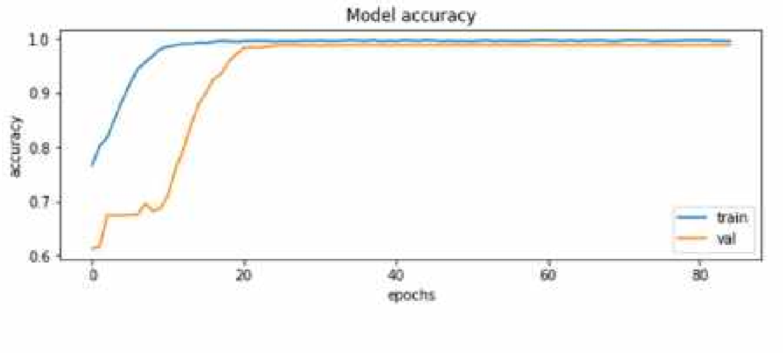


Fig. 4. Accuracy plot for CANN

Table 1. Classification performance of CANN

Methods	AUC	Precision	Loss	Accuracy
CANN	99.8%	98.92%	8.7%	99.42%

The input feature map contains:

h $\in R^{m*u*u}$.

m→feature map.

uxu→size of the feature map.

The size of the convolution kernel is u×v, where v⊂=l .

θ ={w, w^1, b, b^1} = parameter of convolutional autoencoder layer:

b $\in R^{m}$.

w= {w_i, i=1,2...m}.

w_i=$\in R^{nu}$2.

w^1= {w_i ^{1}, i=1, 2, m}.

u×u pixels patch are selected by encoding the input image.

Kernel ‘l’ is used for convolutional calculation. The neuron value is calculated from the output layer by:

O_{ij}=f(x_i)= σ (w_j x_i+b).

σ →activation function.

RELU(v)={ $x\ if\ \ x \geq 0$|| 0 if x < 0 }.

The output O_{ij} has encoded that x_i reconstructed by $O_{ij:}$

$$x_i^1 = f^1\left(\text{Oij} = \varphi\left(w_i^1.\text{Oij} + b^1\right)\right).$$

After reconstruction of input x_i u×u, N patches are obtained from input images:

$X_{i,}$ (i=1, 2...N).

Cost function=$\frac{1}{N}\sum_{i=1}^{N}(x_i\ \ x_i^1)$.

Here; the reconstruction error=$||x_i - x_i^1|| = \left|\left|x_{i-}\ \varphi\left(\sigma(x_i)\right)\right|\right|^2$ a stochastic gradient decent.

The next stage of the convolutional layer is the pooling layer:

O_i ^{j}=max (x_i ^{j}),

where $x_i^j = jth\ region\ of\ ith\ feature\ map,$

O_i ^{j}→the ith neuron of the jth feature map.

So, as we can see above, for every image, there is one reconstructed image rebuilt by the autoencoder technique. For this reason, the training process is so higher with 80 epochs.

4.2.1 Cost Function

After completion of convolutional autoencoder, max-pooling, and fully connected layer, SoftMax is the autoencoder layer of classification is user forgetting optimal result.

Y_i ^{1} is the probability of AD and no AD:

Table 2. Comparison performance of PCA, ICA, and CANN

PCA	ICA	CANN
The algorithm is used against data that is not labeled. It automatically split the dataset into groups based on their similarities.	Independent component analysis is one of the most frequently used dimensionality reduction techniques and is based on information theory. PCA searches for independent functions, which is the main distinction between it and ICA.	By changing the connected layer with the convolutional layer, the simple autoencoder structure changes to the convolutional autoencoder technique.

$$\widehat{Z_l}=\frac{e^{(Oi)}}{\sum_{k=1}^{l} e^{(Ok)}}$$

where O$_i$=σ $(\sum_{v=1}^{l} x^a\ w^a\ b^a)$ presents output features x^agenerated from the fully connected layer.

From Table1, we get the classification performance of CANN, and Fig. 4 shows the accuracy plot.

5 Comparative Study

There are several feature extraction techniques like PCA, ICA, and CANN, which are part of the dimensionality reduction process to create a manageable group to ease the process. These techniques can help to reduce the amount of big data for a better and more accurate model. From table2, we get the comparison performance of PCA, ICA, and CANN.

Similar to PCA (principal component analysis) AEs perform dimensionality reduction in the compression phase. PCA is used for linear transformation and AE for nonlinear transformation.

5.1 ICA

The proposed ICA method is based on some criteria, which are defined by the FastICA algorithm [8]. It is possible to reconstruct an MRI picture using a linear combination of both the basic function and the corresponding coefficient. All MRI data are divided into testing and training datasets with an 8:2 ratio to get the best accuracy with comparable classified into different categories AD, MCI, and CN using SoftMax classifier. Our main motive for using ICA is to separate the observed signal from noise signals.

Here the two unsupervised techniques ICA and CANN are very useful among other related work to classify MRI images to distinguish MCI, AD, and CN subjects. For early diagnosis of Alzheimer's disease, both ICA and CANN are very efficient biomedical techniques.

Therefore, in our research work, we want to give some light on biomedical applications to extract useful features by deleting unnecessary noise. ROI is one of the oldest techniques for analyzing Brain scans and is a very time and manpower-consuming process.Fig.5, shows the architecture of ICA feature learning with proper steps for varieties of ICA techniques applied to separate the data.

Separate AD-related information or signal from noisy signals through ICA gives us a way to further research. For biomedical multi-subject data processing, human error is one of the important drawbacks for feature extraction of any type of disease. As a result, ICA is used for better feature extraction techniques for a more accurate diagnosis.

The extracted feature is fed into machine learning algorithm for classification. CA combined with SVM can make a better model for classifying different stages of Alzheimer's diseases into AD, MCI, and CN. FastICA features extracted with an SVM classifier had a 94.12 percent accuracy rate.

By using hybrid enhanced independent component analysis Basheera et al. analyzed MRI grey matter images with 90.4% [9]. ICA is a very powerful unsupervised technique for analyzing high-dimensional data.

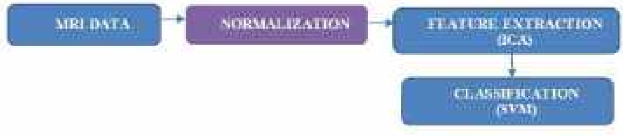


Fig. 5. Architecture of ICA feature learning

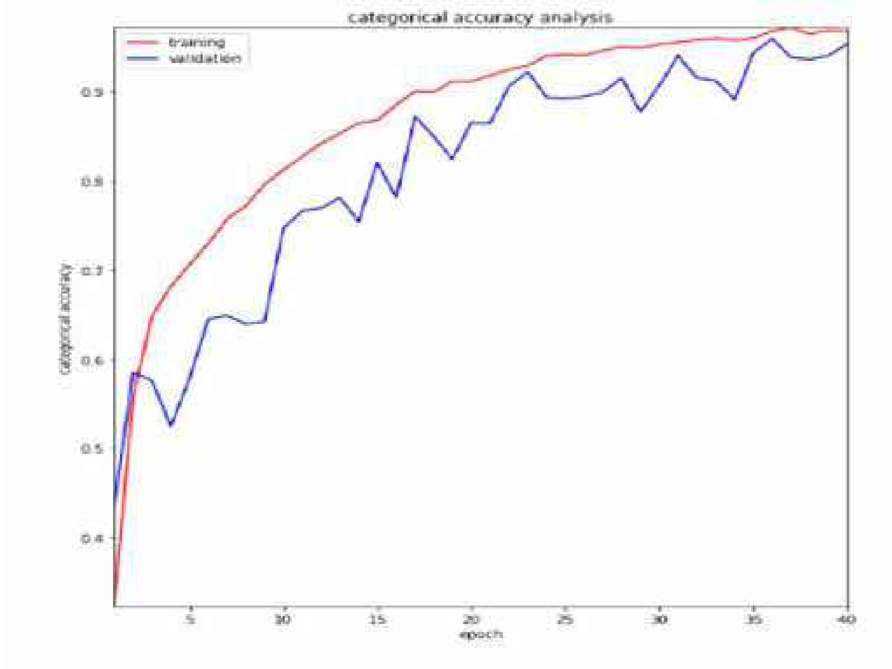


Fig. 6. Accuracy plot for ICA

Table 3. Classification performance of ICA

Method	AUC	Accuracy	Loss	F1
ICA	99.2%	94.12%	17.4%	94%

To effectively distinguish between different stages of Alzheimer's, the mathematical model of ICA makes a difference in our research.

From Table 3, we get the Classification performance of Independent component analysis (ICA), and Fig.6 shows the accuracy plot.

Combining ICA with SVM, we effectively analyzed and classify between different stages of disease into Alzheimer’s disease (AD), mild cognitive impairment (MCI), and cognitively normal (CN). Some of the biomedical techniques are not very suitable for the early diagnosis of disease.

A key challenge to address this problem is different unsupervised techniques to acquire focused based research on early detection of finding of AD.

ICA is among the unsupervised techniques used for feature extraction is also been effective in the case of Alzheimer's diagnosis with high accuracy. Similar techniques can be used for applications in brain imaging and health.

Working with data preprocessed with the ICA-model, artificial neural networks and support vector machines (SVMs) successfully extracted features using FastICA, and then trained SVM-based classifiers to discriminate AD, MCI, and CN separately with an accuracy of 94.12%.

Furthermore, it could be better research performance with machine learning technique for multiclass classification with 40 epochs.

The most notable variance we noticed was a 94.12% accuracy rate using T1 weighted MRI imaging of AD, MCI, and CN. Furthermore, we analyzed more models to process T2 weighted MRI images with other deep learning techniques for better performance.

6 Experiment and Result

6.1 Dataset

The dataset used for our model is taken from the ADNI database. For more detail, please go through the ADNI website. ADNI launches in 2003 as a partnership of both public and private to investigate diseases like Alzheimer in this study.

Researchers collect data different from this website for better research with study resources. Alzheimer's Disease Neuroimaging Initiative (ADNI) improves better clinical trials with a standardized protocol for the prevention and treatment of AD. ADNI has made a global impact for better diagnosis in the medical field. ADNI1 and ADNI2 are two participants who had 3.0 T T1-weighted images.

All images are downloaded in neuroimaging informatics technology initiative (Nifti) file format.

For diagnosed MRI analysis with CNN, procedures were performed on an Intel Xeor silver 4110 -2.1GHZ -3.5 GHz x8 core high-performance computer (HPC). We used the GPU platform, which has the largest collection of ALU cores with model Nvidia GTX1080Ti pascal chipset. Cuda(x) is stored in GPU (Graphics Processing Unit), where lots of external libraries are there and it has a channel where an organization creates its channel.

A collection of brain images was collected and proposed CANN and ICA with various classifiers were taken into consideration. fMRI and MRI are two techniques that are widely used to convey and provide anatomical information and are ideally suited for brain analysis research.

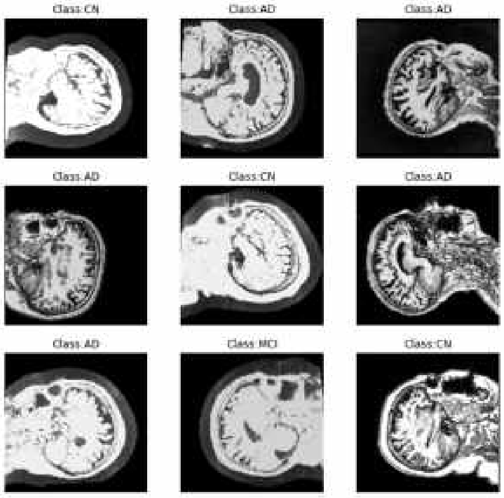


Fig. 7. Different stages of Alzheimer's disease

Table 4. Classification performance of different CNN architecture

CNN Architecture	Accuracy	Precision	Recall
VGG-16	99.43	98.92	98.01
ResNet-50	92.43	92.13	93
AlexNet	93.13	92.12	93.11

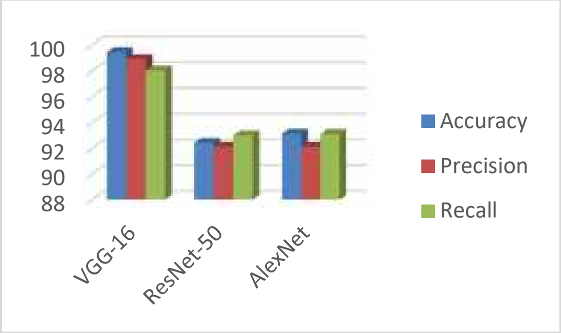


Fig. 8 Performance comparison of three different CNN architectures

An unlabeled data for unsupervised training. The 64x64 patches are captured randomly from AD MRI data. convolutional autoencoder (CANN) consists of 6 layers in the structure. It has 4 groups of connections between different layers of CNN, i.e., convolutional layer, pooling layer, and fully connected layer.

Input: Input: 64×64 patch captured from MRI image. Different patches are collected from MRI slides.

ISSN 2007-9737

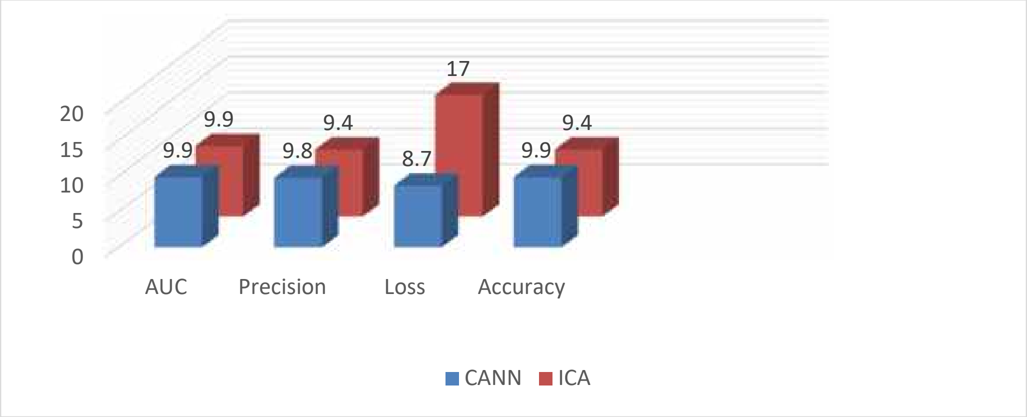


Fig. 9. Performance comparison of the proposed model

C1: kernel is 5 × 5, in the first step, the number of convolution kernels is 50, and the non-linear function is Relu.

P1: max pool and size of pooling are 2 × 2.

C2: kernel is 5 × 5, in the first step, the number of convolution kernels is 50, and the non-linear function is Relu.

P2: max pool and size of pooling are 2 × 2.

C3: convolution kernel is 3×3 in the first step the number of convolution kernels is 50, and the non-linear function is Relu.

P3: max pool and size of pooling are 2 × 2.

Full: fully connected layer.

Output: SoftMax classifier, for 2 classes.

7 Classification

We compare the classification performance among CNN and ICA with unlabeled unsupervised learning in this result we classify to achieve the best accuracy with the lowest error rate in analyzing AD data. Fig. 7 shows the different stages of Alzheimer's disease by using the Keras library.

We employed three distinct architectures to represent similarity and classification as accuracy, precision, recall, and other metrics to conduct a full comparison in the table in Table 4.

As can be observed in Fig. 8, the accuracy is high and is approximately 99.42 percent.

8 Discussion

For a more accurate multiclass classification of Alzheimer's disease, with multiple stages measured through diverse research. We compared the two techniques and concluded that CANN is more applicable compared to ICA due to higher accuracy.

Deep learning approaches are a more satisfactory technique for showing an affordable result in critical studies.

Importantly, our algorithm performed well for varieties of high-dimensional datasets with better performance.

Although it was difficult to interpret heterogeneous MRI and fMRI data using multiple protocols, we found in our study that the examined model is not significantly affected by image value. To achieve high accuracy for dataset-specific approaches when categorizing AD or MCI, or CN patients, CNN was trained, validated, and evaluated in our studies utilizing high-dimensional datasets and various autoencoder techniques.

In conclusion, Fig. 8 shows the potential of performance to structure a model for early detection of AD.

9 Conclusion

In our research, High dimensionality is one of the key difficulties to manage huge medical data. Therefore, here is a detailed preview of one of the features of V's big data analytics characteristics. We examined the effects of two approaches, convolutional autoencoder (CANN) and independent component analysis (ICA), and found that CANN beats ICA models in terms of convergence speed and has a higher accuracy of 99.42%.

We also increase the precision of deep learning techniques for classifying various forms of Alzheimer's disease. We also wish to investigate how the impact manifests itself in other deep learning architectural methodologies with an effective approach to Alzheimer's disease.

Additionally, there is a good chance that the suggested method will be applied to other medical fields.

References

1. **Rian L., Petanceska, S. (2019).** Accelerating medicines partnership® program for Alzheimer's disease (AMP® AD). National Institute on Again, Accessed on: Aug. 3, 2021. [Online]. Available: https://www. nia.nih.gov/research/ amp-ad.
2. **Hippius, H., Neundörfer, G. (2022).** The discovery of Alzheimer's disease. Dialogues in clinical neuroscience, Vol. 5, No. 1, pp.101–108. DOI: 10.31887/DCNS.2003.5.1/hhippius.
3. **Kwasigroch, A., Mikołajczyk, A., Grochowski, M. (2017).** Deep neural networks approach to skin lesions classification—a comparative analysis. 22nd International Conference on Methods and Models in Automation and Robotics, pp. 1069–1074. DOI: 10.1109/MMAR .2017.8046978.
4. **Suk, H. I., Shen, D. (2013).** Deep learning-based feature representation for AD/MCI classification. In International conference on medical image computing and computer-assisted intervention Springer, Berlin, Heidelberg. pp. 583–590. DOI: 10.1007/978-3-642-40763-5_72.
5. **Liu, J., Li, M., Lan, W., Wu, F. X., Pan, Y., Wang, J. (2016).** Classification of Alzheimer's disease using whole brain hierarchical network. IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 15, No. 2, pp. 624–632. DOI: 10.1109/TCBB.2016.2635144.
6. **Islam, J., Zhang, Y. (2018).** Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks. Brain informatics, Vol. 5, No. 2, pp. 1–14. DOI: 10.1186/s40708-018-0080-3.
7. **Despotović, I., Goossens, B., Philips, W. (2015).** MRI segmentation of the human brain: challenges, methods, and applications. Computational and Mathematical Methods in Medicine, Vol. 2015, DOI: 10.1155/2015/450341.
8. **Oja, E., Yuan, Z., 2006.** The FastICA algorithm revisited: Convergence analysis. IEEE Transactions on Neural Networks, Vol. 17, No. 6, pp.1370–1381. DOI: 10.1109/TNN.2006.880980.
9. **Basheera, S., Ram, M. S. S. (2019).** Convolution neural network–based Alzheimer's disease classification using hybrid enhanced independent component analysis based segmented gray matter of T2 weighted magnetic resonance imaging with clinical valuation. Alzheimer's & Dementia: Translational Research & Clinical Interventions, Vol. 5, pp. 974–986. DOI: 10.1016/j.trci.20 19.10.001.

Article received on 26/11/2021; accepted on 23/08/2022. Corresponding author is Binod Kumar Pattanayak.

ISSN 2007-9737

Classification of Paintings by Artistic Style Using Color and Texture Features

Ivan Nunez-Garcia[1], Rocio A. Lizarraga-Morales[2], Uriel H. Hernandez-Belmonte[2], Victor H. Jimenez-Arredondo[2], Alberto Lopez-Alanis[3]

[1] Universidad de Guanajuato,
Departamento de Estudios Multidisciplinarios,
Mexico

[2] Universidad de Guanajuato,
Departamento de Arte y Empresa, Salamanca,
Mexico

[3] Universidad de Guanajuato,
División de Ingenierías Campus Irapuato Salamanca,
Mexico

{i.nunezgarcia, ra.lizarragamorales,uh.hernandez, vhjimenez, a.lopezalanis}@ugto.mx

Abstract. In this paper, an approach for the classification of paintings by artistic style using color and texture features is proposed. Our approach automatically extracts a set of visual features that effectively discriminate among diverse artistic styles. Additionally, our proposal performs an effective selection of the most relevant features to be used in an artificial neural network architecture. Using the most important features allows our system to achieve an efficient learning process. The proposed system analyzes digitized paintings using a combination of color and texture features, which have shown to be highly discriminatory. Our approach consists of two main stages: training and testing. Firstly, in the training stage, the features from seven artistic styles are extracted to train a multi-layer perceptron. Secondly, the learned model is utilized to determine the artistic style of a given incoming painting to our system. The experimental results, on an extensive dataset of digitized paintings, show that our method obtains a higher accuracy in comparison with those obtained by the state-of-the-art methods. Moreover, our proposal attains a higher accuracy rate using fewer features descriptors.

Keywords. ANN, PCA, artistic style, classification, color features, paintings, texture features.

1 Introduction

Painting has been considered as one of the first human artistic expression [10]. It is one of the most important events of human intelligence, therefore, it is considered an intellectual activity [26].

Throughout art history, artistic styles have been defined depending on their space in time, technique, or relevance. In order to classify a given painting, different visual indicators must be considered: e.g. the color palette, the stroke style, color mixing, edge softness, lines, scene theme, among others.

The analysis and classification of a given painting in their artistic style are considered a complicated cognitive task because of the knowledge required about art history and the number of visual cues that needed to be considered [9, 33].

Since there is no simple rule to define the artistic style of a painting, the classification has been exclusively performed by human experts. However, given the massively increasing volumes of digitized paintings, the development of an automatic classification system is an emerging need for the proper administration of such amount of information.

doi: 10.13053/CyS-26-4-4022

The main goal of such a system is to recognize an artistic style by analyzing the visual aspects of a given painting. Automatic systems have also analyzed other forms of art. For example, Scaringella et al. [29] and Markov et al. [22] classify music pieces by their genre and Zhao et al. [34] categorize buildings by their architectural style.

In the existing literature, we can find a number of approaches for the classification of paintings by artistic style. One of the first questions to be answered is the type and number of visual features to analyze. Among the most popular cues included in classification systems are color, and texture.

Concerning the use of color features, Gunsel et al. [12] incorporated color cues such as the percentage of dark colors, the gradient coefficient, and the luminance information to classify 5 artistic styles. The perceptual color spaces, e.g. the color spaces proposed by the *Commission Internationale de l'Éclairage* (CIE) such as CIE 1976 L*a*b* (CIELab) and CIE 1976 L*u*v* (CIELuv), have been adopted in diverse research works such as the work by Siddharth et al. [2] and Guanming et al. [21], to obtain an approach that resembles the human visual system.

However, the task of classifying paintings using color as a single feature may be difficult. Recent research has been found that humans often combine multiple sensory cues to improve the performance of perceptual tasks, motivating recent research on the integration of more than one feature. In fact, it has been found that human perception is performed by using collectively color and texture information [15].

The inclusion of texture cues in classification systems has been essential to determine the style of a given painting, capturing visual features such as stroke style, relief, and the spacial relationship of color. Diverse studies have emphasized the inclusion of texture features in combination with color indicators.

Zujovic et al. [35] to classify *5* artistic styles used a set of features including edges, Gabor filters, and histograms of the hue\saturation\value (HSV) color space. Culjak et al. [6], considered the concatenation of *68* features to classify five different styles.

The features used include the number of edges, dark pixels, symmetry, and average values of the red\green\blue (RGB) and HSV color spaces. Guanming Lu et al. [21] proposed to compute the mean and the standard deviation of each color component in four color spaces. Additionally, proposed to combine the information extracted from the moments of second-order and contrast measurements.

Condorovici et al. [4] used a *3*D Lab color histogram, and combined Gabor filter energy, number of edges, among other cues. In order to classify paintings into *3* styles, Shamir et al. [30] used texture cues such as Gabor filters, statistical moments of the intensity component, perceptual texture features, edge statistics, and spectral analysis.

Siddiquie et al. [31], introduced the use of Gaussian and Laplacian-Gaussian filters to capture the behavior of the isotropic and anisotropic texture within a painting. Additionally, they use histograms of oriented gradients, edges, and color cues from diverse color spaces, for the classification of *6* artistic styles.

Non-supervised learning, such as clustering, has also been proposed for painting classification. Liao et al. [19] proposed to use color, texture, and spatial layout to classify oil painters using a clustering Multiple Kernel Learning Algorithm. Kim et al. [18] use both low-level and high-level features, such as color, shade, texture, saturation, stroke, and color balance, among others to classify oriental painting for wellness contents recommendation services.

Considering that the combination of color and texture cues might result in a high-dimensional feature space, recently, several works have been focused on finding the most relevant set of features for painting classification. Some examples are the research of Huang et al. [14], Gultepe et al. [11], and Paul et al. [27].

The search for the best set of features implies the initial selection of a set of base features. The first selection can be performed automatically or manually. Later, a feature selection process is performed to get the best subset of features from the initial set. The work presented by Huang et al. [14] proposed the usage of a bag-of-features based on MPEG-7 specification.

The bag-of-features contain four descriptors, based on color and texture, which are used to generate an initial feature set for the style classification. Gultepe et al. [11] uses unsupervised features learning with k-means (UFLK), principal component analysis (PCA), and raw pixels without preprocessing.

The authors classify among eight artistic styles: Baroque, Impressionism, Post-impressionism, Realism, Art-Nouveau, Romanticism, Expressionism, and Renaissance. In the work presented by Paul et al. [27], the use of several features for the classification is presented.

They used features such as dense scale-invariant feature transform (SIFT), the dense histogram of oriented gradient, and concatenate 2×2 cells (HOG2X2), local binary patterns (LBP), gradient local auto-correlation (GLAC), color naming, and GIST. In such a manuscript, they found that LBP obtains the best performance for the classification of five different styles.

Despite the remarkable performance of the approaches mentioned above, the proper set of features is still an open problem. Additionally, the number of different artistic styles distinguished by the previously proposed methods is still low in comparison to the number of styles in the art industry.

In this paper, we propose a system for the classification of paintings by artistic styles using color and texture cues in a computational intelligence approach. The proposed system is performed in three stages: feature extraction, feature selection, and an Artificial Neural Network (ANN) as a classifier.

Firstly, we select a set of color and texture features and carefully structure their combination. As has been mentioned before, the integration of color and texture features has been essential to describe paintings attributes such as stroke style, relief and, color spacial relationship.

It is worth to mention that in most of the previously reviewed approaches, the texture features are usually computed from the luminance component or from a gray level image obtained by converting the color image into grayscale. However in order to obtain more visual information, in this research work, we propose exploring the combination of color and texture cues extracted from color components of different perceptual color spaces. Secondly, considering that the obtained feature space is high-dimensional, we propose to reduce its dimensionality by using the principal component analysis (PCA) method to select the most relevant features.

The goal is a painting classification system that uses low-dimensional feature vectors without compromising robustness and accuracy. Thirdly, we propose to use a multi-layer perceptron (MLP) as the classifier. We designed the architecture of the MLP to effectively classify seven different painting styles.

The proposed method was evaluated on an extensive and challenging database of paintings designed for artistic style classification. Experiments indicate that our method attains higher accuracy in comparison to other state-of-the-art methods. From now on, we call our method CTArt, for Color and Texture for Artistic Classification.

The remainder of this manuscript is structured as follows: in Section 2, we present the overall proposed classification framework. Additionally, our feature extraction, feature selection, and classification scheme are also introduced. The experiments and results are given in Section 3, finally, the concluding remarks are presented in Section 4.

2 Methodology

In this section, the theoretical framework of our proposed CTArt is defined. An overview of the proposed approach is illustrated in Figure 1. From this figure, we can observe that our system is divided into two stages: training and testing. In Figure 1(a), the training stage is depicted. Firstly, we perform a color space transformation of the learning images.

Later, extensive feature extraction is performed. Since the obtained set of features is high-dimensional, we use the Principal Component Analysis (PCA) to select the most discriminant features. Then, the obtained principal features are submitted to the multi-layer perceptron (MLP) algorithm for learning purposes.

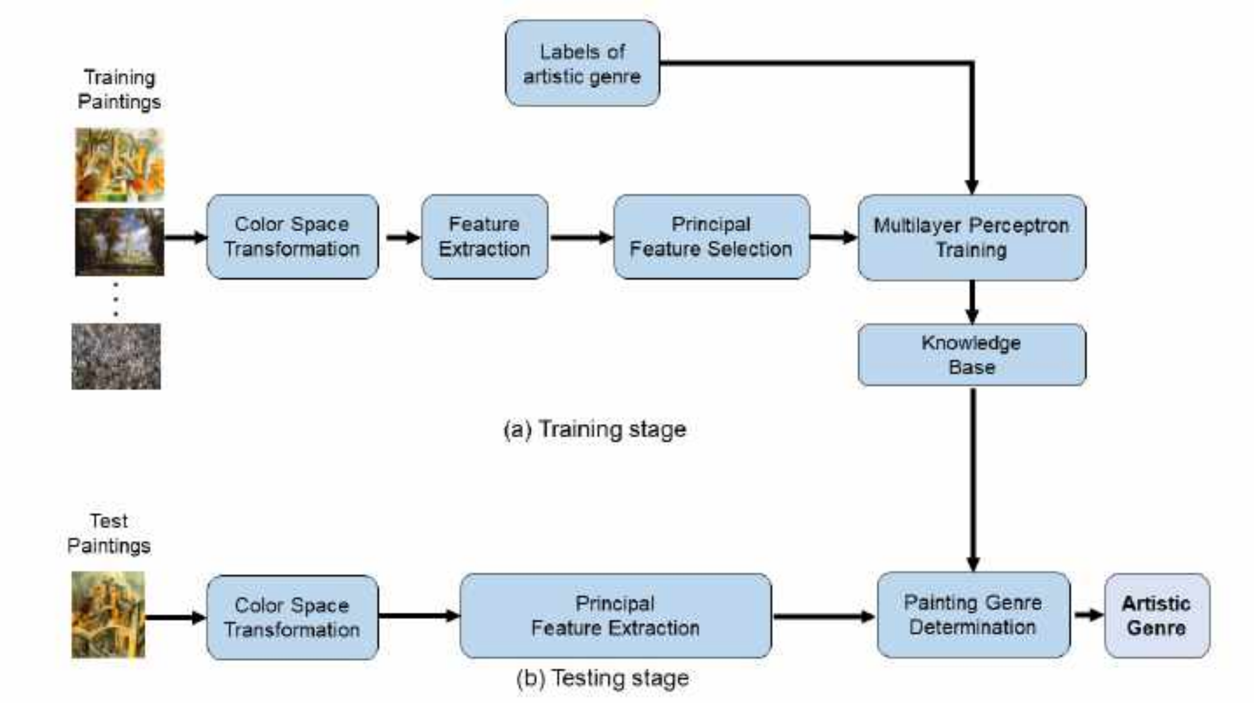

Labels of artistic genre
Training Paintings
Color Space Transformation
Feature Extraction
Principal Feature Selection
Multilayer Perceptron Training
Knowledge Base
(a) Training stage
Test Paintings
Color Space Transformation
Principal Feature Extraction
Painting Genre Determination
Artistic Genre
(b) Testing stage

Fig. 1. Overview of our proposed CTart model. (a) Color space transformation of the learning images is performed. Then, feature extraction is performed and the dimensionality reduction is carried out. Later, the learning process is performed by using a Multilayer Perceptron algorithm. (b) Principal feature extraction is performed. Then, by using the obtained model the determination of the painting style is performed

The obtained model allows us to determine the artistic style of a given painting. In Figure 1(b), the process to test the model is depicted. Only the principal features are computed for a given test image. The outcome of our proposal is the artistic style of the test image. Each block of Figure 1 is detailed in the next subsections.

2.1 Color Spaces and Color Space Transformations

It is well known that the performance of a color analysis method highly depends on the choice of the color space [3]. The RGB color space is the most widely used in the literature. In this color space, a particular color is specified in terms of the intensities of three additive colors: red, green, and blue [8]. The RGB space is the model whose system is based on the cartesian coordinate system where its primary spectral components are: red, green, and blue.

The images of this model are formed by the combination of different portions of each of the primary RGB colors. Although the RGB space is the most used in the literature, this representation does not permit the emulation of the higher-level processes that allow the perception of color in the human visual system [13].

Different studies have been oriented to the determination of the best-suited color representation for a given approach [5], where color is the only feature to be taken. Some of them have found that, for color-alone methods, the so-called perceptual color spaces are the most appropriate.

In our proposal, in order to resemble the human perception of colors, we explore the extraction of features in different color spaces. Hence, in addition to the RGB representation, we explore perceptual color spaces, such as CIELab, CIELuv, and HSV. The CIELab color space is normally used to describe all the colors that the human eye can perceive.

It was proposed in 1976 by the CIE [23] as an approximation to a uniform color space. The CIELab color space is a mathematical transformation from the CIEXYZ color space. The three axes of the CIELab color space are indicated by the names L*, a*, and b*.

They represent, respectively, luminosity, hue from red to green, and hue from yellow to blue. The CIELab color space is described in Eqs. 1-4, when RGB space is previously transformed to the CIE tristimulus CIEXYZ color space:

$$L* = 116f\left(\frac{Y}{Yn}\right) - 16, \tag{1}$$

$$a* = 500f\left[\frac{X}{Xn} - \frac{Y}{Yn}\right], \tag{2}$$

$$b* = 200f\left[\frac{Y}{Yn} - \frac{Z}{Zn}\right], \tag{3}$$

$$f(t) = \begin{cases} t^{1/3}, & t > \alpha^3, \\ t/(3\alpha^2) + 16/116 & t \leq \alpha^3, \end{cases} \tag{4}$$

where $\alpha = \mathit{6/29}$, and X_n, Y_n, Z_n are the white reference for the scene in CIEXYZ. In this work, we have used the standard for a daylight illuminant D65. The color space CIELuv, is defined by Eqs. 5-7. It is worth to mention that the L* component in CIELuv space is identical to the L* component in CIELab color space:

$$L* = 116f\left(\frac{Y}{Yn}\right) - 16, \tag{5}$$

$$u* = 13L * (u' - u'_n), \tag{6}$$

$$v* = 13L * (v' - v'_n), \tag{7}$$

where u', u'_n and v', v'_n are calculated from Eqs. 8-11:

$$u' = \frac{4X}{X + 15Y + 3Z}, \tag{8}$$

$$u'_n = \frac{4X_n}{X_n + 15Y_n + 3Z_n}, \tag{9}$$

$$v' = \frac{9Y}{X + 15Y + 3Z}, \tag{10}$$

$$v'_n = \frac{9Y_n}{X_n + 15Y_n + 3Z_n}. \tag{11}$$

The tristimulus values $X_n,\ Y_n,\ and\ Z_n$ correspond to the illuminant, with $Y_n = \mathit{1}$. The HSV model, defined by Smith [28], is a non-linear transformation from the RGB color representation. Each color in this model is defined by hue, saturation, and value dimensions. The HSV color space is described by Eqs. 12-14:

$$H = \begin{cases} 0°, & \Delta = 0, \\ 60° \times \left(\frac{G' - B'}{\Delta}\right), & C_{max} = R', \\ 60° \times \left(\frac{B' - R'}{\Delta} + 2\right), & C_{max} = G', \\ 60° \times \left(\frac{R' - G'}{\Delta} + 4\right), & C_{max} = B', \end{cases} \tag{12}$$

$$S = \begin{cases} 0, & C_{max} = 0, \\ \frac{\Delta}{C_{max}}, & C_{max} \neq 0, \end{cases} \tag{13}$$

$$V = C_{max}, \tag{14}$$

where $R' = R/255$, $G' = G/255$, $B' = B/255$. Besides, $\Delta = C_{max} - C_{min}$, and C_{max}, C_{min} are the maximum and minimum value of the normalized components.

2.2 Color Features

In order to clearly define the color information used by our proposed system, we consider the set of color components such as R, G, B, L^*, a^*, b^*, u^*, v^*, H, S, and V. Firstly, from the set of color components mentioned above, we compute the regular mean and standard deviation of each color component. In such a way, we have *22* features, a mean and a standard deviation by each of the *11* color channels.

Considering that the color palette used by the artist is one of the essential features to categorize a painting [6], we include in our set of color features, the *7* most important colors within the image. Such colors are selected from the RGB space by using the k-*means* algorithm [20]. The k-*means* algorithm is defined in Eq. 15:

$$W_n = \frac{1}{n}\sum_{i=1}^{n} \min_{1 \leq j \leq k} \parallel X_i - a_j \parallel^2, \tag{15}$$

where n is the number of pixels of each image. For our proposal, we set $k = 7$. The *7* resulting means give us, the RGB coordinates of the *7* main colors that compose the painting. Therefore, we add *21* features to our feature set. In total, we obtain a set of *43* color features.

2.3 Texture Features

Visual texture is a perceived property on the surface of all objects and it is a significant reference for their characterization and discrimination. There is a number of perceived qualities, which play an important role in describing the visual texture.

In artistic painting, the texture is of great importance, since it can give us hints of the stroke, the contrast of colors, edge softness, lines, etc. In our methodology, we propose to use two of the most widely used texture features such as Sum and Difference Histograms, and Local Binary Patterns.

Such texture features, which are described in the following paragraphs, have demonstrated to be highly discriminant and robust in diverse classification tasks.

On one hand, we use a subset of the statistical features extracted from the Sum and Difference Histograms (SDH) proposed by Unser [32]. The SDH establishes a numeric relation between two different pixels separated by a given distance d, within a gray-scale image I with N_g gray levels. Consider two picture elements, y_1 and y_2, as seen in Eq. 16, in one relative position given by (d_1, d_2):

$$\begin{cases} y_1 &= y_{k,l}, \\ y_2 &= y_{k+d_1,l+d_2}. \end{cases} \tag{16}$$

The non-normalized sums and differences, associated with relative displacement (d_1, d_2) are defined in Eqs. 17 and 18:

$$S_{k,l} = y_{k,l} + y_{k+d_1,l+d_2}, \tag{17}$$

$$D_{k,l} = y_{k,l} - y_{k+d_1,l+d_2}. \tag{18}$$

The histograms of sums and differences with parameter (d_1), where d_1 is an element in a relative position of an image, over the domain D, are defined in Eqs. 19 and 20:

$$h_s(i; d_1) = h_s(i) = \mathrm{Card}\left\{(k,l) \in D, S_{k,l} = i\right\}, \tag{19}$$

$$h_d(j; d_1) = h_d(j) = \mathrm{Card}\left\{(k,l) \in D, D_{k,l} = j\right\}. \tag{20}$$

Then, the normalized sum and difference histograms are defined in Eqs. 21 and 22:

$$P_s(i) = \frac{h_s(i)}{N}; \ (i = 2, ..., N), \tag{21}$$

$$P_d(j) = \frac{h_d(j)}{N}; \ (j = -N_g + 1, ..., N_g - 1). \tag{22}$$

From the set of features proposed by Unser, we only used a subset of seven features defined in Eqs. 23-29:

$$\text{Mean} = \frac{1}{2}\sum_{i=1} iP_s(i) = \mu, \tag{23}$$

$$\text{Variance} = \frac{1}{2}\left[\sum_{i=1}(i-2\mu)^2 P_s(i) + \sum_j j^2 P_d(j)\right], \tag{24}$$

$$\text{Energy} = \sum_i P_s(i)^2 \sum_j P_d(j)^2, \tag{25}$$

$$\text{Correlation} = \frac{1}{2}\left[\sum_i (i-2\mu)^2 P_s(i) - \sum_j j^2 P_d(j)\right], \tag{26}$$

$$\text{Entropy} = -\sum_i P_s(i)\log\{P_s(i)\} - \sum_j P_d(j)\log\{P_d(j)\}, \tag{27}$$

$$\text{Contrast} = \sum_j j^2 P_d(j), \tag{28}$$

$$\text{Homoge.} = \sum_j \frac{1}{1+j^2} P_d(j). \tag{29}$$

On the other hand, we include in our proposed set of features, the local binary patterns descriptor (LBP), proposed by Ojala et al. [24, 25]. The LBP is a theoretically simple yet efficient approach, to characterize the spatial structure of local texture.

The LBP is a distribution that describes the local texture. According to Ojala, a monochrome texture image T in a local neighborhood is defined as the joint distribution of the gray levels of $P(P > 1)$ image pixels $T = t(g_c, g_0, \ldots, g_{P-1})$, where g_c is the gray value of the center pixel and $g_p(p = 0, 1, \ldots, P - 1)$ are the gray values of P equally spaced pixels on a circle radius $R(R > 0)$, that form a circularly symmetric neighbor set. If the coordinates of g_c are (x_c, y_c), then the coordinates of g_p are $(x_c - R\sin(2\pi p/P), y_c + R\cos(2\pi p/P))$.

The LBP value for the pixel g_c is defined in Eq. 30 and 31:

$$\mathrm{LBP}_{P,R}(g_c) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \tag{30}$$

$$s(t) = \begin{cases} 1, & t \geq 0, \\ 0, & \text{otherwise}\,. \end{cases} \tag{31}$$

A number of variants of the LBP descriptor have been proposed over the years. Hence, the LBP descriptor has become one of the most widely used texture descriptors on diverse applications. In our approach, we propose to use the rotation invariant operators with uniformity of 2 (riu2): $\mathrm{LBP}_{8,1}^{\mathrm{riu2}}$ and $\mathrm{LBP}_{16,2}^{\mathrm{riu2}}$.

As has been mentioned before, the use of color and texture information collectively has strong links with the human perception and in many applications, the color-alone or texture-alone information is not sufficient to describe an artistic painting.

Ilea and Whelan [15] have found that algorithms that, i) integrate the color and texture attributes in succession and, ii) the methods that extract the color and texture features on independent channels and then combine them using various integration schemes proved to be more promising for applications where links with the human perception are assumed.

Taking this into account, in our study, we obtain the texture features from each color channel of the painting images transformed to the $CIELab$, $CIELuv$, and HSV color spaces. Therefore, we obtain *7* SDH features by each color channel, which adds *77* texture features to the painting descriptor.

Additionally, we obtain *10* and *18* features from the $\mathrm{LBP}_{8,1}^{\mathrm{riu2}}$ and the $\mathrm{LBP}_{16,2}^{\mathrm{riu2}}$ descriptors respectively, by each color channel. The concatenation of each descriptor results in a vector of *308* texture features. In Figure 2, we depicted the complete feature extraction procedure performed by our system.

Firstly, we transform the original RGB image into *3* perceptual color spaces. Secondly, from each color component, we obtain color features and texture features in cascade. The concatenation of both sets of features results in a feature vector of *428* dimensions.

2.4 Principal Component Analysis

The use of high dimensional feature vectors is very common in different applications and it is often difficult to analyze. In order to interpret and classify such vectors, most methods require to reduce vector dimensionality, in such a way that the most relevant information in data is preserved.

Several techniques have been developed for this purpose. Among the diverse techniques, the principal component analysis (PCA) has positioned as a classic technique. The main goal of such a technique is to reduce the dimensionality of a feature vector while preserving as much statistical information as possible [16].

The PCA obtains the most relevant features by using the linear transformation of correlated variables. Firstly, the principal component analysis starts by normalizing the feature data. The mean is substracted from data and then, it is divided by the standard deviation.

Secondly, in Eq. 32, the singular value decomposition of the covariance matrix from the normalized data is performed, resulting in eigenvalues and eigenvectors of the covariance matrix. Thirdly, eigenvalues are sorted in a decreasing order effectively representing decreasing variance in the data [17].

Principal components are obtained by multiplying the originally normalized data with the leading eigenvectors whose exact number is a user-defined parameter. Lastly, the high dimensional feature vector is now represented by relatively few uncorrelated principal components that are later used in the learning process. In this paper, we reduce the original *428*-dimensions size vector to a *168*-dimensions size vector:

$$C^{n\times n} = \left(C_{i,j}, C_{i,j} = \mathrm{cov}\left(\mathrm{Dim}_i,\ , \mathrm{Dim}_j\right)\right), \tag{32}$$

where $C^{n\times n}$ is a matrix with n rows and n columns, and Dim_x is the xth dimension.

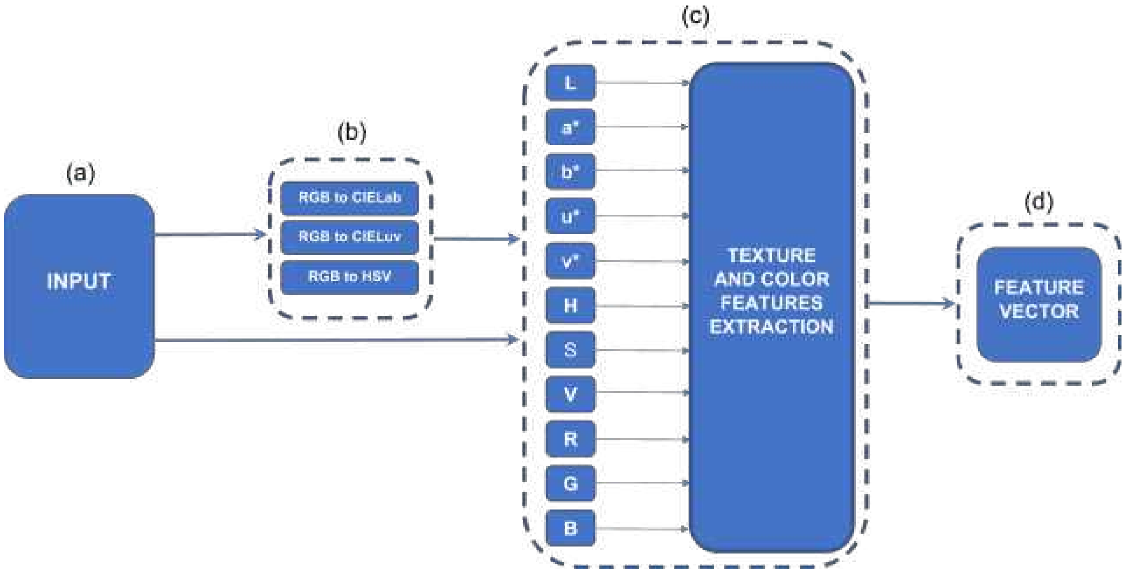


Fig. 2. The procedure of color and texture feature extraction. (a) The input image in RGB color space. (b) Transformation of the RGB input image into *3* color spaces such as CIELab, CIELuv, and HSV. (c) Extraction of color and texture features from the components of the *4* color spaces used. (d) The resulting vector feature is comprised of the concatenation of the diverse extracted features

2.5 Classification Approach Using Artificial Neural Networks

The artificial neural networks (ANNs) are widely used for classification purposes on diverse applications. Recent studies have found that neural network models such as feedforward and feedback propagation ANNs are performing better in their application to problems where strong links with human perception are assumed [1].

There are several different architectures for ANNs. However, one of the most well-known architectures is the multilayer perceptron (MLP), which maps the features at the input to a group of outputs. The information in the MLP runs from the input layer of neurons, through the hidden layer, to the nodes at the output.

In general, the MLP accomplishes a nonlinear transformation of the information from previous layers, applying weighted summations. The MLP is typically trained using the supervised learning method of backpropagation.

The backpropagation algorithm maps the input data to the required outputs by reducing the error between the target outputs and the computed outputs [7]. In this paper, our CTArt approach consists of an ANN with MLP architecture which contains *168* input neurons, *88* neurons in the hidden layer, and *7* neurons at the output.

In order to determine the best number of hidden neurons to our proposed network, we carried out preliminary experiments by varying the number of neurons in the middle layer. It is worth to mention that we varied the number of hidden neurons in a bounded range.

We found that setting as 88 the number of hidden neurons produces the best result. The main goal of the proposed architecture is to classify artistic paintings into their corresponding artistic styles.

Each artistic style is associated with an output node. The activation function, the sigmoid function (S), defined in Eq. 33 is used for the neurons in the hidden and output layers, where the weighted-input summation to the nodes is represented by η.

The Levenberg–Marquardt learning method was used in the backpropagation algorithm to train the network offline, with randomly-set initial weights and biases, and a mean square error of 1×10^{-8} as the learning rate (η):

$$S(\eta) = \frac{1}{1+e^{-\eta}}. \tag{33}$$

Table 1. Comparison and results table with Huang et al. [14]

Method	Feature Vector Dimension	Styles	Feature Reduction Approach	Classifier	Accuracy
Huang et al. [14]	186	6	SAHS	SVM	69.80%
CTArt	168	6	PCA	ANN	74.57%

Table 2. Performance comparison of Huang method and our proposal using the same dataset

Artistic styles	Huang			CTArt		
	Recall	AIR	NIR	Recall	AIR	NIR
	(%)	(%)	(%)	(%)	(%)	(%)
Cubism	77.30	60.60	72.70	30.00	13.33	16.00
Fauvism	54.50	37.83	45.40	83.00	66.33	79.60
Impressionism	72.70	56.03	67.20	67.00	50.33	60.40
NaïveArt	75.00	58.33	70.00	80.00	63.33	76.00
Pointillism	72.70	56.03	67.24	90.00	73.33	88.00
Realism	68.20	51.50	61.80	88.00	71.33	85.60
Average	70.06	53.38	64.05	**73.00**	**56.33**	**67.60**

3 Experimental Results

In this section, we present the experimental setup, we describe the dataset used, the parameter settings, and the performance metrics utilized. Additionally, we compare the results obtained by our proposed CTArt system and state of the art methods.

In order to perform an objective comparison, we propose to directly compare our method with the approach recently introduced by Huang et al. [14] which uses a self-adaptive harmony search (SAHS) for feature reduction and support vector machine (SVM) for the classification of 6 artistic styles.

The comparison is performed by using the same dataset which includes the following artistic styles: Cubism, Fauvism, Impressionism, Naïve art, Pointillism, and Realism. Additionally, we perform the experiments based on Huang et al. [14] evaluation methodology. In such approach performance metrics for the classification of each artistic style are proposed:

Normalized Improvement Ratio (NIR), the Absolute Improvement Ratio (AIR) and Accuracy. Normalized improvement ratio (NIR) is defined in Eq. 34 and the absolute improvement ratio (AIR) metric is defined as the numerator of the NIR metric. Accuracy is defined as in Eq. 35:

$$\mathrm{NIR} = \frac{\mathrm{recall} - \frac{1}{n}}{1 - \frac{1}{n}}, \tag{34}$$

where recall is the ratio of correctly identified paintings for a style, and n is the number or painting styles in a dataset:

$$\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{TN} + \mathrm{FN}}, \tag{35}$$

where TP, TN, FP and FN are the results for True Positives, True Negatives, False Positives and False Negatives, respectively. The experimental results are shown in Table 1. From this table, we can observe that our method attains 74.57% of accuracy.

On the other hand, Huang attains a lower accuracy of 69.80%. It is worth to mention that the feature vector dimension used in the approach of Huang is 186. On the contrary, our proposal uses a feature vector dimension of 168.

In Table 2, we present the results of the other metrics recall, NIR and AIR, for each artistic style. From this table we can observe that, in average, the performance of our proposal overcomes the Huang approach in all the metrics.

Our proposal attains a higher recall of 73.00% and Huang method obtains 70.06%. On the other hand, our CTArt method obtains a 56.33% in AIR metric, while Huang attains 53.38%. The proposal obtains a 67.60% in NIR metric, in contrast, Huang computes 64.05%. The best average performance in each metric is highlighted in bold.

4 Conclusions

In this paper, a combination of color and texture features for paintings classification by artistic styles is proposed. We found that the combination of color information and texture cues improves the classification rate of artistic paintings by their style.

The proposed color and texture features are extracted from each color component of different color representations. Considering that the resulting feature vector is high dimensional, the use of PCA for feature reduction is proposed.

As a classifier, we propose to use a MLP, which is a widely known classification approach. The evaluation of the proposed CTArt was performed by testing on a challenging database. Quantitative results indicate that our CTArt is robust in discriminating among *6* artistic styles, and it has shown to be more accurate than other state-of-the-art approaches.

Acknowledgments

The authors acknowledge the financial support by the University of Guanajuato and the Mexican Council of Science and Technology (CONaCyT).

References

1. **Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A., Arshad, H. (2018).** State-of-the-art in artificial neural network applications: A survey. Heliyon, Vol. 4, No. 11, pp. 938–979. DOI: 10.1016/j.heliyon.2018.e00938.

2. **Agarwal, S., Karnick, H., Pant, N., Patel, U. (2015).** Genre and style based painting classification. IEEE Winter Conference on Applications of Computer Vision, pp. 588–594. DOI: 10.1109/WACV.2015.84.

3. **Busin, L., Shi, J., Vandenbroucke, N., Macaire, L. (2009).** Color space selection for color image segmentation by spectral clustering. IEEE Int. Conf. on Signal and Image Process. Appl., pp. 262–267. DOI: 10.1109/ICSIPA.2009.5478603.

4. **Condorovici, R. G., Florea, C., Vranceanu, R., Vertan, C. (2013).** Perceptually-inspired artistic genre identification system in digitized painting collections. 18th Scandinavian Conference on Health Informatics, pp. 687–696. DOI: 10.1007/978-3-642-38886-6_64.

5. **Correa-Tome, F. E., Sanchez-Yanez, R. E., Ayala-Ramirez, V. (2011).** Comparison of perceptual color spaces for natural image segmentation tasks. Vol. 50, No. 11, pp. 1–12. DOI: 10.1117/1.3651799.

6. **Culjak, M., Mikus, B., Jez, K., Hadjic, S. (2011).** Classification of art paintings by genre. Proceedings of the 34th International Convention MIPRO, pp. 1634–1639.

7. **Deperlioglu, O., Kose, U. (2011).** An educational tool for artificial neural networks. Computers & Electrical Engineering, Vol. 37, No. 3, pp. 392–402. DOI: 10.1016/j.compeleceng.2011.03.010.

8. **Fairchild, M. D. (2013).** Color Appearance Models. John Wiley & Sons Ltd. DOI: 10.1002/9781118653128.

9. **Goguen, J. (1999).** Art and the brain. Journal of Consciousness Studies, Vol. 6, No. 7, pp. 5–14.

10. **Gombrich, E. H., Gombrich, E. (1995).** The story of art, Vol. 12. Phaidon London.

11. **Gultepe, E., Conturo, T. E., Makrehchi, M. (2018).** Predicting and grouping digitized paintings by style using unsupervised feature learning. Journal of Cultural Heritage, Vol. 31, pp. 13–23. DOI: 10.1016/j.culher.2017.11.008.

12. **Gunsel, B., Sariel, S., Icoglu, O. (2005).** Content-based access to art paintings. IEEE International Conference on Image Processing, Vol. 2, pp. 558–561.

13. **Gupta, P., Saxena, S., Singh, S., Dhami, S., Singh, V. (2012).** Color image segmentation: A state of the art survey. International Journal of Computational Intelligence Research, Vol. 8, No. 1, pp. 17–26.

14. **Huang, Y. F., Wang, C. T., Hsieh, Y. S. (2019).** Relevant feature selection in the context of painting classification. Pattern Analysis and Applications, Vol. 22, No. 4, pp. 1455–1468. DOI: 10.1007/s10044-018-0723-2.

15. **Ilea, D. E., Whelan, P. F. (2011).** Image segmentation based on the integration of colour texture descriptors: A review. Pattern Recognition, Vol. 44, No. 10, pp. 2479–2501. DOI: 10.1016/j.patcog.2011.03.005.

16. **Jolliffe, I. T., Cadima, J. (2016).** Principal component analysis: A review and recent developments. Phil. Trans.R. Soc. A., Vol. 374, No. 2065, pp. 20150202.

17. **Jolliffe, I. T., Cadima, J. (2016).** Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, Vol. 374, No. 2065. DOI: 10.1098/rsta.2015.0202.

18. **Kim, M., Kang, D., Lee, N. (2019).** Feature extraction from oriental painting for wellness contents recommendation services. IEEE Access, Vol. 7, pp. 59263–59270. DOI: 10.1109/ACCESS.2019.2910135.

19. **Liao, Z., Gao, L., Zhou, T., Fan, X., Zhang, Y., Wu, J. (2019).** An oil painters recognition method based on cluster multiple kernel learning algorithm. IEEE Access, Vol. 7, pp. 26842–26854. DOI: 10.1109/ACCESS.2019.2899389.

20. **Lloyd, S. P. (1982).** Least squares quantization in pcm. IEEE Trans. Inform. Theory, Vol. 28, No. 2, pp. 129–137. DOI: 10.1109/TIT.1982.1056489.

21. **Lu, G., Gao, Z., Qin, D., Zhao, X., Liu, M. (2008).** Content-based identifying and classifying traditional chinese painting images. Congress on Image and Signal Processing, Vol. 4, pp. 570–574. DOI: 10.1109/CISP.2008.477.

22. **Markov, K., Matsui, T. (2014).** Music genre and emotion recognition using gaussian processes. IEEE Access, Vol. 2, pp. 688–697. DOI: 10.1109/ACCESS.2014.2333095.

23. **McLaren, K. (1976).** XIII the development of the CIE 1976 (L* a* b*) uniform colour space and colour difference formula. Journal of the Society of Dyers and Colourists, Vol. 92, No. 9, pp. 338–341. DOI: 10.1111/j.1478-4408.1976.tb03301.x.

24. **Ojala, T., Pietikäinen, M., Harwood, D. (1996).** A comparative study of texture measures with classification based on feature distributions. Pattern Recognition, Vol. 29, No. 1, pp. 51–59.

25. **Ojala, T., Pietikäinen, M., Mäenpää, T. (2002).** Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 7, pp. 971–987. DOI: 10.1109/tpami.2002.1017623.

26. **Parsons, M. J. (1987).** How we understand art: A cognitive developmental account of aesthetic experience. Cambridge University Press.

27. **Paul, A., Malathy, C. (2018).** An innovative approach for automatic genre-based fine art painting classification. In Advanced Computational and Communication Paradigms. Springer, pp. 19–27. DOI: 10.1007/978-981-10-8237-5_3.

28. **Ray, S. A. (1978).** Color gamut transform pairs. SIGGRAPH Comput. Graph., Vol. 12, No. 3, pp. 12–19. DOI: 10.1145/965139.807361.

29. **Scaringella, N., Zoia, G., Mlynek, D. (2006).** Automatic genre classification of music content: A survey. IEEE Signal Processing Magazine, Vol. 23, No. 2, pp. 133–141. DOI: 10.1109/MSP.2006.1598089.

30. **Shamir, L., Macura, T., Orlov, N., Eckley, D. M., Goldberg, I. G. (2010).** Impressionism, expressionism, surrealism: Automated recognition of painters and schools of art. ACM Trans. Applicta. Percept., Vol. 7, No. 2, pp. 1–17. DOI: 10.1145/1670671.1670672.

31. **Siddiquie, B., Vitaladevuni, S. N., Davis, L. S. (2009).** Combining multiple kernels for efficient image classification. Workshop on Applications of Computer Vision (WACV), pp. 1–8. DOI: 10.1109/WACV.2009.5403040.

32. **Unser, M. (1986).** Sum and difference histograms for texture classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 1, pp. 118–125.

33. **Zeki, S. (1999).** Art and the brain. Journal of Consciousness Studies, Vol. 6, pp. 76–96.

34. **Zhao, P., Miao, Q., Song, J., Qi, Y., Liu, R., Ge, D. (2018).** Architectural style classification based on feature extraction module. IEEE Access, Vol. 6, pp. 52598–52606. DOI: 10.1109/ACCESS.2018.2869976.

35. **Zujovic, J., Gandy, L., Friedman, S., Pardo, B., Pappas, T. N. (2009).** Classifying paintings by artistic genre: An analysis of features and classifiers. IEEE Int. Workshop Multimedia Signal Process., pp. 1–5.

Article received on 08/08/2021; accepted on 26/08/2022.
Corresponding author is Rocío A. Lizarraga-Morales.

ISSN 2007-9737

Adaptation of Models and Processes for Web-based Development

Melquizedec Moo-Medina[1], Luis Alberto Muñoz-Ubando[2], Rodrigo Mazún Cruz[1]

[1] SEP/Tecnológico Nacional de México (TecNM),
Instituto Tecnológico Superior Progreso, Yucatán,
Mexico

[2] Tecnologico de Monterrey,
School of Engineering and Sciences,
Mexico

{mmoo,rmazun}@itsprogreso.edu.mx, amunoz@tec.mx

Abstract. Currently, technology is a challenge which is rapidly changing the Software Development process, and it is also evolving within the field of Software Engineering itself. In this article, the authors present an adaptation of multiple methodologies and processes in Software Engineering, applied specifically to Web Developments. The Personal Software Process (originally designed by [5]) was developed for the known third generation languages; however, it has been adapted to our current world in such a way that companies are still certified in this process in order to develop software. Throughout twenty-five projects developed using this process combined with (1) Modular Programming, (2) Model View Controller and (3) Agile Methodologies, an adaptation in the process has been achieved in each development. Additionally, the results presented are not only the adaptation to the application in Web Development, but also the incorporation and modification of the activities to carry out to guarantee the quality of the process and the finished product.

Keywords. Software engineering, process innovation, custom software process, controller view model, modular programming.

1 Introduction

The traditional Waterfall Model [11] has been diversifying through light changes that produce new specific methodologies for these application areas when they are incorporated into new Software Development and Engineering technologies. [5] referring to the *Personal Software Development Process* (**PSP**) mentions: The PSP is a personal process that can be adapted to suit the needs of an individual developer[1].

It is not specific to any programming or design methodology; therefore, it can be used with different methodologies, including Agile software development. Software Engineering methods can be considered to vary from predictive to adaptive. PSP is a predictive methodology, and Agile is considered adaptive, but despite their differences, the TSP / PSP and Agile methodologies share several concepts and approaches, particularly regarding team organization.

In this way, we can see that the process established in PSP can be combined with any other methodology, be it traditional or agile, and additionally they can coexist by adapting their processes to development. In this research, a brand-new empiric and experimentally validated proposal is designed and analyzed to improve and innovate within the PSP methodology. The results are then applied to the development process inside the Web environment through an adaptation to the MVC Layered Model (Model View Controller) [2], using modular programming as an agile methodology.

[1] It became a very popular and easy-going methodology supported by intensive efforts towards its teaching and popularization. It can be understood in a nutshell [7]

doi: 10.13053/CyS-26-4-3955

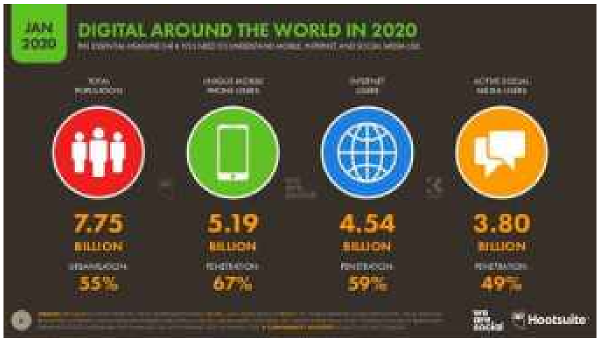


Fig. 1. The use of Digital Development

The objective of this research is to present an improvement made to the activities of the PSP combined with the MVC and using modular programming as an agile methodology on twenty-four projects developed in the web environment, through the analysis of performance, productivity and Cost of quality.

The importance of the project stems from the innovation of applying a traditional process based on counting lines of PSP code to a functional, modular process under the MVC that adapts to a main and majority sector such as Web development. As is seen in 1.

The wearesocial studies carried out in January 2020, reveal worldwide penetration levels in terms of internet users, which are 59%. This represents an increase of 14% over the previous year [9].

In the Web software development process, responsive web design is used, so that users of mobile devices can have access to applications. In the previous graph, it is observed that the population of mobile users stands at 67%, with an annual increase of 28%.

The quality of the software is an issue that in itself generates discussion due to differing points of view.

However, in 1986 [5] undertook a project from where he changed the world of Software Engineering, and its final result is a process that allows the measurement of the Quality of Software Development, and therefore, the quality of the finished product.

It is worth thinking about web development through the basic principle, which states: ”If there was quality during the Software development process, the finished product will have that quality seal as well.” [5].

The advantage of using the proposed methodology is the quality improvement in an incremental and iterative manner; the more it is executed and regulated, the more the process and its deliverables are improved. The times, the sizes, the libraries are enhanced, unleashing a series of benefits which ends in the collection of historical data that will serve for future estimates.

2 Problem Statement

The problem to be solved is the lack and limitations of clear and specific methodologies designed for new technologies that are currently emerging. It is such that different frameworks arise adapting to standards designed for the development of desktop software. Although these are used on the Web, it leaves many gaps in knowledge that are considered in desktop software but not in software for the Web. In this way, it can be said that there is not an appropriate software methodology that can provide a solution to these emerging technologies, such as [10], which controls projects as a solution to these needs.

In their struggle to have a better organization during their projects, many Software development companies use a combination of different methodologies to form new ways of working; however, each one manages to adapt these methodologies in a different way.

3 State of the Art

This paper describes a position about use of the personal software process (PSP) metrics [5]. The position presented describes how and why PSP metrics can be used in teaching and learning about software engineering. PSP can not solve all problems that students and professionals have in developing software, but it can support and guide said software developers in establishing disciplined practices that can be analyzed and improved upon.

The data and metrics provided by PSP, form the basis for an engineering and scientific approach to such an analysis, and they appear to provide more promise for success than any other method or tool on the horizon. These metrics also encourage and support a new, more effective paradigm for education that replaces the programming teacher with a coach that is able to give detailed, specific counselling on how students can improve [3].

The personal software process (PSP) provides software engineers a way to improve the quality, predictability, and productivity of their work. It is designed to address the improvement needs of individual engineers and small software organizations. A graduate level PSP course has been taught at six universities and the PSP is being introduced by three industrial software organizations.

The PSP provides a defined sequence of process improvement steps coupled with performance feedback at each step. This helps engineers to understand the quality of their work and to appreciate the effectiveness of the methods they use. Early experience with the PSP shows an average test defect rate improvement of ten times, and an average productivity improvement of 25% or more [6].

Previous works regarding web design methodologies include the RMM (Relationship Management Methodology), OOHDM (Object Oriented Hypermedia Design Method) and UML Based Web, a model based on techniques from object orientation.

— RMM it was originally introduced in [8] and has since evolved in various ways in response to the rapidly growing demand for hypermedia applications on the World Wide Web. The revamped methodology is demonstrated in rich web application design. It discusses Design and implementation issues, including database integration, and top-down and bottom-up approaches toward developing Web Information System (WIS). The graphical and programming language notations for the new RMM constructs are presented. RMM promotes sound design and the sustainable development of hypermedia.

— OOHDM the object-oriented hypermedia design method is a model-based approach to building large hypermedia applications. It has been used to design different types of applications such as: websites and information systems, interactive kiosks, multimedia presentations, among others. It comprises four different activities: Conceptual Design, Navigation Design, Abstract Interface Design and Implementation. They are carried out in a combination of incremental, iterative and prototype-based development styles. Treating interface, navigation and conceptual design as separate activities allows us to focus on different concerns, one at a time. As a result, we get more modular and reusable designs, and we get a framework for reasoning in the design process, encapsulating the design expertise specific to that activity. Furthermore, interface design primitives can be easily mapped to non-object-oriented implementation environments or languages (such as HTML or Toolbook); thus, OOHDM can be used regardless of whether the target system is a pure one, an object-oriented environment, or even a hybrid (like those we usually find on the Internet) [12].

4 Methods

The process performs estimates of historical data based on linear regression and average standard deviations.

These were initially calculated for more than twenty-five thousand lines of code, more than sixty-two inserted programs in the database as tests, and correcting fifteen internal versions before its first official version in [5].

We began some methodological changes from the traditional Web Development process towards the MVC methodology. This was done by adapting to the PSP process with modular development. The results show innovative changes to the PSP process applied to Web pages, as well as the solution to problems found for the adaptation of the MVC methodology and Modular development.

To work on the proposal, we have the development of twenty-four modules created under

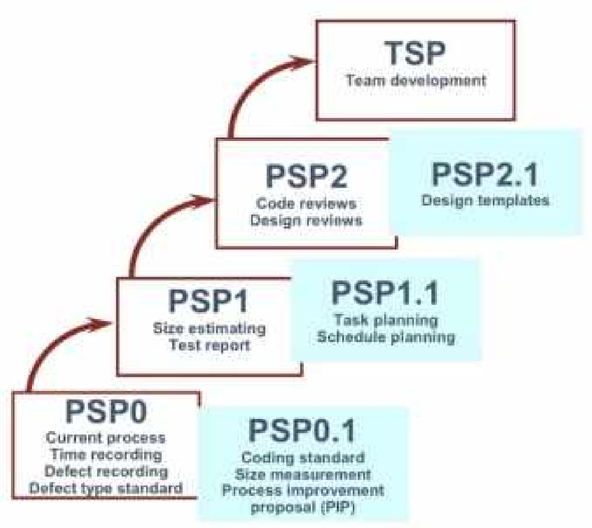


Fig. 2. The traditional phases of software development

the Web scope and more than one hundred and seventy delta hours in the PSP process in version 2.1.

4.1 Historial Facts

The analysis of the implementation is found in 1, compiled from historical data developed in the methodologies and processes submitted in the initial section of the proposal.

4.2 Testing Methods for Size

Taking into account the development history outlined in Table 1, when generating a new module in the project, an example of the estimations of this module is shown to the general project of the system and it is explained how the estimation is carried out.

The calculation of the estimation of code lines to be programmed for a 183 LOC module (Lines of Code) is 206 LOC in the estimation method B which is based on the planning data and 237 LOC in the estimation method A based on historical data. These have a deviation of 88.2 and 90.4 respectively, as shown in 2.

This indicates that method B is the best option to select, due to the precision of the historical data entered in the planning stage that ensures a better estimate in the development of the next module.

4.3 Estimated Productivity

The productivity estimate for planned development sizes and times is 42.2 LOC / Hr, which is consistent to the current date with a productivity of 35.2 LOC / Hr. (± 19.5).

4.4 The Value Proposition of this Paper

The proposal is based on the modification of the traditional PSP process that was initially developed for third generation languages, inserting key tasks and activities that adjust to this process and adapt it to Web Development.

A key factor will be the correct interpretation of historical data that allow estimations on the scope of the quality of the software. Different methodologies will be used for this, such as the MVC, Modular programming and agile methodologies that bring the user or client closer to the Development process.

The expected outcome will be a new process of Software Development for the Web field which is both innovative and functional.

4.5 Proposed Methodology

To start the adaptation to the Development process, you must begin with the application of the PSP 0, 0.1, 1, 1.2, 2 and 2.1 successively, as shown in 2. During the adaptation of the process, changes that have a direct impact have been added to web application development.

In PSP, all the tasks and activities that the software engineer must carry out during the development process of a software product are specifically defined in a set of documents known as scripts. These must be followed in a disciplinary way since the success of the improvement sought out will depend on it. (The Personal Software Process (PSP).

To carry out the proposal, the PSP 0 process is applied first, which consists of developing the software on a daily basis, only increasing the recording of development times on the Waterfall methodology. The data collected will serve for

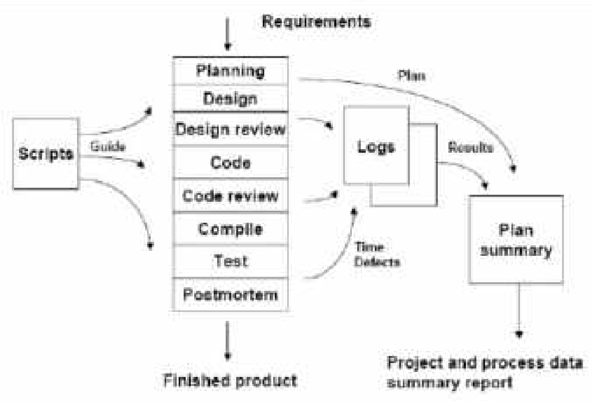


Fig. 3. The PSP phases of software development

future estimates. At this stage, it is optimal to create changes in the design stage by adding the Web design (Web View Design) that consists in the generation of the HTML tagging already with the responsive styles of bootstrap and CSS3, or according to the Design methodology.

In the MVC, this change consists of producing, in the project's View folder, the Web interface without functionality and presenting it to the user for approval (User Review).

Before moving on to coding, the changes requested by the user will be modified in the design stage. This single change will serve to generate confidence and concentration in the use of time control.

At the end of the Compile stage, the generated module will be presented to the client again for approval as part of the Test.

The next step is to apply the PSP 0.1 process which allows us to add the code line count in the development process. The objectives of PSP 0.1 are: ”to measure the size of the programs that are produced, to perform the size counting for the programs and, to make precise and exact measurements of size”.

(Using PSP 0.1) In order to approach the number of functions that will be carried out and get a more accurate count; in this step, it is recommended to both add the draft of the module that will be implemented (Sketchboard Web) in the requirements and obtain the ER diagram with the functionalities in the requirements survey stage.

Likewise, it is suggested to add the unit tests of the Programmer (Unit Test of Coder) in the coding stage. This consists of verifying the functionalities programmed during this stage on localhost.

In PSP 1, the personal planning process is added, which consists of establishing an orderly and repeatable procedure to develop software size estimates. (Using PSP 1, 2006).

At this stage, it is suggested to add the technical conceptual design of the software. This will help demonstrate the relationship of the files that will be used in the development, and assist in organizing the MVC and its functions.

In this case, it can also be adapted to the use of a Development Framework.

The next step of Growth is the use of PSP 1.1 which aims to introduce and practice methods to make resource plans and schedules by tracking performance against these plans, and establishing probable project completion dates.

At this stage, it is suggested to also add to the process the synchronization of the Database and the source files to the server during the compilation stage in a designated section for tests, so that in turn it is validated by the user in the Test stage.

In the PSP 2 stage, two stages presented in the PSP numbering are inserted, which aims to increase the quality of the development by looking for injected defects in the design and coding stages.

At this stage it is suggested to verify the design and coding standards developed in the PSP 0.1 process [13].

In the PSP 2.1 stage, Design combined with UML is added in the Development process.

At this stage, additional measures are introduced to manage the quality of the process, as well as design templates that provide an orderly framework and format for recording designs.

The following list shows the elements added in this stage:

1. **PSP** 2.1 design review script,
2. **PSP** 2.1 design review checklist,
3. Operational specification template,
4. Functional specification template,

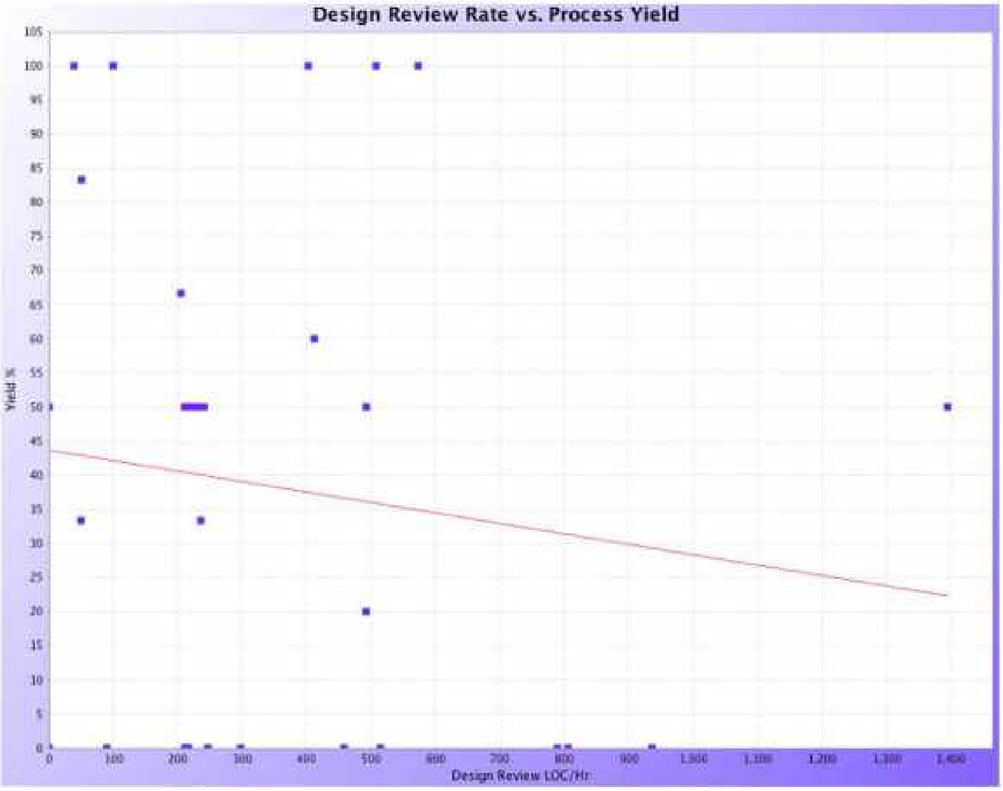


Fig. 4. Experimented process

5. State specification template,

6. Logic specification template.

The deliverables that have been made as an improvement in this stage for the Web design are the Use Case Diagram and Specification, the sequence diagram, the State diagram, the Relational Diagram, the Data Dictionary and the Database Design Data.

In summation, the activities and the deliverables, forms, templates or standards requested by PSP in its different versions as described in this section, are shown in 4.

Specifically, in version 2.1, all the deliverables of the other versions can be verified.

Table 4 shows where the deliverables should be prepared according to the PSP version chosen.

In Table 5, it is observed that, starting from the previous table and with the stages already classified, the deliverables added to ensure functionality in Web development and its quality are shown on the right side of the table.

5 Experimental Results

The main objectives of PSP are: maximize software quality, achieve a discipline of continuous improvement in the development process, improve the quality of the development process and increase productivity. Considering that quality is based on measurement, PSP integrates a set of forms, which are also used by PSP to collect the necessary metrics [1]. There are three basic metrics that PSP establishes such as size, time, defects, and the others are measurements derived from these basic metrics. Defects is a metric related to software quality and using PSP can reduce defects as stated in [1].

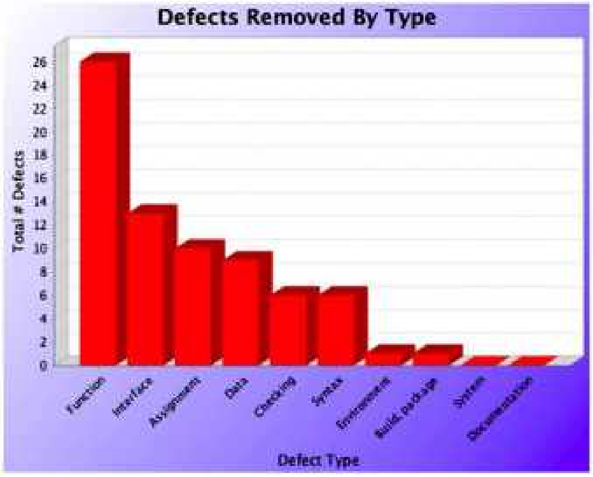


Fig. 5. Failure inventory

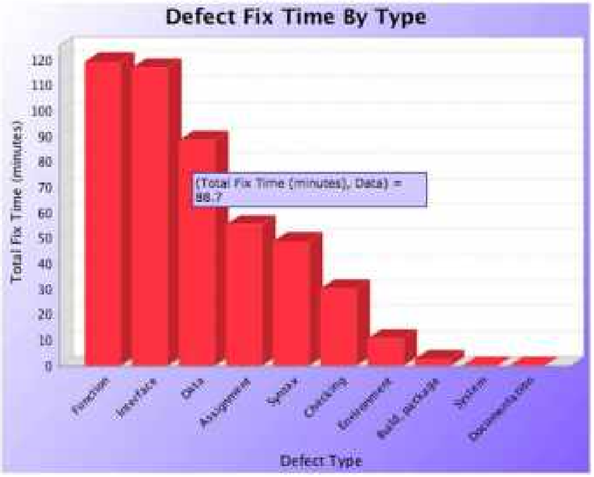


Fig. 6. Time lag for failure solving

5.1 Performance Results

In the PSP, performance is measured in two ways: the percentage of total defects found and removed in a stage, and Process Yield, which refers to the percentage of defects removed before the first test, compilation and testing 3.

In 4, it is presented the statistics of the Performance obtained in the projects during the design review stage with a negative slope from 45% to 22%, this indicates that the number of defects decreases in the progress of the projects in the Design review, and more defects are repaired before reaching the testing stage.

5.2 Results of Defect Classification

The results of the total types of defects injected into the system in the development of twenty-four projects in the Web field are shown in 5 and 6.

In the graph on the left, it can be seen that the number of defects per interface is less than the functional one, while on the graph on the right it can be seen that the time to repair defects in interfaces is as high as the functional one with an approximate time of 120 minutes. This shows that there are few repairs in the Interfaces, but with a sum of minutes equal to the repair time of the functionality.

The question that arises when observing the graphs is: Why does it take longer to make a repair in Interface than in Controllers or Models? This increase is derived from the adaptation of three activities during the design review:

— User Review.

— Modify to Web View.

— Acceptance User.

These three activities are not included in the traditional PSP and are necessary for web development, since they integrate the end user or client in the design review stage.

Here, they accept the Web view or request modifications to be included in the requirements. The Modify to Web View activity is optional only if modifications are required. These elements are a part of agile methodologies.

The advantage of completing these activities is that when coding begins, there is already a design that is accepted and endorsed by the user. Therefore, it would be very difficult to find a design error on something that has already been double validated, resulting in a beneficial performance of 4.

At the same time, this helps reduce possible design defects in the testing stage and, as in Figure 4, performance increases due to the decrease in defects to be repaired during the testing stage.

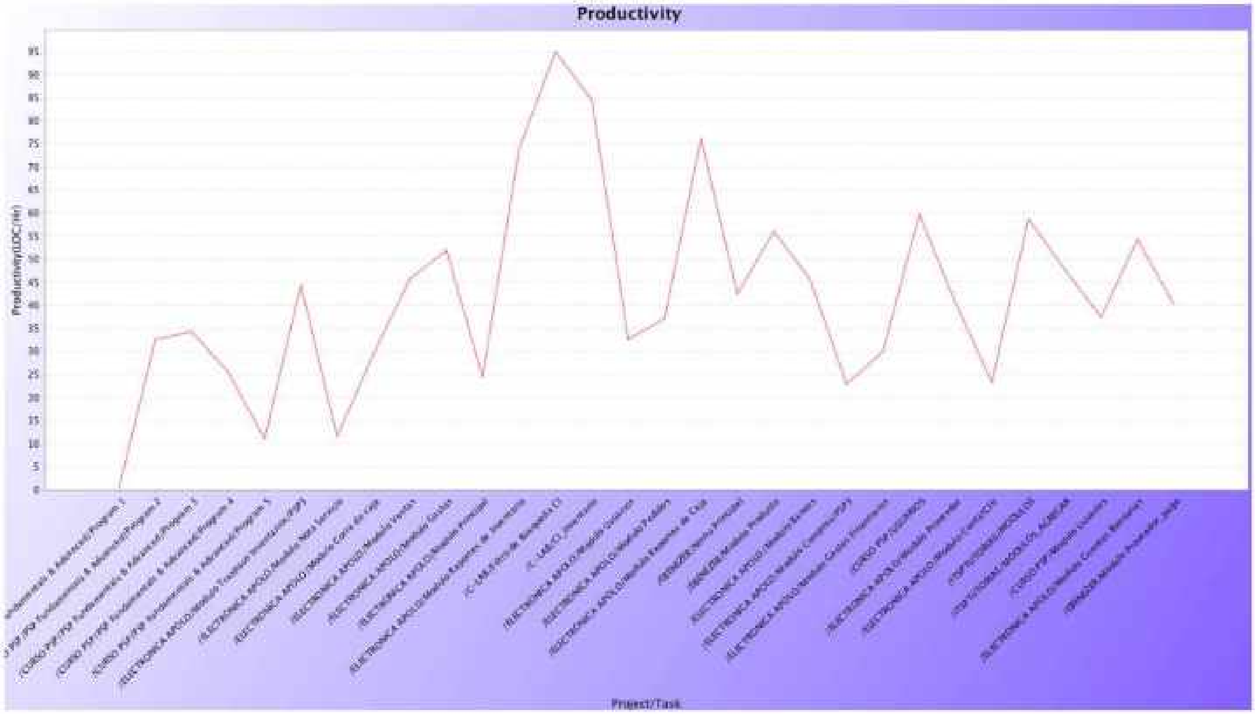


Fig. 7. The productivity improvement

Both 4, 5 and 6 support the quality of the software due to the improvement in performance.

5.3 Productivity Results

Productivity in PSP is measured by the number of Lines of Code (LOC) developed per hour. In Figure 6, it can be observed how average productivity went from 30 LOC / Hr. in the first program, to 40 LOC / Hr. and has a positive slope.

In the same way, a difference is observed between the less complex modules which raise productivity above 50 LOC / Hr. and more complex ones up to 40 LOC / Hr. Likewise, an exception is observed in the CI search filter module, which had a productivity of close to 90 LOC / Hr. due to a change in user requirements during the design stage.

Productivity allows for a self-evaluation to be carried out each time a project is completed, from which it can be concluded whether there is an improvement in the production process.

5.4 Quality Cost Results

The failure rate relationship measures the quality of the engineering process, using quality cost parameters, as stated by the application of Juran's quality improvement program. This is presented to assist software engineering management in the identification and control of quality costs [4]. Evaluation quality cost is the percentage of development time spent on quality evaluation activities.

From project 6, we observed an increase in the cost of quality due to the increase in defects in the testing stages or final stages of development. In the last 10 projects, a slight decrease in the cost of quality was noted due to the decrease in defects in the final stages, which mostly occurs when the user is included in the Review Design stage.

6 Discussion

According to the results obtained, it was possible to observe an improvement in the quality of

Table 1. The list of projects

Project/Task	Estimated Proxy Size	Planred Added & Modified Size	Actual Added Modifield Size	Estimated Hours	Actual Hours
/EBENEZER/Modulo Proveedor_pago	117	147	NA	3.65	4.1
/ITSPTUTORIAS/MODULOS_AGREGAR	94.4	121	105	2.98	2.2
/CURSO PSP/Modulo Usuarios	240	259	215	7.13	5.75
/ELECTRONICA /Modulo Trapaso Inventarios/ PSP3	120	131	312	5.55	7.03
/ELECTRONICA /Modulo Nota Servicio	135	191	99	6.12	8.55
/ELECTRONICA /Modulo Corte de caja	314	346	404	14.8	13.7
/ELECTRONICA /Modulo Ventas	387	478	463	15.2	10.1
/ELECTRONICA /Modulo Gastos	215	264	263	9.72	5.07
/ELECTRONICA /Modulo Principal	193	233	194	8.72	7.95
/ELECTRONICA /Modulo Reportes de Inventario	167	207	457	6.13	6.18
/C-LAB/Filtro de Busqueda CI	253	321	296	9.8	3.12
/C-LAB/CI inventario	292	361	252	2	2.98
/ELECTRONICA /Modulo Usuarios	163	180	144	5.7	4.42
/ELECTRONICA /Modulo Pedidos	84.7	89.4	94	2.56	2.53
/ELECTRONICA /Modulo Reportes de Caja	240	307	250	8.6	3.28
/EBENEZER/Menu Principal	81	84.8	109	2.46	2.57
/EBENEZER/Modulo Producto	278	325	446	8.45	7.97
/ELECTRONICA /Modulo Bancos	254	316	425	8.0	9.3
/ELECTRONICA /Modulo Compras/ PSP3	236	309	322	10	14.1
/ELECTRONICA /Modulo Gastos Financieros	196	221	184	5.84	6.15
/ELECTRONICA/ Modulo Cuentas Bancarias	131	163	60	3	NA
/CURSO PSP/Usuarios	201	219	230	6.26	3.85
/ELECTRONICA /Modulo Proveedor	168	221	328	4.78	8.1
/ELECTRONICA /Modulo ContarClic	48.5	87.5	35	1.76	1.5
/ITSTUTORIAS/MODULOS	90	113	181	2.92	3.08

Table 2. Load/Size times

Method	Estimate	r^2	*Beta0*	*Beta1*	Range(70%)	LPI	UPI	Variance	StdDev
B	206	0.53	22.6	1.0	95.6	110	301	7785	88.2
A	237	0.5	23.9	1.16	98.0	139	335	8174	90.4
C2	204		0	1.11					
C1	240		0	1.31					

Table 3. Estimated executing times

Method	Estimate	r^2	*Beta0*	*1/Beta1*	Range(70%)	LPI	UPI	Variance	StdDev
C2	5.63		0	32.6 LOC/Hr					
C1	6.61		0	27.7 LOC/Hr					
C3	5.2		0	35.2 LOC/Hr					
A	6.51	0.4	1.15	34.2 LOC/Hr	3.03	3.48	9.54	7.8	2.79
B	5.72	0.4	1.26	41.1 LOC/Hr	3.04	2.68	8.76	7.86	2.8

development, the process and the results of the process by combining the methodologies of PSP, Modular Programming, MVC and agile methodologies. All of which contributed to increasing the quality of Software Development.

The primary methodologies such as Waterfall and Spiral, were used in principle for structured development and are evolving to adapt to new technologies, methodologies and Development processes. This combination of processes is not intended to replace Waterfall or Spiral methodologies, but take them to an updated Web environment with better performance.

7 Conclusion

Throughout the development of each project, it was observed that there is an increment in performance by increasing the number of defects found in the Design and Design Review stages. Additionally,

Table 4. Phases proposed

Process Version	**PSP0**	**PSP0.1**	**PSP1**	**PSP1.1**	**PSP2**	**PSP2.1**
Process Scripts and Summaries						
PROBE Estimating Script			x	x	x	x
Forms, Templates, Standards, and Instructions						
Time Recorging Log	x	x	x	x	x	x
Defect Recording Log	x	x	x	x	x	x
Defect Type Standard	x	x	x	x	x	x
PIP		x	x	x	x	x
Coding Standard		x	x	x	x	x
Test Report Template			x	x	x	x
Size Estimating Template			x	x	x	x
Task Planning Template				x	x	x
Schedule Planning Template				x	x	x
Design Review Checklist					x	x
Code Review Checklist					x	x
Use Case Specification Template						x
Functional Specification Template						x
State Specification Template						x
Logic Specification Template						x

Table 5. Phases proposed

Stage	**PSP 2.1 Standard**	**Web Development**
Requirements	Documented requirements statement	
		Sketchboard web
		ER-Diagram
Planning	Project Plan Summary form	
	Size Estimating template	
	Task and Schedule Planning templates	
		Program conceptual design
	Estimated defect data	
	Time and size prediction intervals (PROBE)	
Design	Functional Specification	
	Operational Specification	
	Use Case Diagram and specification	
	Sequence Diagram	
		StateCharat Diagram (se omite para diseños web)
		Relational Diagram Complete
		Dictionary Data
		Data Base Design
		Screenshots of Web View design
Review Design	Design Review Checklist	
		User Review
		Modify to Web View
		Acceptance User

Table 6. Phases proposed

Code	Coding Standard Review	
	Code	
		Unit Test of Coder
Code Review	Code Review Checklist	
		Coding Standard Verifing
		Unit Test of Coder
Compile		Server Config
		Sources synchronization
		DataBase synchronization
Test	Test Report Template	
		integral Test of User
	Test Result	
Postmortem	Time Recording Log	
	Defect Recording Log	
	Defect Type Standard	
	PIP	
	Size Estimating Complete	
	Schedule Planning Complete	
	Task Planning Complete	

productivity was increased by increasing the number of lines of code written per hour, and the cost of quality was reduced by minimizing the number of defects that make it to and beyond testing.

It can therefore be concluded that establishing these new and refined stages during the PSP phase presents a clear adaptation to the Web Software Development process.

This contributes to setting up basic improvements throughout the application in the modular projects, and promotes the quality of the software process and the finished product.

It is recommended that those who wish to join these processes have previously used PSP and some Web technology. Because this process is a Web Software Development methodology, and is not intended to assist in learning Website development languages.

References

1. **Chavarria, A. E., Ore, S. B., Pastor, C. (2016).** Quality assurance in the software development process using CMMI, TSP and PSP. Revista Iberica de Sistemas e Tecnologias de Informacao, Vol. 20, pp. 62–77. DOI: 10.17013/risti.20.62-77.

2. **Deacon, J. (2000).** Model-view-controller (MVC) architecture. John Deacon Computer Systems Development, Consulting & Training.

3. **Hilburn, T. B. (1999).** PSP metrics in support of software engineering education. Proceedings 12th Conference on Software Engineering Education and Training (Cat. No.PR00131), pp. 135–136. DOI: 10.1109/CSEE.1999.755194.

4. **Hollocker, C. P. (1986).** Finding the cost of software quality. IEEE Transactions on Engineering Management, Vol. EM–33, No. 4, pp. 223–228. DOI: 10.1109/TEM.1986.6447683.

5. **Humphrey, W. S. (1989).** Managing the Software Process. Addison-Wesley Longman Publishing Co., Inc.

6. **Humphrey, W. S. (1994).** The personal process in software engineering. Proceedings of the Third International Conference on the Software Process. Applying the Software Process, pp. 69–77. DOI: 10.1109/SPCON.1994.344422.

7. **Humphrey, W. S., Over, J. W. (1997).** The personal software process (PSP) a full-day tutorial. Proceedings of the (19th) International Conference on Software Engineering, pp. 645–646.

8. **Isakowitz, T., Stohr, E., Iyer, B. (1995).** RMM: A methodology for structured hypermedia design. ACM, Vol. 38, No. 8, pp. 34–44. DOI: 10.1145/208344.208346.

9. **Kemo, S. (2020).** Global digital overview. We are social. Accessed 2020-2-2.

10. **Markovtsev, V., Long, W. (2018).** Public Git archive: A big code dataset for all. Proceedings of the 15th International Conference on Mining Software Repositories, pp. 34–37. DOI: 10.1145/3196398.3196464.

11. **Riddle, W., Williams, L. (1986).** Modelling software development in the large. 3rd ISPW, IEEE Computer Society, pp. 81–84.

12. **Schwabe, D., Rossi, G., Barbosa, S. D. (1996).** Systematic hypermedia application design with OOHDM. Proceedings of the the Seventh ACM Conference on Hypertext, pp. 116–128.

13. **Suranto, B. (2014).** PSP and PQI: How do they improve individual software process. Teknoin, Vol. 20, No. 4. DOI: 10.20885/teknoin.vol20.iss4.art1.

Article received on 29/04/2021; accepted on 27/08/2022. Corresponding author is Melquizedec Moo-Medina.

ISSN 2007-9737

Numerical Technique for Implementation of SDBD Plasma Actuators for Flow Control Applications in Wing Surfaces

Raúl Bernal Orozco[1], Oliver Marcel Huerta Chávez[1], José Ángel Ortega Herrera[2], Alfredo Arias Montaño[1]

[1] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Ticomán,
Sección de Estudios de Posgrado e Investigación,
Mexico

[2] Instituto Politécnico Nacional,
Escuela Superior de Ingeniería Mecánica y Eléctrica Zacatenco,
Sección de Estudios de Posgrado e Investigación,
Mexico

{rulozco, lecram_21, oeha430210}@hotmail.com
alfredo.ariasmontano@gmail.com

Abstract. In the subsequent study a Computational Fluid Dynamics (CFD) analysis technique to simulate a plasma actuator over an airfoil Re = O(20^5) is presented. The technique uses a two-dimensional Reynolds-Averaged Navier-Stokes Method coupled to the Kloker plasma-fluid model to study the effects of a Single Dielectric-Barrier Discharge (SDBD) as a Plasma Actuator. The CFD technique have been implemented in OpenFOAM platform for two setups, when: i) the actuator was located at x/c = 0.03 and ii) x/c = 0.1 of the chord length of the airfoil. The magnitude of the body force is equivalent to the results obtained by Hofkens and the actuator operates for both cases in the continuous and in the burst mode. To perform the numerical technique for a stable solution in OpenFOAM, various numerical procedures were tested, including a mixed solver between PISO and SIMPLE algorithm better known as pimpleFoam with nCorrector. The cases were solved in parallel on distributed processors using OpenMPI implementation and the accuracy of the results are strongly depends on the choice of grid size, y-plus, wall function and discretization scheme. The results indicate a high potential, suitability and great capabilities of this numerical technique implemented in OpenFOAM platform for free instability flow simulation.

Keywords. SDBD, airfoil with plasma actuator, flow separation control, low reynolds.

1 Introduction

The control of flow separation has been studied since the performance of aerodynamic bodies is affected by the separation. Passive such as dimples and vortex generator, are chosen by its simplicity.

Although it cannot control the flow well situations that are beyond design limits of the device and under dynamic conditions.

On the other hand, active flow control devices have also been investigated and developed in order to overcome the shortcomings of passive flow control devices.

Active flow control techniques like blowing and suction have been studied for more than 50 years, and its effectiveness for a separated flow control has been demonstrated.

But these devices have serious drawbacks due to its complexity and the addition of weight which limits its application.

The periodic excitation, a type of active flow control, have contributed to better understand the separation phenomena, and the development of linear stability theory for shear flows had led to the invention of modern active flow control techniques.

doi: 10.13053/CyS-26-4-3424

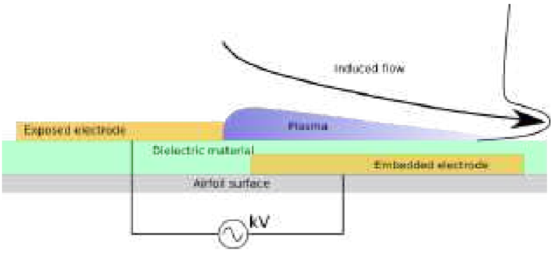


Fig. 1. SDBD scheme

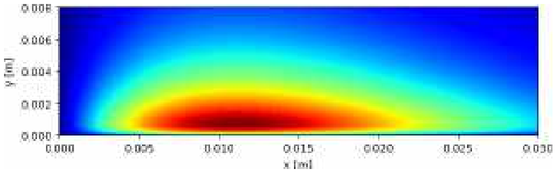


Fig. 2. Extension of the body force for an actuator 0.03x/c length

Modern flow control techniques using unsteady actuation require even less actuator power than the steady boundary layer control techniques (of the order of 10^{-1} to 10^{-2}) to modify the external flow fields [10].

In recent years to overcome the disadvantages of more traditional flow control devices, pulsed micro-jets [11], synthetic jets and Single Dielectric Barrier Discharge (SDBD) plasma actuator are being investigated. Plasma actuators have different applications in aerodynamics, they have been used to reattach the flow over an airfoil as did by Corke [3].

Huang [9] used them to control the separation in low-pressure turbine blades. Other use is to enhance the performance of wind as studied by Greenblatt [7]. In the present work a technique to simulate a plasma actuator for flow control over an airfoil is presented.

The plasma actuator is implemented by means of a body force source term which is directly introduced into the momentum equation. We studied the flow control in a NACA 0015 airfoil with a SDBD plasma actuator at Reynolds number (Re) of 20^5.

Plasma actuators have great advantages, such as responsivity, low weight, a simple structure, and a low energy consumption. For this work we will focus the results to show the developed technique capability to be used for flow control and it is emphasized the usage of `fvOptions` to implement the DBD force in the simulations, since in a previous work [2] we had performed a deeper analysis on the flow control with a periodic excitation by SDBD plasma actuator.

An SDBD actuator is a device that through an electrical discharge ionizes the air and by the acceleration of the ions by the electromagnetic forces there is a transfer of momentum inside the boundary layer which induces a jet inside the boundary layer, as seen in schematic diagram of Fig. 1, this causes a decrease and delay of the flow separation.

To simulate the propulsive force from the actuator, several computational models that seeks to accurately reproduce the properties of plasma, such as ion formation rate, and particle collisions. These models entails an increase in the computational cost making them unpractical for engineering applications.

The other type has an engineering focus such as the Suzen model [15] or Shyy model [14], these are based on the principle that the body force is directly proportional to the product of the charge density by the electric field.

However, these models have some drawbacks such as the electrodes should be strictly modeled into the domain since the thickness and position of these are necessary for the equations that determine the electric field, in addition to adding electric equations to the solver.

To avoid that disadvantages Dörr and Kloker [4] have used a technique in which the body forces are modeled as source terms. Kloker states that a SDBD can be modeled as a body force parallel to the wall. In the work done by Sato [13] he used the Suzen model to calculate the magnitude of the force that is applied as a source term directly in the momentum equation.

Plasma actuators have two modes of operation, they can work in a continuous mode in which it is continually activated, or through pulses as in the so-called burst mode. It has been proved that an actuator operating in the burst mode is more energy efficient, than an actuator in the continuous mode. [16, 13, 1, 6].

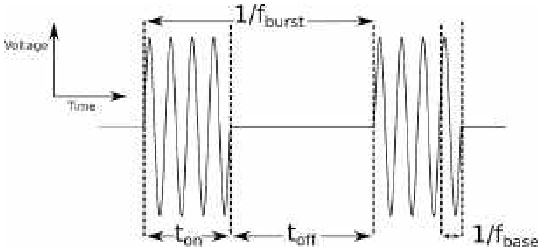


Fig. 3. Signal scheme for burst mode

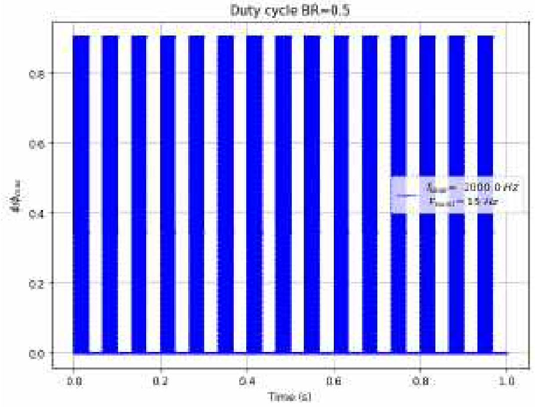


Fig. 4. Duty cycle for $F_{\text{burst}} = 15$ Hz and $f_{\text{base}} = 2$ kHz

2 Modeling of the Flow and the Actuator

2.1 Flow Modeling

The flow field around the plasma actuator an the airfoil is described by the Reynolds-averaged Navier–Stokes equations (RANS) equations. In this work consider the flow as incompressible, the change in temperature is neglected, as it is known most of the energy that the plasma actuator consumes is transformed into kinetic energy, this leads to the required equations are the continuity and momentum equations:

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{p}}{\partial x_j} + \frac{\partial}{\partial x_j}\left[(\nu + \nu_t)\frac{\partial \bar{u}_i}{\partial x_j}\right] + \frac{f_b}{\rho}, \tag{2}$$

where p is the pressure, u the speed, ρ the density, ν the kinematic viscosity, ν_t the eddy viscosity, and f_b the body force from the actuator. Since to fully resolve the Navier-Stokes equations would require much time and computational resources, a turbulence model is required. In this work we used the $k-\omega$SST model.

This model uses a $k-\omega$ formulation in the inner part of the boundary layer from the wall to the viscous sub-layer. This formulation changes the behavior to $k-\epsilon$ model in the free current, leading to the advantage of avoiding the problem of over sensitivity, of the $k-\omega$ model to the turbulence intensity in the free current.

It is a model capable of capturing separation. The $k-\omega$SST model is a lineal model, in which the Reynolds stress are modeled by the Boussinesq hypothesis that is a linear relationship:

$$-\rho\langle u_i u_j\rangle = 2\mu_t S_{i,j} - \frac{2}{3}\rho k \delta_{ij}, \tag{3}$$

where μ_t is the eddy viscosity, k the turbulence kinetic energy and $S_{i,j}$ the strain rate tensor. Then the transport of energy from the turbulence fluctuations is modeled through the eddy viscosity. It is a two-equations model, with one for turbulent kinetic energy equation (4), and one for specific dissipation equation (5):

$$\begin{aligned}\frac{\partial k}{\partial t} + U_j \frac{\partial k}{\partial x_j} &= \tau_{ij}\frac{\partial u_i}{\partial x_j} - \\ &\beta^* \omega k + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_{k1}\nu_t)\frac{\partial k}{\partial x_j}\right],\end{aligned} \tag{4}$$

$$\begin{aligned}\frac{\partial \omega}{\partial t} + U_j \frac{\partial \omega}{\partial x_j} = \frac{\gamma_1}{\nu_t}\tau_{ij}\frac{\partial u_i}{\partial x_j} - \beta^* \omega^2 \\ + \frac{\partial}{\partial x_j}\left[(\nu + \sigma_\omega \nu_t)\frac{\partial k}{\partial x_j}\right] \\ + 2(1-F_1)\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial t x_i}\frac{\partial \omega}{\partial x_i}.\end{aligned} \tag{5}$$

The turbulent viscosity ν_t in the model $k-\omega$SST is described by the equation (6):

$$\nu_t = \frac{a_1 k}{\max(a_1\omega, b_1 F_{23} S)}. \tag{6}$$

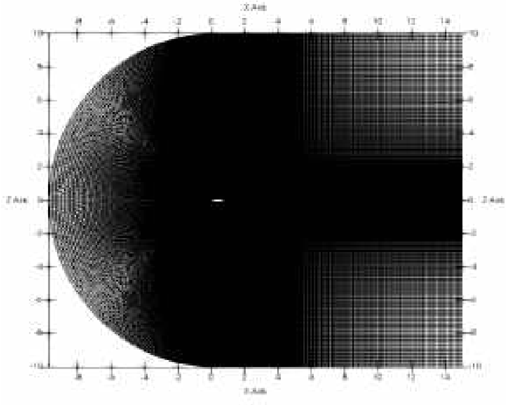


Fig. 5. General aspect of the mesh

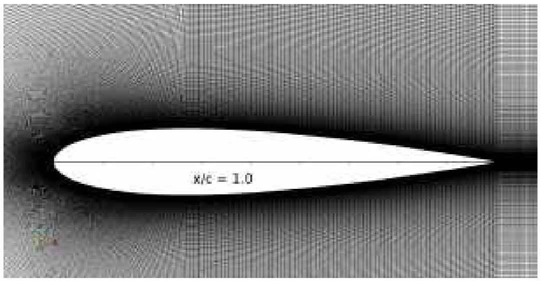


Fig. 6. Mesh detail

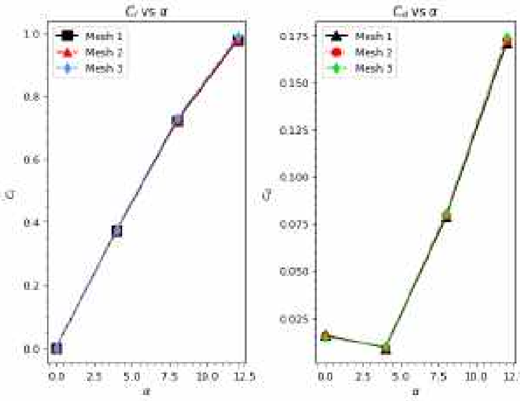


Fig. 7. C_l and C_d convergence

2.2 Actuator Modeling

As mentioned in the previous subsection, the body forces provided by the plasma actuator are modeled by adding the source term, f_b into the momentum equation (2).

In this work we used the Kloker model [4] to obtain the the plasma extent over the wall, by using the equation (7), we obtained that the body force extends from the wall up to a distance of 0.003x/c, in the Fig. 2 the extension of body force is shown:

$$f(x,y) = c_x \left[\left(a_0 a_1 x + a_0^2 a_2 x^2\right) e^{-a_0 x}\right] \cdot \left[\left(b_1 y + b_2 y^2\right) e^{-b_0 y^{2/5}}\right]. \tag{7}$$

Then the magnitude of the body force was selected from the results obtained by Hofkens [8] in which he determined the strength of a plasma actuator from velocity field data.

To obtain the unsteadiness of the body force, we multiply the body forces by the square of the sine function, as can be seen in the equation (8):

$$f_b(x,y,t) = f(x,y)\sin^2\left(2\pi f_{\text{base}}\, t\right), \tag{8}$$

where f is the force magnitude over the space, f_{base} is the base frequency, t is the time, and f_b is the unsteady body force. This model is based on the assumption that the temporal variation of the body force can be characterized as push/push (Font [5]). As last step of modeling it is required to model the burst mode.

The burst mode is produced by a signal generated by the multiplication of low-frequency square signal with a high-frequency sinusoidal base signal, that way the actuation is periodically switched on and off. In Fig. 3 is shown a schematic of the burst mode, and in Fig. 4 a plot of the signal generated by a burst frequency of 15 Hz and a base frequency of 2 kHz is shown.

The burst mode is controlled by the burst ratio equation (9), and the burst frequency. If the burst ratio is equal to one (BR=1.0) the actuator is operating in the continuous mode. The burst actuation has the advantage for separation control, as some periodic excitation controls, the boundary layer receptivity, acoustic disturbances, coherent vortex shedding:

$$BR = \frac{t_{\text{on}}}{t_{\text{on}} + t_{\text{off}}} = \frac{t_{\text{on}}}{T}. \tag{9}$$

Table 1. Mesh properties

Parameter	Mesh 1	Mesh 2	Mesh 3
nodes	629,700	775,500	104,7340
cells	313,650	386,400	522,000
max aspect ratio	306.773	309.543	277.867
max non-orthogonality	29.171	29.5681	29.6446
max skewness	0.45114	0.45267	0.45037

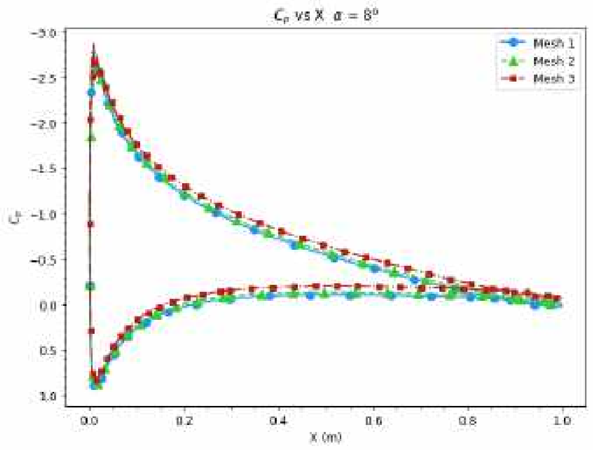


Fig. 8. C_p convergence for a case at 8°

where the active time for each pulse t_{on} is defined by equation (10) an the total time T of each pulse by equation (11):

$$t_{\text{on}} = \frac{BR}{f_{\text{burst}}}, \tag{10}$$

$$T = t_{\text{on}} + t_{\text{off}} = \frac{1}{f_{\text{burst}}}. \tag{11}$$

The dimensionless base frequency $f^+ = f_{\text{base}} L/U_\infty$ and dimensionless burst $F^+ = f_{\text{burst}} L/U_\infty$, where L is the length reference (airfoil chord) and U_∞ is the free stream velocity.

2.3 Computational Cases

The cases were divided in two setups in the first setup, the actuator was located at 3% (x/c=0.03) and 10% (x/c=0.01) of the chord length from the leading edge, then BR=1.0 and the $f_{base} = 2$ kHz. In the second setup the actuator is placed at 3% (x/c=0.03) and it operates both in the continuous mode and in the burst mode, to clarify the control mechanisms of direct momentum addition. In the burst mode cases, the BR was set to 0.5 for all the burst mode cases, the burst frequency (f_{burst}) was set to 3, 5 and 15 Hz, the base frequency (f_{base}) was set to 2 KHz for all the cases both in continuous and burst mode.

See Table 2 for the first setup, and Table 3 for the second setup of cases. The maximum force magnitude was $416\ \text{N}/\text{m}^3$ for all the cases, for comparison usually it is used the momentum coefficient (C_μ), this is defined by:

$$C_\mu = BR\frac{F_{b,tot}}{1/2U_\infty^2 L}, \tag{12}$$

where $F_{b,tot}$ is calculated as

$$F_{b,tot} = \int_0^\infty \int_0^\infty f_b \, dx \, dy. \tag{13}$$

This gives us a momentum coefficient $C_\mu = 2.89 \times 10^{-3}$ for a BR=1.0, and $C_\mu = 1.44 \times 10^{-3}$ for a BR=0.5.

3 Computational Setting Up

3.1 Numerical Grid

A structured C-type mesh was generated with the OpenFOAM utility `blockMesh`, Fig. 5 and Fig.6 shows the mesh. The airfoil has a chord of 1.0=x/c, the mesh has a length of 25x/c and a height of 20x/c. Three meshes were used to perform a mesh independence analysis, the properties of each mesh are shown in the Table 1.

The `blockMesh` utility generates parametric meshes with layering growth and curved edges. `blockMesh` works by breaking down the main geometry into a set of one or more three-dimensional hexahedral blocks. The mesh manifests as several cells in each direction of the block. Each block of the geometry is defined by eight vertices.

To use `blockMesh` a `blockMeshDict` file is needed, this file is in the `case/system` directory, this file contains the definition of the vertices, blocks, layering growth, block merging and boundaries.

doi: 10.13053/CyS-26-4-3424

Table 2. Cases for the study of the position of the actuator

$\alpha°$	Position	F^+	f_{burst}	f^+	f_{base}	BR	C_μ
$0°$ - $16°$	0.03x/c	Continuous	-	660	2.0 kHz	1.0	$C_\mu = 2.89 \times 10^{-3}$
$0°$ - $16°$	0.10x/c	Continuous	-	660	2.0 kHz	1.0	$C_\mu = 2.89 \times 10^{-3}$

Table 3. Cases for the study of the frequency of the actuator

$\alpha°$	F^+	f_{burst}	f^+	f_{base}	BR	C_μ
16	1	3 Hz	660	2.0 kHz	0.5	$C_\mu = 1.44 \times 10^{-3}$
16	5	15 Hz	660	2.0 kHz	0.5	$C_\mu = 1.44 \times 10^{-3}$
16	15	45 Hz	660	2.0 kHz	0.5	$C_\mu = 1.44 \times 10^{-3}$
16	Continuous	-	660	2.0	1.0	$C_\mu = 2.89 \times 10^{-3}$

To define the coordinates of the airfoil an .stl file with a 3D geometry of the airfoil was loaded into the blockMeshDict to define the airfoil coordinates in the mesh, this technique its easier than using the airfoil equation.

```
actions
( // Plasma
{
   name    plasmaCellSet;
   type    cellSet;
   action  new;
   source  rotatedBoxToCell;
   sourceInfo
   {
      origin (0.099 -0.15 0.0583);
      i (0.0310258 0 0.0056285);
      j (0 0.3 0);
      k (-0.002 0 0.002814);
   }
}
{
   name    plasma;
   type    cellZoneSet;
   action  new;
   source  setToCellZone;
   sourceInfo
   {
      set plasmaCellSet;
   }
}
);
```

By executing the command blockMesh the mesh is generated, and the running status is printed in the terminal, any mistakes are picked up by blockMesh and the resulting error message appears in the command window. After the mesh the command checkMesh is executed to perform a mesh quality check. The mesh independence analysis was carried out, to ensure that the numerical solution obtained is independent of the number of nodes used.

Steady state simulations with same initial and boundary conditions were performed for a range of angles of attack from $0°$ to $12°$. Through the comparison between the aerodynamic coefficients and the pressure distribution it was found that the solutions converge independently of the mesh, the mesh 3 was selected to carry out the study, the y^+ was kept to $y^+ \leq 5$.

3.2 OpenFOAM Configuration

The simulations were carried out in the open-source software OpenFOAM. The actuator is simulated in OpenFOAM as force source term, that is added into the momentum equation. There are two ways of adding the force one involves modifying a solver and the other trough fvOptions.

The body force can not be directly added by modifying a solver and adding the body terms in the momentum equation, since by doing this the body force will be applied to the whole domain as for example a gravitational force, as for this work is required that the force is applied only into a specific region, then we used the fvOptions application that allow us to apply an arbitrary source term into a set of specific cells (cell zone).

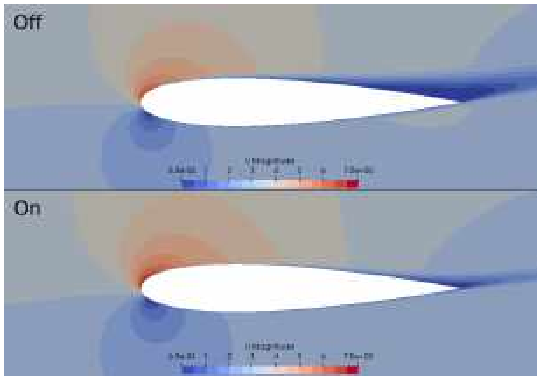


Fig. 9. Velocity field at $\alpha = 12^\circ$, actuator location x/c=0.03

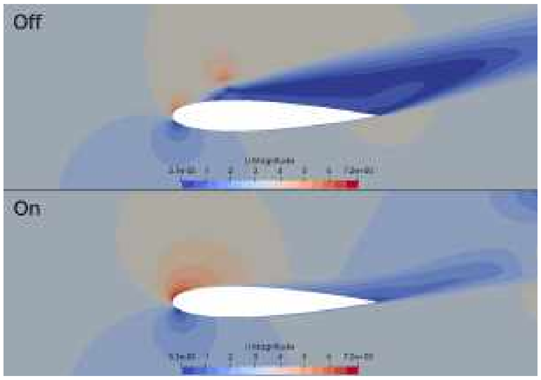


Fig. 10. Velocity field at $\alpha = 16^\circ$, actuator location x/c=0.03

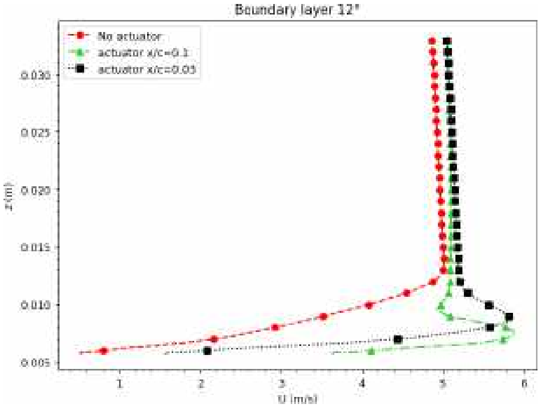


Fig. 11. Boundary layer measured at x/c=0.1 at $\alpha = 12^\circ$

To generate the cell zone the `topoSet` utility was employed, this works with a dictionary `system/topoSetDict`.

In `topoSetDict` file the name of the region, its type, and source type are established. In this work the source was a `rotatedBoxToCell` which is a skewed, rotated box. Given as origin and three spanning vectors.

We used a rotated box because since the length of the actuator is small which allows the cell zone to adjust to the wall slope change along the airfoil chord. execute the function the command `topoSet` is used.

The actuator is simulated in OpenFOAM as force source term, that is generated with the `CodedSource` a `fvOptions` utility in which trough `C++` language several types of source terms can be programmed. Once the arbitrary source term is programmed it will be acting on the selected cell zone.

A typical `CodedSource` file contains hook functions that allows the entry of sources through `codeAddSup`, restriction values before the equation is solved using `codeSetValue` and applying corrections after the equation was solved with `codeCorrect`. For this work inside the `codeAddSup` brackets the cell zone and time are called as follows:

```
cellSet selectedCells (mesh_, cellSetName_);
labelList cells = selectedCells.toc ();
const Time& time = mesh().time();
```

Then constants for the actuator frequency and strart time are defined:

```
const scalar pi(M_PI);//  pi math constant

const scalar f_base = 2e3;//Hz  base frequency
const scalar f_burst = 15;//Hz busrt frequency

const scalar startTime = 2.0;//s
```

After the force term is defined as a `DimensionedField` vector object with acceleration per volume dimensions, the magnitude is established and the term is added to the momentum equation:

```
UIndirectList <vector> (Su, cells) =
vector (0.006705, 0, 0) / V;
eqn += Su*pow(sin(2*pi*f_base*time.value()),2);
```

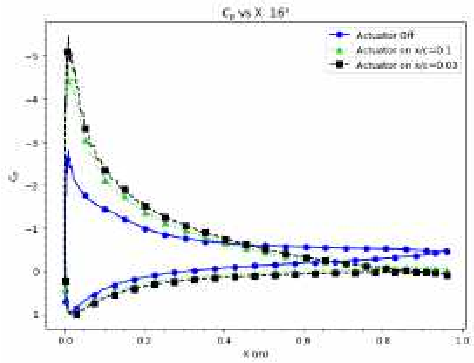


Fig. 12. C_p distribution at $\alpha = 16°$

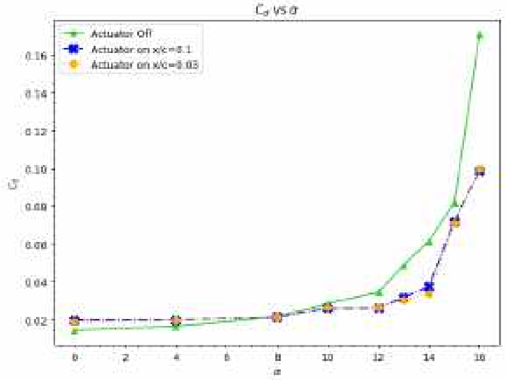


Fig. 13. Comparison of the C_d versus angle of attack for Re=2.0×10^5

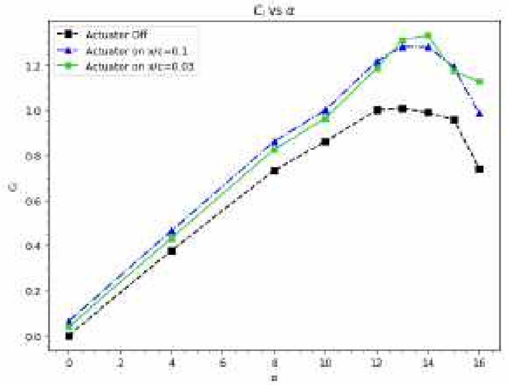


Fig. 14. Comparison of the C_l versus angle of attack for Re=2.0×10^5

For the boundary conditions, a condition of type `fixedValue` is applied to the entry to set the free current speed, a condition `inletOutlet` which provides a generic outlet when applying zero gradient for positive flows (outside the domain).

A boundary condition of type `noSlip` sets the speed with a value of zero. Also, in the airfoil wall functions are applied for turbulent kinetic energy `kLowReWallFunction`, turbulent viscosity `nutkWallFunction` and for the specific turbulent dissipation `omegaWallFunction`.

To ensure a two-dimensional flow, a boundary condition of type `empty` is applied to the boundaries perpendicular to the flow. The solver `pimpleFoam` was used, this solver uses the PIMPLE algorithm (merged PISO-SIMPLE), to execute the PIMPLE solver in the so called PISO mode the keyword `nOuterCorrectors` must be ≤ 1 as follows:

```
PIMPLE
{
   nOuterCorrectors 1;
   nCorrectors     2;
}
```

The keyword `nCorrectors` indicates that the pressure poisson equation is being resolved twice so the pressure-momentum coupling is now stable enough to produce a non-diverging solution. The discretization schemes were a second order accurate and fully bounded setup to give precision and stability:

— time: backward,

— gradient: cellMDLimited Gauss linear 0.5,

— divergence: Gauss linearUpwind,

— laplacian: Gauss linear limited 1.0,

— interpolation: linear.

The linear solver was GAMG (geometric-algebraic multi-grid) for the symmetric matrix p, and for the asymmetric matrices U,k,omega and nuTilda a `smoothSolver` with `GaussSeidel` smoother.

The time step for the simulations was controlled by means of maximum Coruant number of 0.5, it was used and adaptable time step function to reduce the computation time for each simulation.

The simulations were solved in parallel on distributed processors. OpenFoam uses the public domain openMPI implementation of the standard message passing interface (MPI).

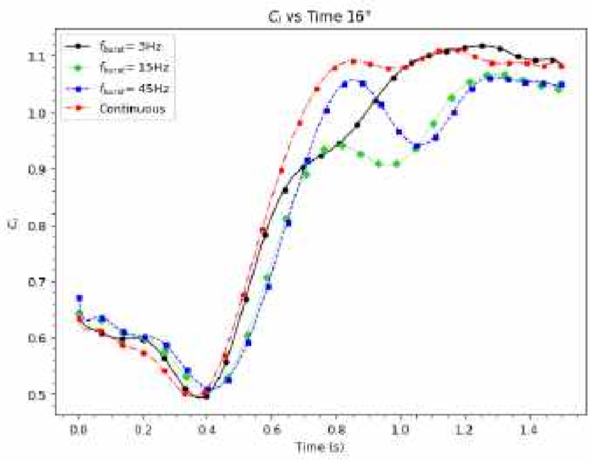


Fig. 15. C_l versus actuation time, at $\alpha = 16^\circ$

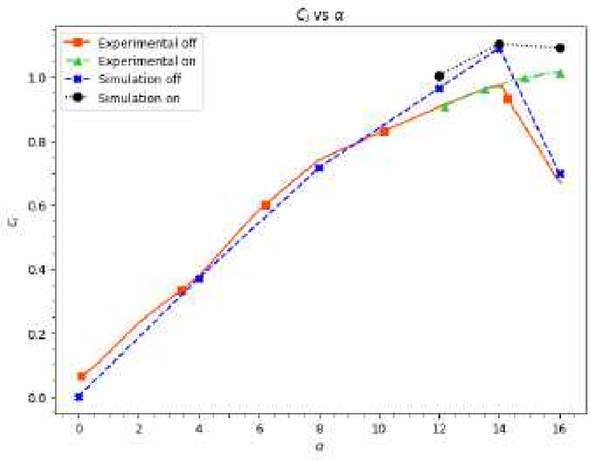


Fig. 16. Simulation and experimental C_l validation plot

The decomposition method used was `simple` that is a geometric decomposition in which the domain is split into pieces by direction, in this work the domain was decomposed in 16 sub-domains.

4 Results

The simulations of a NACA 0015 airfoil were performed at Re of $\times 10^5$ with the following initial conditions $U_\infty = 3.0$ m/s, $\rho_\infty = 1.0$ kg/m^3 and $\nu = 1.5 \times^- 5$ m^2/s. To evaluate the effect of the position of the actuator along the chord over the lift coefficient C_l and drag coefficient C_d of a NACA 0015 airfoil were executed for a range of angles of attack from 0° to 16°, the actuator was placed at 3% (x/c=0.03) and 10% (x/c=0.1), the cases for this study correspond to the Table 2.

The Fig. 9 shows the velocity field when the actuator is inactive (off), and active (on). It is observed that when the actuator is inactive there is a flow separation zone that goes from the trailing edge to distance near 0.6x/c.

When the actuator is active the flow remains attached, the separation zone is reduced, its only appreciated near the trailing edge, in this case the actuator was placed at 0.03x/c. The flow reattachment its more visible when the airfoil is stalled as seen in the Fig. 10, in this case the actuator is able to take out the airfoil from the stall.

As mentioned in the section 1 plasma actuators add momentum to the flow, this induces a jet within the boundary layer, the simulations proved to be able to replicate the boundary layer jet as seen in Fig. 11. The C_p distribution for the cases on and off at the stalling angle $\alpha = 16^\circ$ are shown in the Fig.12.

With the actuator off the pressure peak is almost vanished, and consequently the lift. When the actuator is on there is significant low pressure recovery near the trailing edge. Fig.14 shows the enhancement of the airfoil as the C_l when the actuator is on is clearly higher compared to stall case with the actuator off.

When the actuator is on there is a significant reduction of the drag at high angles of attack from a range from 12° to 16°, see Fig. 13, this reduction of the drag is due to the reattachment of the separate flow. To test the effect the burst mode, simulations were performed at 16°, the conditions of the actuator for each case are listed in the Table 3. In the Fig.15 the C_l versus the time when the actuator is on.

It is observed that independent of the burst ratio frequency the C_l have a behavior similar have a behavior similar to that of a step function, the main differences between them are the smoothness of the C_l increase evolution, and time to reach the maximum C_l. The highest C_l is for the continuous mode case (BR=1.0) and for the case with $f_{\text{burst}} = 3$ Hz (BR=0.5).

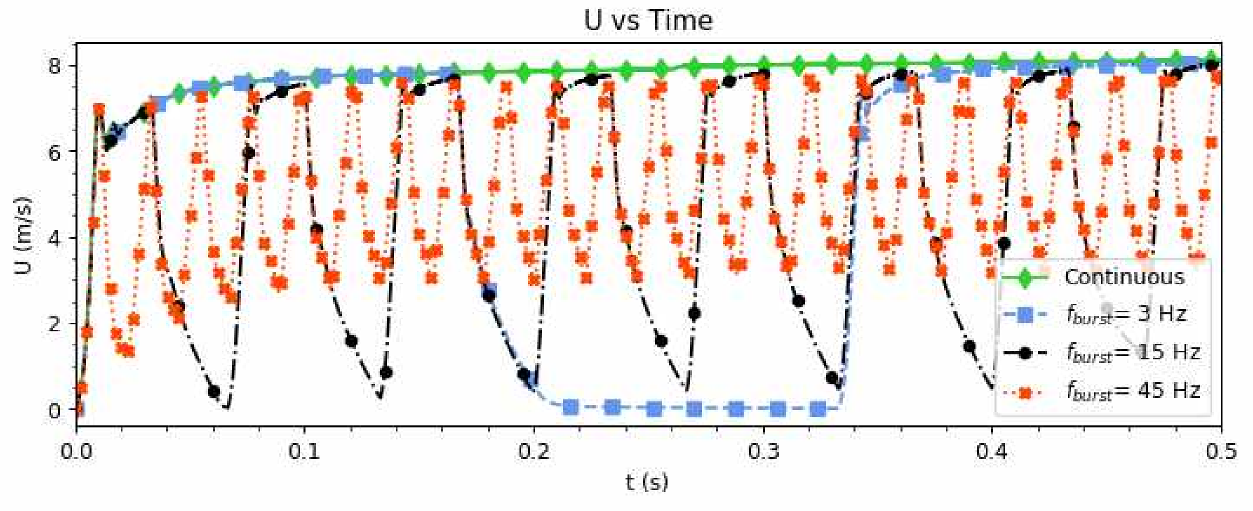


Fig. 17. Evolution of the velocity in a point at x= 3 mm downwind the actuator and z=1 mm over the wall, $f_{base} = 2$ kHz

Fig. 17 shows the temporal evolution of the velocity in a half period (T=0.5), measured in a point 3 mm downwind the actuator and 1 mm over the wall, from this plot the effect of the burst mode as the actuator is being activated and deactivated, for example for a $f_{\mathrm{burst}} = 15$ Hz it is possible to count 7.5 on and off states, since the graph only shows a half period.

4.1 Validation

To validate the technique implemented in this work, were performed at a Re of 1.58×10^5 to match the Re used by Post in his experiments, and compare the C_l and C_p from the simulation with the data reported by Post [12]. In Fig.16 the C_l polar shows the experimental data from Post and the simulations.

It is observed that the simulations C_l has similar behaviour than the one from the experiments, when closer to the α_{maxCl} the C_l tends to be overestimated. This may be caused by the use of RANS turbulence model, it may be possible to get a closer approximation with a more precise turbulence model as Large Eddy Simulation (LES) models and with an smaller time step.

5 Conclusion and Future Work

The results of the CFD analysis technique applied to an airfoil section showed us a great potential and high capabilities to simulate and evaluate the Kloker plasma-fluid model for a Single Dielectric-Barrier Discharge and the $k - \omega SST$ turbulence flow model during the coupled.

This technique is cheap, fast, effective and free of the software licenses to simulate the performance of any plasma solver to attach the boundary layer after the stall or bubble separation in a wing section.

The numerical algorithm called pimpleFoam with nCorrector is one of the most stable setups for this case and the accuracy of the results strongly depends on the choice of grid size, y-plus, wall function and discretization scheme.

The tested cases showed that when the actuator its closer to the separation region produces a higher increase of the C_l and a larger reduction of the separation zone, as a matter of fact if the separation occurs at 0.5x/c the actuator placed at 0.1x/c generates a C_l increase slightly higher than the one from the actuator at 0.03x/c, keeping in mind than the separation depends on the angle of attack.

Plasma actuator has great advantages, such as responsivity, low weight, a simple structure, and a low energy consumption.

Reason enough, for this paper who provides detailed data support for subsequent numerical and experimental studies on airfoil cases with a 3D flow and more complex turbulence models.

References

1. **Aono, H., Sekimoto, S., Sato, M., Yakeno, A., Nonomura, T., Fujii, K. (2015).** Computational and experimental analysis of flow structures induced by a plasma actuator with burst modulations in quiescent air. Mechanical Engineering Journal, Vol. 2, No. 4, pp. 15–233. DOI: 10.1299/mej.15-00233.

2. **Bernal Orozco, R. A., Arias-Montano, A., Huerta, O. (2020).** Separated-flow control simulation with a periodic excitation by sdbd plasma actuator at re=o(20_5). Journal of Aerospace Engineering, Vol. 33. DOI: 10.1061/(ASCE)AS.1943-5525.0001185.

3. **Corke, T., Post, M. (2005).** Overview of plasma flow control: concepts, optimization, and applications. 43rd AIAA Aerospace Sciences Meeting and Exhibit, pp. 563. DOI: 10.2514/6.2005-563.

4. **Dörr, P., Kloker, M. (2015).** Numerical investigation of plasma-actuator force-term estimations from flow experiments. Journal of Physics D: Applied Physics, Vol. 48, No. 39.

5. **Font, G., Enloe, C., McLaughlin, T. (2010).** Plasma volumetric effects on the force production of a plasma actuator. AIAA Journal, Vol. 48, No. 9, pp. 1869–1874. DOI: 10.2514/1.J050003.

6. **Fujii, K. (2014).** High-performance computing-based exploration of flow control with micro devices. Philosophical Transactions of the Royal Society A, Vol. 372, No. 2022. DOI: 10.1098/rsta.2013.0326.

7. **Greenblatt, D., Müller-Vahl, H., Lautman, R., Ben-Harav, A., Eshel, B. (2015).** Dielectric barrier discharge plasma flow control on a vertical axis wind turbine. Active Flow and Combustion Control. Springer, Vol. 127, pp. 71–86. DOI: 10.1007/978-3-319-11967-0_5.

8. **Hofkens, A. (2016).** Determination of the body force generated by a plasma actuator through numerical optimization. Master's thesis.

9. **Huang, J., Corke, T. C., Thomas, F. O. (2006).** Unsteady plasma actuators for separation control of low-pressure turbine blades. AIAA Journal, Vol. 44, No. 7, pp. 1477–1487.

10. **Joslin, R. D., Miller, D. N., Lu, F. K. (2000).** Fundamentals and applications of modern flow control. American Institute of Aeronautics and Astronautics.

11. **Kumar, V., Alvi, F. S. (2006).** Use of high-speed microjets for active separation control in diffusers. AIAA Journal, Vol. 44, No. 2, pp. 273–281. DOI: 10.2514/1.8552.

12. **Post, M., Corke, T. (2004).** Separation control using plasma actuators-stationary & oscillating airfoils. 42nd AIAA Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics, pp. 841. DOI: 10.2514/6.2004-841.

13. **Sato, M., Nonomura, T., Okada, K., Asada, K., Aono, H., Yakeno, A., Abe, Y., Fujii, K. (2015).** Mechanisms for laminar separated-flow control using dielectric-barrier-discharge plasma actuator at low reynolds number. Physics of Fluids, Vol. 27, No. 11.

14. **Shyy, W., Jayaraman, B., Andersson, A. (2002).** Modeling of glow discharge-induced fluid dynamics. Journal of Applied Physics, Vol. 92, No. 11, pp. 6434–6443. DOI: 10.1063/1.1515103.

15. **Suzen, Y., Huang, G., Jacob, J., Ashpis, D. (2005).** Numerical simulations of plasma based flow control applications. AIAA Paper, Vol. 4633. DOI: 10.2514/6.2005-4633.

16. **West, T., Hosder, S. (2012).** Numerical investigation of plasma actuator configurations for flow separation control at multiple angles of attack. 6th AIAA Flow Control Conference, pp. 3053. DOI: 10.2514/6.2012-3053.

Article received on 08/06/2020; accepted on 20/12/2021. Corresponding author is Oliver Marcel Huerta Chávez.

ISSN 2007-9737

WSC2RCNN: A Deep Learning Actions-based Classifier for Improved Web Service Discovery

Meghazi Hadj Madani, Aklouf Youcef

University of Science and Technology Houari Boumediene,
Research Laboratory in Informatics, Intelligence,
Mathematics and Applications, Algiers,
Algeria

{hmeghazi, yaklouf}@usthb.dz

Abstract. Due to the increasing popularity of Web services and their tremendous number, discovery tasks are shown to be the most important and difficult steps. The difficulty lies in the limited data provided on them, which is, in most cases, a short textual description. Based on these descriptions, many interesting works have been proposed trying to classify them in an efficient way. In this work, we propose a new approach based on deep learning WSC2RCNN that uses Web services descriptions and actions within these descriptions to improve the classification task, which has given promising results and outperforms state-of-the-art approaches using the same data.

Keywords. Web service, service classification, service actions, deep neural network.

1 Introduction

As a result of the strong demand for Web applications and the evolution of Web technologies, especially Service Oriented Architecture, providing more suitable solutions for interoperability, this led many software vendors to publish their applications as Web Services (WS). Like any good craftsman, if we have the right tools, we will necessarily get a quality end product; hence, the importance of the Web services discovery process.

Given their large number, the process of Web services discovery/selection has become problematic and inaccurate. Several research works have been done to solve this and have shown that Web services classification / clustering is the solution to reduce complexity, not only for the discovery process but also for recommendation, selection and composition. [21, 9, 6, 23, 4]. Early methods achieved clustering using textual WSDL descriptions as they are or by applying data mining techniques on them[18]; in the same context [9] approached the problem by selecting a set of Web service features such as WSDL contents, messages, types, and ports.

The problem here is that these kinds of methods are purely syntactic and devoid of semantics [2]. To remedy this, some formalisms have been proposed to add a semantic dimension, such as WSDL-S [1], OWL-S [15], and WSMO [12].

These formalisms have everything they need to shine but on a small scale, since they require a lot of effort for the development of high-level ontologies, the annotation of web services and associated inference operations.

Authors in [14] wanted to approach the problem from a social point of view and tried to exploit Web services interactions by building social networks defined on top of three relations categories which are: competition and substitution for web services having the similar functionalities, in addition to a collaboration category for those having different functionalities.

The social dimension was also exploited in [8] where a *global-social-service* network was created by associating to services related ones based on their functionalities, then capitalizing on service-to-service exploration.

doi: 10.13053/CyS-26-4-4069

We find also all the works that use the description documents of Web services to bring out the semantic features and classify them. Some used probabilistic topic model like LDA (Latent-Dirichlet-Allocation) to extract latent-topic features [7, 3, 20, 13, 17].

In the last few years, research in the field has changed course, especially after moving to the popular API style as a common mode of Web service consumption, and an additional category has emerged as a result of the great success of Deep Learning.

By exploiting deep learning techniques' potential and efficiency in natural language processing, new possibilities have emerged for Web services classification since their descriptions are available as short texts. Therefore, several approaches from the field [11, 25, 24, 27, 30, 16, 26, 22, 31] have dethroned traditional ones.

In this paper, we propose a new approach that belongs to Deep learning methods using service description. Named WSC2RCNN, our Web Service Classifier combines two Recurrent Convolutional Neural Networks for text classification.

The first one is applied to extract semantic features by capturing contextual information from Web services' embedded descriptions; the second network does the same thing but on words related to service actions generated from their original descriptions, this to highlight actions that reflect what a Web service really does.

After having done considerable experiments on a Dataset crawled from a well-known repository “ProgrammableWeb” with a good number of real-world web services, we have clearly shown that our proposed method can achieve a more precise classification and surpass the state-of-the-art methods. The rest of this paper is organized as follows.

Section 2 gives an overview of related work including Deep Learning approaches for Web services classification. Section 3 is dedicated to present our approach. Section 4 describes evaluation metrics used to measure different methods' classification performances. In section 5, we exhibit our experimental setup and results. Section 6 concludes the paper.

2 Related Work

The discovery of web services has aroused the interest of the research community and has seen significant activity. Considered as a flagship solution, research on Web services classification using Deep Learning techniques can be done using two main data sources [26]: Web services description documents and details collected from their ecosystem.

For instance, in [29] Web service clustering is accomplished by obtaining the effective words using information gain theory then combining them to an attention-based Bi-LSTM neural network. To seize the most relevant semantic information automatically in a Web service document, Cao et al. [5] adopted the Bi-LSTM architecture.

Then their model incorporated the Bi-LSTM architecture through a topical attention-mechanism to perform Web service classification prediction. In [30], a deep neural network model (DeepWSC) is trained based on probabilistic topic model and and taking advantage of the RCNN [11] capabilities (which has been customized), to get Web service description implicit contextual features.

The above methods share with ours the source of training data which is the description documents of Web services. However, in order to be able to provide a quality semantic information, they use, for the most, a probabilistic-topic-model to supervise this process. Although [30] benefits from RCNN, our work, however, differs from the latter in the trained-word-embedding model and the way of processing training data.

The rest of methods often combine the information collected from the Web services ecosystem with their descriptions. As a supervised method, Servnet [26] proposes a deep neural network to predict the service categories from service specifications, then automatically extract features from the service name and description to combine them under a unified feature in order to predict service classification.

In [22] authors realize automatic extraction of function-description-documents by providing a new deep neural network which integrates Graph-Convolutional-Network (GCN), to extract the global spatial features, with Bidirectional

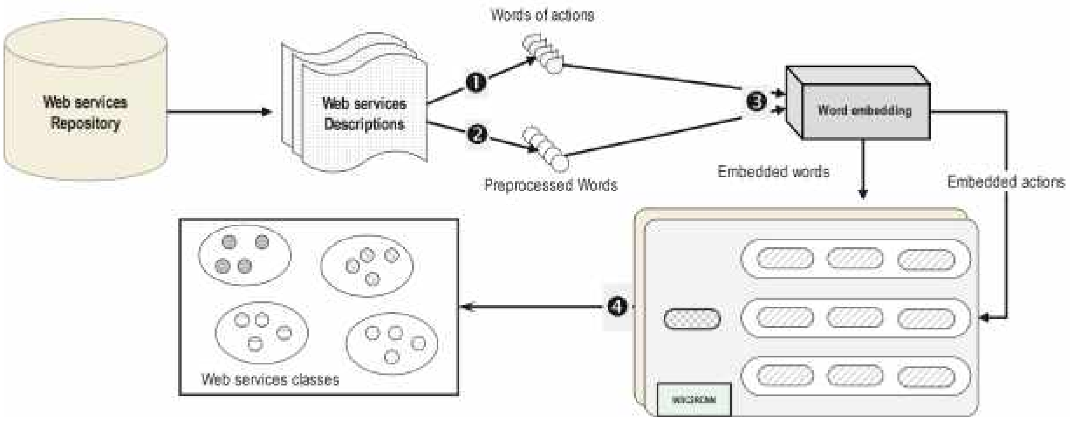


Fig. 1. WSC2RCNN framework for web services classification

Long Short Term Memory (Bi-LSTM) network, to learn sequential features. In an unsupervised manner, to achieve clustering, the second version of DeepWSC [31] gets Web services implicit features by combining deep neural network, to generate semantic features, with service composability relationship, extracted from an invocation convolutional network.

Authors in [28] also tried to design a supervised similarity knowledge graphs KSN to improve service description semantic. Associated to a CNN network, it is employed to extract context information. It is clear that the current trend is to develop graphs that attempt to capture different relationships and exploit them in order to improve the discovery process by finding specific types of patterns.

Even if DeepWSC [31] did good results by grafting a ”Service Composability Network” on the old version, our method was able to show better results. This is due, in the first place, to the way of treating the words of the descriptions, then, it has a great relation with the importance that we have given to the words which describe the actions of a web service.

As shown in section 5.3, our method gives very good results which exceed those mentioned, by taking into consideration the very nature of the words which appear in the descriptions of the web services.

3 Our Approach

Web Services are meant to execute operations and accomplish tasks, that's what they are and that's what defines them. However, in the most cases, and as representation data on them, we can only have short textual description. So to efficiently discover them through it, the question is:

“*which words (or parts) within these descriptions represent really what a service does?*”.

Our first intuition was that if we are in front of a text, actions contained in this text are represented naturally by ’verbs’. So, we started by extracting all possible verbs all over web services descriptions, by using pre-trained natural language models. Here, we faced two problems: The first one, is that these models can not detect all verbs correctly, for example the verb ’searches’ in the sentence:

“*Service X searches for all worldwide airlines that operate in a given country*”.

Is tagged as a ‘Noun’; What if we are going to take each word apart (and out of its context)?. The second problem is that since the majority of descriptions are short, it is hard to have significant number of verbs to work with, using these models.

What makes this process even harder is that after doing text cleaning and pre-processing step (especially for the Stemming part), many words loose their initial tag, so a lot of verbs are detected as nouns, adjectives ... etc.

```
Algorithm 1: Extract Actions
  Input  : Web service textual description
  Output: List of actions
  Start parsing the sentence tree from the beginning;
  if item is a verb and has xcomp then
      got to the xcomp node;
  else
      if item is a verb and has dobj then
          add action to the list;
      else
          if item is a verb and has conjunction then
              go to the conjunction item; once you reach the noun, add action to the list;
          end
      end
  end
```

For example, the following sentence: *"Service x is used to validate monetary transactions."* the words '*Service*', '*used*' and '*validate*' are tagged as follow:

Original Tags	Stem Tags
('Service', 'proper noun')	('servic', ' adjective')
('used', ' **verb**')	('use', 'adjective')
('validate', '**verb**')	('valid', 'adjective')

This leads, in most cases, to changes in the semantic meaning of each word and consequently in the overall meaning of the description. Therefore, in our case, we tried to overcome this since the first step of our approach. Our method is based on four main steps (Figure 1):

We start in **Step 1** by a soft pre-processing of the services descriptions to preserve as possible words' original tags. Even this was not enough, because we get a very small number of verbs devoid of their context.

To overcome this, we used in **Step 2** the 'Extract-actions' algorithm, to extract what we qualify as actions and not only verbs.

After applying this process on every Web service description in the Dataset, in **Step 3**, we inject the pre-trained GloVe [19] model representation of both descriptions and extracted actions as inputs of the respectively two Recurrent Convolutional Neural Networks (RCNN) to be trained:

— **xcomp**: *An open clausal complement*,

— **dobj**: *The direct object*.

For both networks we used the original RCNN [11] which aims to capture text semantics by taking into account sequence words order. It makes the most of both RNN to learn local context of tokens and CNN for long-term dependencies. For the first RCNN network, we use $c_l(w_i)$ and $c_r(w_i)$ as, respectively, the left and right context of a word w_i, calculated using the following equations:

$$c_l(w_i) = f(W^{(l)}c_l(w_{i-1}) + W^{(sl)}e(w_{i-1})), \quad (1)$$

$$c_r(w_i) = f(W^{(r)}c_r(w_{i+1}) + W^{(sr)}e(w_{i+1})), \quad (2)$$

where f is a non-linear activation function. $e(w)$ is the word embedding vector of a word. $W^{(l)}$ is a matrix used to transform the context into the next hidden layer.

$W^{(sl)}$ is a matrix used to associate the semantic of the current word with the next word's left context. To get the latent semantic representation of the y_i vector, the equation (3) is used:

$$y_i = \tanh(Wx_i + b), \quad (3)$$

where x_i is the representation of a word w_i and

$$x_i = [c_l(w_i); e(w_i); c_r(w_i)]. \quad (4)$$

For the second network, y_a (of actions) is calculated in a similar manner but applied to the words of the extracted actions. As targets of WSC2RCNN, both RCNN networks outputs are associated to the 20 high-quality Web services primary categories from the Dataset.

The results obtained of the two RCNN networks are then added to each other, in **Step 4**, to prepare the final output. Since we have a multi-class problem, a softmax activation layer is applied to normalize the final output named *M_output*. Figure 2 illustrates our model.

Table 1. The classification performances

Models	**Purity**	**NMI**	**Recall**	**F1-Score**
LDA+K	0.5200	0.4262	0.3199	0.3383
LDA	0.5285	0.4341	0.3321	0.3503
WE-LDA+K	0.5372	0.4363	0.3282	0.3466
WE-LDA	0.5420	0.4403	0.3370	0.3543
Text-CNN+WE-LDA+K	0.5553	0.4668	0.3572	0.3733
RCNN+WE-LDA+K	0.5708	0.4856	0.3821	0.3969
RCNN+WE-LDA+Heuristics	0.6379	0.5273	0.4186	0.4356
WSC2RCNN	*0.6288*	**0.5648**	**0.4754**	**0.4852**
WSC2RCNN+SC	**0.6470**	**0.5755**	**0.5045**	**0.5129**

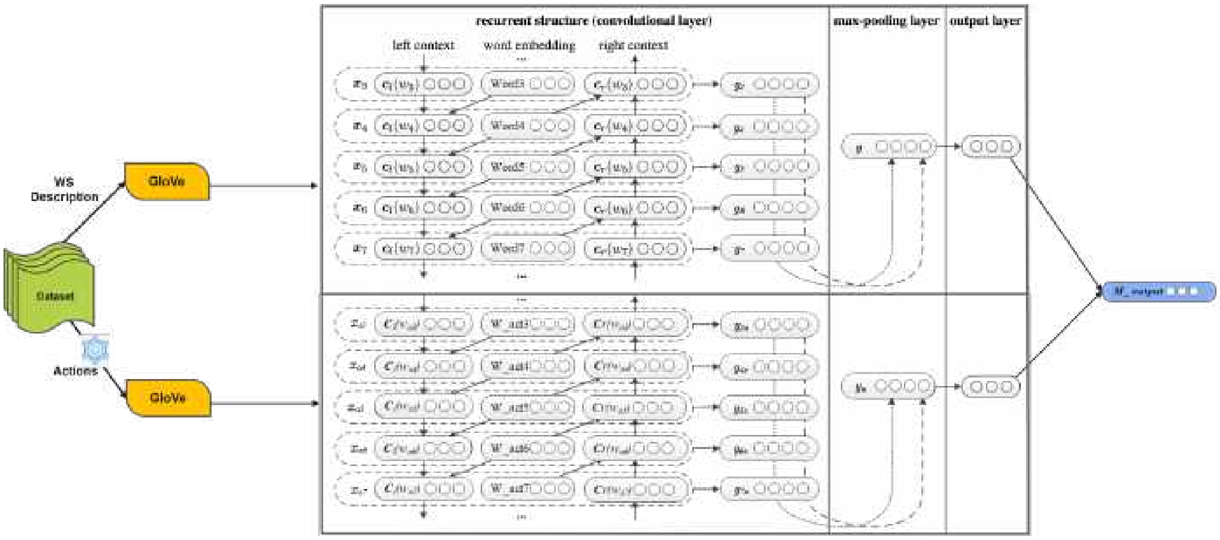


Fig. 2. Our model: WSC2RCNN

4 Used Evaluation Metrics

WSC2RCNN has been evaluated on four most used evaluation measures which are: Purity, Normalized-Mutual-Information (NMI), Recall and F_1-measure: **Purity** is one of the supervised cluster validation measures, calculated using the following formula:

$$\mathrm{Purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |\omega_i \cap c_j| , \quad (5)$$

where $\Omega = \{w_1, w_2, ..., w_K\}$ represents the set of web services clusters and $\mathbb{C} = \{c_1, c_2, ..., c_J\}$ is the set of Web services classes. **NMI** is a measure based on the mutual information and defined as:

$$\mathrm{NMI}(\Omega, \mathbb{C}) = \frac{2 \times I(\Omega; \mathbb{C})}{H(\Omega) + H(\mathbb{C})}, \quad (6)$$

where we can get the mutual information *I* using:

$$I(\Omega; \mathbb{C}) = \sum_{i=1}^{k} \sum_{j=1}^{k} P(\omega_i \cap c_j) \log \frac{P(\omega_i \cap c_j)}{P(\omega_i) \cap P(c_j)}. \quad (7)$$

Table 2. Distribution of Web services over the different categories

Id	Primary Category	Nbr of services
0	Tools	887
1	Financial	757
2	Messaging	591
3	eCommerce	553
4	Payments	553
5	Social	510
6	Enterprise	509
7	Mapping	429
8	Government	371
9	Science	357
10	Telephony	342
11	Security	312
12	Reference	304
13	Email	299
14	Search	290
15	Travel	294
16	Video	281
17	Education	277
18	Advertising	274
19	Transportation	269

The entropy H is defined as:

$$H(\Omega) = -\sum_{i=1}^{k} P(\omega_i) \log P(\omega_i). \quad (8)$$

Recall is used to find out how much proportion of the true class is correctly predicted. It is calculated as follow:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad (9)$$

where *TP* is the number of services assigned to their correct class and *FN* is the number of those where the model incorrectly predicts their positive class as negative. The last used measure is $\mathbf{F}_1$ which is calculated using the following formulas:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad (10)$$

where *FP* refers to the number of services where the model incorrectly predicts their negative classes as positive:

$$F_1\text{-measure} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \quad (11)$$

5 Experiments and Results

5.1 Experimental Environment

We conducted our experiments to evaluate and show the effectiveness of WSC2RCNN, and all of them have been done using python and carried out on a platform with an Intel(R) Xeon(R) platinum 8259CL CPU@2.50GHz (32 cores) and 256 GB RAM.

Since ProgrammableWeb is the largest online Web Services registry with over 23K API, from it we used the Dataset[1] of [30, 31] which contains 17,923 crawled real-world web services.

The experimental data contains 8,459 best quality web services of the top 20 categories, see Table 2. We also used Glove6B with 100-dimensions word-vectors for our model's networks as pre-trained word-vectors.

5.2 Evaluation Methods

By applying the following evaluation metrics: Purity, Normalized-Mutual-Information (NMI), Recall and F1-measure; We have tested our model on the same Dataset and compared our results with the best existing state-of-art methods listed by the authors of [31] as follow:

LDA [7], each web service has its own vector of probabilistic-topic-distribution generated, Web services clustered together are those with the same latent topic. In **LDA+K** [3], the K-means++ algorithm uses the similarity between Web services' vectors from the previous method to cluster them.

For **WE-LDA** [20], Web services are clustered together when they have the same latent topic, after that, they will be attributed to their highest value latent topics, and for **WE-LDA+K**, the K-means++ algorithm uses the similarity between Web services' vectors from WE-LDA to cluster them.

The **Text-CNN+WE-LDA+K** [31] method is based on a Text-CNN [10] service feature-extractor trained with WE-LDA. The last two methods use both a trained RCNN network [11], the first one **(RCNN+WE-LDA+K)** [30] is

[1] https://github.com/aourhtnowvherlcaer/programmableWeb

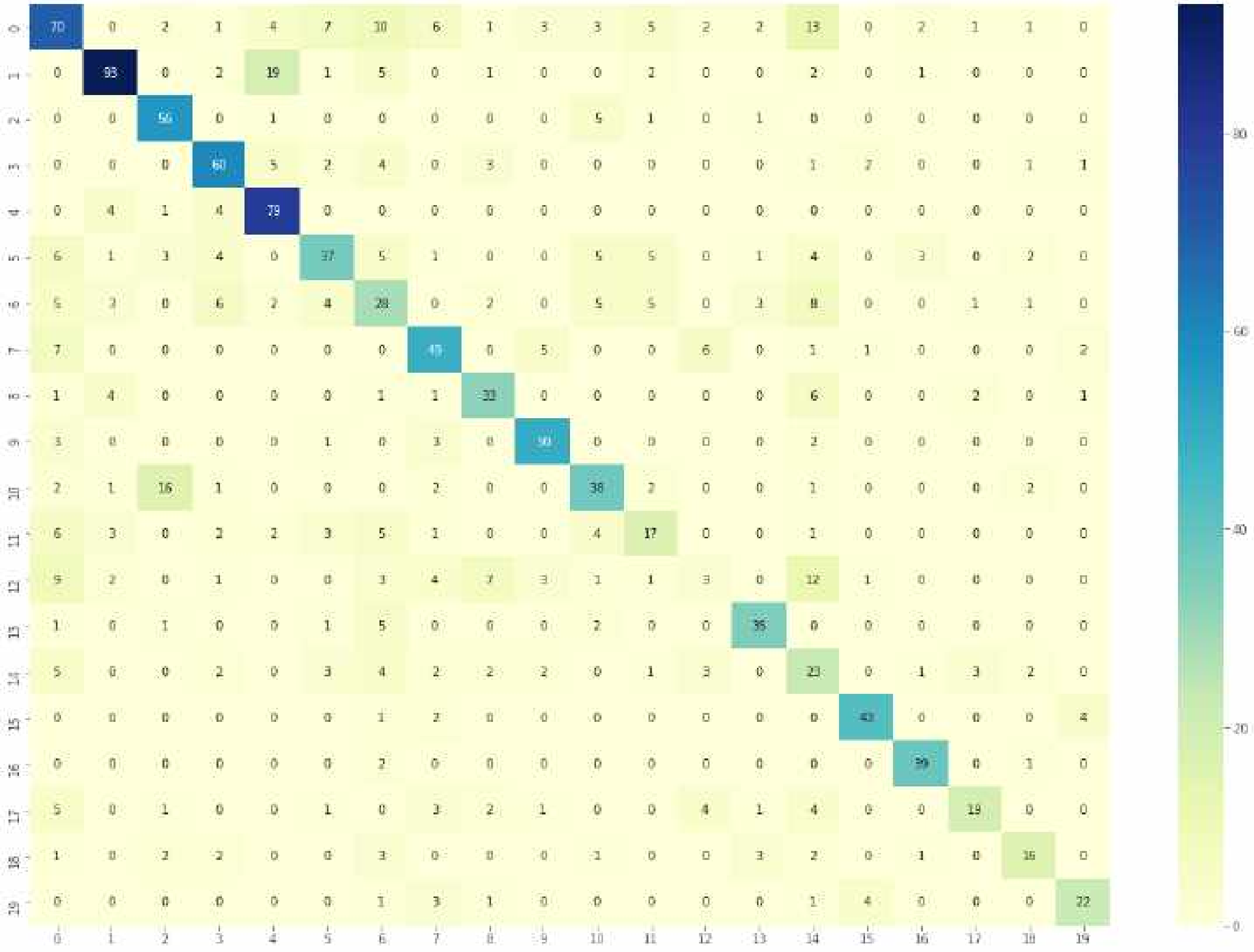


Fig. 3. M2RCNN Confusion Matrix

based on an LDA model. The second one **(RCNN+WE-LDA+Heuristics)** [31] uses service-deep-semantic-feature (RCNN) and service-composability-features (of a service composability built network) guided by a WE-LDA model.

5.3 Experimental Results and Discussions

By seeing Table 1, it is very clear that the first version of our approach exceeds all the candidate methods on the NMI, Recall and F$_1$ measures, and only the DeepWSC latest method with heuristics [31] exceeds it slightly on the Purity but it still gives very good results.

However, our second version, where we injected the embedded representation of secondary categories names (using GloVe), succeeded to outperform all candidate methods on all used measures.

Compared to LDA best performance, we notice an average improvement of 32.68% on all measures. If we take WE-LDA, the average of improvement is about 30.58%. On the best performance, which is DeepWSC with heuristics, we observed an average advantage of 7.66%.

The incorporation of the secondary categories names vectors to the combined [*actions*, *verbs*] feature on the second RCNN network did the work and improved obtained results.

Our new WSC2RCNN model outperforms the previous version with an average of 4.15% on all the evaluation measures. In addition to the good results we were able to have with our method, and if we look closely the confusion matrix, Figure 3, which is related to one of our models, we find that **WSC2RCNN** performs better than it looks.

For example, if we take line number 1, the model reacted well by making the prediction of the majority of web services in their correct class which is '***Financial***', whereas 19 Web services were predicted to belong to the class '***Payments***'. However, if we analyze the two classes from a semantic point of view, the two classes are quite close, which leads us to say that our model is not completely wrong on these predictions, since there is a relation between the two classes.

The same thing for the 16 services (line No.10) which have to be classified under the '***Telephony***' class but the model has assigned them to the '***Messaging***' class. From these results, we can see that our model gives positive results and takes into consideration the semantic aspect of Web services descriptions.

This leads us to believe that if we take into account the similarities between the classes, our method will give much better results than those already obtained. Without forgetting, the majority of works, dealing with the problem, perceive it as a classification problem where the classes are completely disjoint. This does not reflect the reality that a service can belong to several classes at the same time.

6 Conclusion and Future Works

In this paper, we propose a Web Service Classifier, named WSC2RCNN, based on Recurrent Convolutional Neural Networks for Text Classification (RCNN) and tested on a real-world Dataset with 8,459 web services.

The particularity of our approach is that it highlights the actions of web services extracted from their textual descriptions and accentuates the learning on these actions by training a dedicated and a customized RCNN Neural Network. Comparative experiments have shown that our model has been able to exceed all state-of-the-art approaches for web service classification on all the performance metrics. Since we have also shown that our model can give better results, in future work, we plan to extend our approach with more advanced techniques to take into account the semantic relations that exist between the classes of web services, in addition to an enhanced BERT embedding, adapted to our model to further results improvements.

In this work, the sequence of actions is analyzed and compared only within service descriptions. It will be more interesting to discover and capture sequences of inter-services actions in order to enrich the process of discovering web services with composite ones made up of other services and which offer the same expected functionalities.

References

1. **Akkiraju, R., Farrell, J., Miller, J. A., Nagarajan, M., Sheth, A. P., Verma, K. (2005).** Web service semantics - WSDL-S.
2. **Al-Masri, E., Mahmoud, Q. H. (2007).** Wsce: A crawler engine for large-scale discovery of web services. IEEE International Conference on Web Services (ICWS 2007), IEEE, pp. 1104–1111. DOI: 10.1109/ICWS.2007.197.
3. **Cao, B., Liu, X., Li, B., Liu, J., Tang, M., Zhang, T., Shi, M. (2016).** Mashup service clustering based on an integration of service content and network via exploiting a two-level topic model. IEEE International Conference on Web Services (ICWS), IEEE, pp. 212–219. DOI: 10.1109/ICWS.2016.35.
4. **Cao, B., Liu, X. F., Rahman, M. D. M., Li, B., Liu, J., Tang, M. (2020).** Integrated content and network-based service clustering and web apis recommendation for mashup development. IEEE Transactions on Services Computing, Vol. 13, No. 1, pp. 99–113. DOI: 10.1109/TSC.2017.2686390.
5. **Cao, Y., Liu, J., Cao, B., Shi, M., Wen, Y., Peng, Z. (2019).** Web services classification with topical attention based Bi-LSTM. International Conference on

Collaborative Computing: Networking, Applications and Worksharing, Springer International Publishing, pp. 394–407. DOI: 10.1007/978-3-030-30146-0_27.

6. **Cassar, G., Barnaghi, P., Moessner, K. (2014).** Probabilistic matchmaking methods for automated service discovery. IEEE Transactions on Services Computing, Vol. 7, No. 4, pp. 654–666. DOI: 10.1109/TSC.2013.28.

7. **Chen, L., Wang, Y., Yu, Q., Zheng, Z., Wu, J. (2013).** WT-LDA: user tagging augmented LDA for web service clustering. International conference on service oriented computing, Springer, Springer Berlin Heidelberg, pp. 162–176.

8. **Chen, W., Paik, I., Hung, P. C. K. (2015).** Constructing a global social service network for better quality of web service discovery. IEEE Transactions on Services Computing, Vol. 8, No. 2, pp. 284–298. DOI: 10.1109/TSC.2013.20.

9. **Elgazzar, K., Hassan, A. E., Martin, P. (2010).** Clustering WSDL documents to bootstrap the discovery of web services. 2010 IEEE international conference on web services, IEEE, pp. 147–154. DOI: 10.1109/ICWS.2010.31.

10. **Kim, Y. (2014).** Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Association for Computational Linguistics, pp. 1746–1751. DOI: 10.3115/v1/d14-1181.

11. **Lai, S., Xu, L., Liu, K., Zhao, J. (2015).** Recurrent convolutional neural networks for text classification. Proceedings of the AAAI Conference on Artificial Intelligence, AAAI Press, Vol. 29, No. 1, pp. 2267–2273.

12. **Lara, R., Roman, D., Polleres, A., Fensel, D. (2004).** A conceptual comparison of WSMO and OWL-S. European Conference on Web Services, Springer, pp. 254–269. DOI: 10.1007/978-3-540-30209-4_19.

13. **Liu, X., Agarwal, S., Ding, C., Yu, Q. (2016).** An lda-svm active learning framework for web service classification. 2016 IEEE International Conference on Web Services (ICWS), IEEE, pp. 49–56. DOI: 10.1109/ICWS.2016.16.

14. **Maamar, Z., Faci, N., Wives, L., Badr, Y., Santos, P., De Oliveira, J. P. M. (2011).** Using social networks for web services discovery. IEEE Internet Computing, Vol. 15, No. 4, pp. 48–54. DOI: 10.1109/MIC.2011.27.

15. **Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M. (2005).** Bringing semantics to web services: The OWL-S approach. International Workshop on Semantic Web Services and Web Process Composition, Springer, pp. 26–42.

16. **Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J. (2020).** Deep learning based text classification: A comprehensive review. arXiv preprint arXiv:2004.03705. DOI: 10.48550/ARXIV.2004.03705.

17. **Nabli, H., Djemaa, R. B., Amor, I. A. B. (2018).** Efficient cloud service discovery approach based on lda topic modeling. Journal of Systems and Software, Vol. 146, pp. 233–248. DOI: 10.1016/j.jss.2018.09.069.

18. **Nayak, R. (2008).** Data mining in web services discovery and monitoring. International Journal of Web Services Research (IJWSR), Vol. 5, No. 1, pp. 63–81. DOI: 10.4018/978-1-61520-684-1.ch012.

19. **Pennington, J., Socher, R., Manning, C. D. (2014).** Glove: Global vectors for word representation. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics, pp. 1532–1543.

20. **Shi, M., Liu, J., Zhou, D., Tang, M., Cao, B. (2017).** WE-LDA: a word embeddings augmented lda model for web services clustering. 2017 IEEE International Conference on Web Services (ICWS), IEEE, pp. 9–16. DOI: 10.1109/ICWS.2017.9.

21. **Skoutas, D., Sacharidis, D., Simitsis, A., Sellis, T. (2010).** Ranking and clustering web services using multicriteria dominance relationships. IEEE Transactions on Services Computing, Vol. 3, No. 3, pp. 163–177. DOI: 10.1109/TSC.2010.14.

22. **Wang, X., Liu, J., Liu, X., Cui, X., Wu, H. (2020).** A spatial and sequential combined method for web service classification. Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Springer, pp. 764–778. DOI: 10.1007/978-3-030-60259-8_56.

23. **Xia, B., Fan, Y., Tan, W., Huang, K., Zhang, J., Wu, C. (2015).** Category-aware api clustering and distributed recommendation for automatic mashup creation. IEEE Transactions on Services Computing, Vol. 8, No. 5, pp. 674–687.

24. **Xiong, R., Wang, J., Zhang, N., Ma, Y. (2018).** Deep hybrid collaborative filtering for web service recommendation. Expert systems with Applications, Vol. 110, pp. 191–205. DOI: 10.1016/j.eswa.2018.05.039.

25. **Xu, J., Wang, P., Tian, G., Xu, B., Zhao, J., Wang, F., Hao, H. (2015).** Short text clustering via convolutional neural networks. Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 62–69. DOI: 10.3115/v1/w15-1509.

26. **Yang, Y., Qamar, N., Liu, P., Grolinger, K., Wang, W., Li, Z., Liao, Z. (2018).** Servenet: A deep neural network for web services classification. 2020 IEEE International Conference on Web Services (ICWS), pp. 168–175. DOI: 10.48550/ARXIV.1806.05437.

27. **Yao, L., Mao, C., Luo, Y. (2018).** Graph convolutional networks for text classification. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 1, pp. 7370–7377. DOI: 10.48550/ARXIV.1809.05679.

28. **Yu, Y., Zeng, J., Yao, J., Wen, J., Xing, B. (2020).** Web service discovery based on knowledge graph and similarity network. 2020 IEEE World Congress on Services, IEEE, pp. 231–236. DOI: 10.1109/SERVICES48979.2020.00054.

29. **Zhang, X., Liu, J., Cao, B., Xiao, Q., Wen, Y. (2019).** Web service discovery based on information gain theory and bilstm with attention mechanism. International Conference on Collaborative Computing: Networking, Applications and Worksharing, Springer, pp. 643–658. DOI: 10.1007/978-3-030-12981-1_45.

30. **Zou, G., Qin, Z., He, Q., Wang, P., Zhang, B., Gan, Y. (2019).** DeepWSC: A novel framework with deep neural network for web service clustering. 2019 IEEE International Conference on Web Services (ICWS), IEEE, pp. 434–436. DOI: 10.1109/ICWS.2019.00077.

31. **Zou, G., Qin, Z., He, Q., Wang, P., Zhang, B., Gan, Y. (2022).** Deepwsc: Clustering web services via integrating service composability into deep semantic features. IEEE Transactions on Services Computing, Vol. 15, No. 4, pp. 1940–1953. DOI: 10.1109/TSC.2020.3026188.

Article received on 27/10/2021; accepted on 11/08/2022. Corresponding author is Meghazi Hadj Madani.

ISSN 2007-9737

On Causality Problem in Natural Language Processing Field

Altynay Yerkhassym[1], Alexandr A. Pak[1,2], Iskander Akhmetov[1,2],
Amir Yelenov[1,2], Alexander Gelbukh[3]

[1] Institute of Information and Computational Technologies,
Kazakhstan

[2] Kazakh-British Technical University,
Kazakhstan

[3] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

{yerkhassym.altynay, aa.pak83, iskander.akhmetov, greamdesu, gelbukh}@gmail.com

Abstract. Natural language processing (NLP) field has been developing rapidly recently. This article consists mainly of literature review of the basic understanding and solving the causality problem in natural language processing field. Existing models may benefit from the concept of causality because conventional language models are brittle and spurious [10]. Incorporating the principle of causality could assist in resolving this issue. Since this issue affects seriously on the accuracy value of NLP methods and algorithms, it is worth paying attention to. Content of the article includes the authors who have been covered this topic and have made researches respecting mentioned problem, the results that have been achieved, the methods and approached that have been used and the data that was used in researches.

Keywords. Natural language processing, neural network, causality.

1 Introduction

Natural language processing is a subfield of Artificial intelligence branch focused on allowing computers to perceive human language. NLP-based systems are primarily designed to comprehend and interact with human voice and text. Companies and organizations throughout the world are increasingly utilizing NLP-enabled solutions to obtain client information and enhance the automation of regular procedures.

These tools do numerous tasks, including translation, keyword extraction, subject classification, etc. To automate these procedures and provide precise results, however, machine learning is required. Machine learning is the application of algorithms that train machines to automatically learn from experience and improve without being explicitly programmed.

AI-powered chatbots, for instance, employ natural language processing to read what users say and what they mean to do, and machine learning to automatically provide more correct responses by learning from previous interactions.

However, this accuracy is never 100 percent, as determining causation remains a challenging task for machine learning algorithms and, consequently, natural language processing. More examples are used to train natural language processing models in an effort to tackle these issues.

As the environment becomes increasingly complicated, however, it becomes impossible to cover the full distribution by adding more training instances. Due to a lack of comprehension of cause-and-effect interactions, it is extremely challenging to generate accurate predictions and successfully adapt to novel situations.

doi: 10.13053/CyS-26-4-4434

The machine can forecast the outcome of every action, but this does not mean its predictions are always accurate.

Since there is always a possibility that certain events do not fit particular patterns. In this instance, the outcome is erroneous. In contrast to humans, who can construct a causal logic and forecast a more accurate output based on collected data, machines are incapable of doing so.

2 Authors who Have Addressed this Topic in their Works and Articles

In contrast to numerous challenges in natural language processing, the causal relationship has not been thoroughly examined. Recently released studies and works on this topic provide additional evidence. To comprehend how to implement this term into the work of algorithms and models, it is necessary to comprehend this element itself.

Determining the causal relationship in natural language, including analysis and psycholinguistics, is therefore the initial step in the investigation of this subject. Torgrim Solstad and Oliver Bott had made some researches on this topic and had written the article named Causality and causal reasoning in natural language [15].

This article offers a synopsis of theoretical and psycholinguistic approaches to causation in language. The primary phenomenological focus of the paper is on causal relations as articulated intra-clausally by verbs (such as break and open) and inter-clausally by discourse markers (e.g. because, therefore).

Special consideration is given to Implicit Causality verbs that elicit explanation expectations in the succeeding conversation. The article also analyzes linguistic terms, such as counterfactual conditionals, that do not convey causation as such but appear to require a causal model for their proper interpretation.

The study of the phenomena is supplemented with a summary of key characteristics of their cognitive processing as revealed by psycholinguistic research. Due to the strong relationship between machine learning and natural language processing, the problem of causality in machine learning is reflected in the NLP discipline.

Bernhard Scholkopf wrote an article [13] that can serve as an introduction to some relevant concepts of graphical or structural causal models for a machine learning. Algorithms and methods of Artificial Intelligence cannot reason and make decisions like humans or animals. They neglect numerous variables that can influence pattern formation and depend solely on generalized models based on uniformly distributed data. In addition, these models are poor in imagining and navigating imagined spaces.

The author thinks that causality, with its emphasis on modeling and reasoning about treatments, can make a significant contribution to understanding and resolving these challenges, thereby advancing the science.

Further continuation of the previous article can be found in the work of Bernhard Scholkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner Anirudh Goyal, Yoshua Bengio named Towards Causal Representation Learning [14].

In this article, the authors describe different levels of causal and statistical modeling, investigate the Independent Causal Mechanisms (ICM) principle as a key component that enables the estimation of causal relations in artificial intelligence agents, and examine existing methods for learning causal relations.

Primarily, authors present examples of causality and machine learning in scientific applications and hypothesize on the benefits of merging the skills of both domains to create a more adaptable AI. The extraction of causal patterns from natural language texts and using it in methods of natural language processing systems are described in the article written by P. Maslov [11].

This research proposes a technique for extracting and characterizing causal facts from Russian business prose documents. In addition, the implementation of the derived cause-and-effect relationships within the algorithm for anticipating severe scenarios is offered.

The majority of modern techniques are either lexico-semantic pattern matching or feature-driven

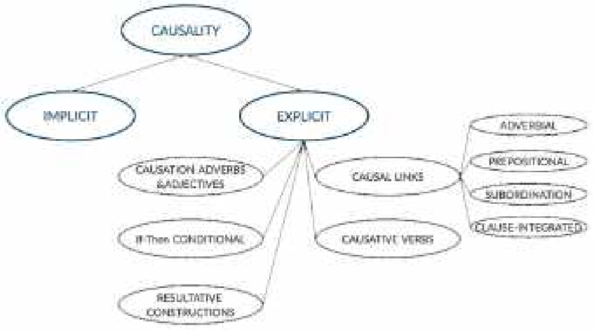


Fig. 1. Causality in natural language

supervised techniques. Consequently, as anticipated, these methods are better suited for managing explicit causal links, with limited coverage for implicit relationships, and are difficult to generalize.

In the paper written by Vivek Khetan, Roshni Ramnani, Mayuresh Anand, Subhashis Sengupta, Andrew E. Fano [9] they investigate the language models capabilities for causal association among events expressed in natural language text using sentence context combined with event information, and by leveraging masked event context with in-domain and out-of-domain data distribution.

A more specific application of causal relations is given in the work written by Son Doan, Elly W.Yang, Sameer Tilak and Manabu Torri [3]. Using natural language processing techniques, the authors assessed a method for extracting health-related causal linkages from Twitter conversations.

The analysis of health-related tweets would assist us comprehend the health conditions and worries we face on a regular basis, especially in the present day.

3 Methods and Approaches

Understanding causal relationships between commonplace events is crucial for common sense language comprehension. The majority of existing causality comprehension techniques rely on language pattern-matching rules or feature engineering to train supervised machine learning algorithms.

The types and structure of causality are illustrated in Figure 1. This section focuses on the methods and approaches utilized in works that directly address the topic in the field of natural language processing.

The paper [9] focuses on understanding the causality between events expressed in natural language text. The intent is simply to identify possible causal relationships between marked events implied by a given sequence of text.

Authors causality understanding approach can be simplified as a binary classification of “Cause-Effect” / “Other” relationship between events expressed in natural language text. The methodology in this work involves:

— Fine-tuning BERT based feed forward network for Cause Effect/Other relationship label between events expressed in natural language text. In this network architecture, authors feed the input sentence as a sequence of tokens to the BERT model and take the overall sentence context vector from the BERT models [2] output, feed it to a non-linear activation layer followed by two fully connected layers. Mathematical formulation for C-BERT model is:

$$H'_0 = W_0(\tanh(H_0)) + b_0, \tag{1}$$

$$h'' = W_1(H_0) + b_1, \tag{2}$$

$$p = \mathrm{softmax}(h''), \tag{3}$$

where W_0, W_1, H_0 is the output token of bi-directional context (i.e. [CLS]) of BERT, and L = 2 (Cause-Effect, Other).

— Combining both the events context and BERTs sentence context to predict “Cause-Effect” / “Other” relationship label between events. This methodology works on the intuition that the interaction between two events is result of the information in the sentence as well as in the events.

They can be more than a single token, resulting in many vectors when the input sentence is fed into a pre-trained BERT model. Authors averaged them to get the final context of each event expression and passed the sentence context as well as both the events context to a non-linear activation layer followed by a fully connected layer. The sentence context is concatenated with both the events averaged context and is feed to another fully connected layer followed by a softmax layer.

The model is trained using backpropagation with Adam-optimizer on a binary loss function to predict the "Cause-Effect" / "Other" relationship between events.

— Combining both the events masked context with BERTs sentence context to predict "Cause-Effect" / "Other" relationship label between events. This network architecture is very similar to the event aware C-BERT network architecture, where the whole span of event text is replaced with a "BLANK" token.

As each event is just a single blank token, unlike Event aware C-BERT we dont need to take an average to get the final context of any event. Each model trained by this approach is then fine tuned using actual event information using the Event Aware C-BERT model described above.

P. P. Markov in his article [11] facts describing cause-and-effect patterns are understood as text objects $s_i \in S$ (the set of vertexes of noun groups, predicates, and definitions that are syntactically consistent with subjects), semantically related by relationships $R_C \subseteq S \times S$, $R_A \subseteq S \times S \times S$ and by relationship groups $RE \subseteq S \times S$.

More detailed description of the connections between objects is also provided. The relationship properties are specified by means of attributes $A = a\{r,\ v\} \in R \times V$, where V is the set of valid attribute values.

Attributes are divided into $A_A \in A$ to describe the properties of symmetry, transitivity, reflexivity, etc. and $AV \in A$ to indicate the values of standard types, for example, to indicate the probabilistic characteristics of cause-and-effect relationships.

The result is formed by searching for all possible substitutions in the arguments of cause-and-effect relationships R_C , taking into account the ordering of objects. In this case, first of all, the facts whose arguments have the maximum weight are output, then, respectively, by reducing the weight. Authors in [3] use two methods for extraction the health related causality from tweets. They are:

— Natural Language Processing (NLP) pipeline. The NLP pipeline for extracting causal relation is summarized as follows: First, the corpus is filtered using the target keywords. Next, a series of basic NLP components are applied: sentence splitter, Part-of-Speech (POS) tagger, and dependency parser. Finally, causal relations are identified based on syntactic relations generated by the dependency parser.

— Cause-Effect Relation Extraction. Authors created a set of six general rules to identify cause-effect relationship from verb and noun phrase. Those rules are based on syntactic relations derived from a dependency graph generated by a dependency parser. For example, a Semgrex [16] pattern =subj < subj (word: /cause/=target > dobj =cause) finds a match in a sentence Stress caused my insomnia, where Stress is matched with the pattern =subj and insomnia is matched with the pattern =cause. Using Semgrex, we extracted the triple <cause, relation, effect> from tweets, where effect is one of the three health-related topics of our focus: insomnia, stress and headache.

The final step is to extract causality from extracted cause effect relations. To do so, we extracted the triple <cause, relation, effect>, where effect is one of the three health-related topics of our focus: insomnia, stress and headache.

4 Data Used in Researches

Authors in work [9] use three different datasets to train and evaluate the models described in previous section. Semeval 2007 [4] and Semeval 2010 [7] is curated using pattern-based web search while ADE is curated from a biomedical text as in [5].

Table 1. Statistics for curated datasets

		Train Dataset		
Dataset	**Max Sentence Length**	**Total**	**Cause-Effect**	**Other**
Semeval2010	(85, 60)	8000	1003	6997
Semeval2007	(82, 62)	980	80	900
ADE	(135, 93)	8947	5379	3568
		Train Dataset		
Dataset	**Max Sentence Length**	**Total**	**Cause-Effect**	**Other**
Semeval2010	(85, 60)	2717	134	2389
Semeval2007	(82, 62)	549	46	503
ADE	(135, 93)	2276	1341	935

Table 2. Comparison of F1 score of models

	Semeval2007	Semeval2010	ADE
C-BERT	93,78	97,68	97,10
Event Aware C-BERT	94,94	98,35	97,85
Masked Event C-BERT + Event Aware C-BERT	95,31	97,85	97,85

1. SemEval 2007 is an evalution task designed to provide a framework for comparing different approaches to classifying semantic relations between nominals in a sentence. For this work, authors use part of the SemEval 2007 dataset with the Cause-Effect relationship.

   For a given sentence, if the interaction between marked events is causal, they label it as “Cause-Effect” else the sentence is labeled as “Other”.

2. Similar to the above dataset, authors use SemEval 2010 dataset with causal interaction between events labeled as “Cause-Effect”, and all the other types of interactions between events in rest of the sentences are labeled as “Other”.

3. ADE dataset [6] is a collection of biomedical text annotated with drugs and their adverse effects.

The first corpus of this dataset has drugs as well as effects annotated. In the second corpus, where drugs are not causing any side-effect, the drug and its effect name are not manually annotated.

Authors curated a list of unique drugs and affect names using the first corpus data and use this set to annotate the drugs and effect names in the second corpus.

While they take sentences with two or more drugs/effect mention in them; for simplicity, we do not replicate the sentence in our final corpus.

To evaluate the precision of causal relation extraction, authors compared system outputs with human annotations. Table 4 provides us with the comparison results. P. Maslov [11] does not allocate a particular dataset. His work is designed using texts of the Russian business prose genre.

Business prose is defined by its strict means of expression, unambiguity of the transmitted information, economy of language means, clarity of the function of each communication, and other advantageous characteristics.

This genre provides information on objects (events, phenomena, people, etc.) that can be represented by an abbreviation of facts provided directly in the examined text. 24 million tweets were employed in the job of identifying health-related causality from Twitter messages [3].

This information was collected over a four-month period from four cities (New York, Los Angeles, San Francisco, and San Diego) (Sep 30, 2013 and Feb 10, 2014). The Twitter Streaming API was utilized to retrieve 1% of all tweets from these cities during the specified time frame. Three terms were chosen as the intended “effects”: stress, sleeplessness, and headache.

5 Achievements in this Field

To be more accurate about the outcomes of applying causality phenomena in natural language processing, It was more suitable to present the results of [11, 9, 3] as the methodologies and datasets have already been described.

The authors of [9] constructed three distinct BERT-based network architectures on each of the datasets to evaluate the language model’s ability to understand the “cause - effect” relationship between events.

Table 2 compares the performance of our models developed utilizing three distinct network architectures and trained on in/out of domain

Table 3. F1 score after pre-training on masked event C-BERT model (dataset 1) and fine-tuning on event aware C-BERT (dataset 2)

Dataset2 Dataset1	Semeval2007	Semeval2010	ADE
Semeval2007	95,31	98,42	97,27
Semeval2010	97,14	98,38	97,47
ADE	96,42	98,49	97,85

Table 4. Precision of extracted causal relations when comparing to human annotators

	Strict evaluation	Relax evaluation
Insomnia	73,81%	88,10%
Stress	82,65%	96,04%
Headache	56,10%	85,37%
Micro-average	74,59%	92,27%

data distribution to previously reported F1 performance metrics.

Table 3 shows the result of another set of experiments where authors examine the performance of the models [8] when pretraining and fine-tuning are conducted using in-domain data distribution rather than when pretraining is performed using out-of-domain data distribution.

They pre-trained three models for each of the target data distributions using the other out of domain data distribution. In general, pre-training on a dataset distinct from the target data distribution resulted in either comparable or enhanced performance.

According to [11] the presented method is at the stage of practical implementation and is made in the form of a system of logical inference of cause-and-effect patterns. The weights of objects and attributes are also partially taken into account.

To receive the results the authors in [3] observed that the number of tweets containing specific health-related cause-effect relationships is small in comparison to the overall corpus. The number of sentences matched by the rules is 501 from 29705 tweets for stress (1.6 %), 72/3827 (1.8 %) for insomnia, and 94/11252 (0.8 %) for headache.

The final causality extracted from the matched sentences are 41, 98 and 42 for insomnia, stress and headache, respectively. To evaluate the precision of causal relation extraction, authors compared system outputs with human annotations. Table 4 shows the comparison results.

6 Conclusion

In conclusion, various theories and methods of causal relationships have already been created. However, we currently face the challenge of incorporating these methods and approaches into natural language processing algorithms.

This paper attempted to highlight the most pertinent and focused papers on causality in natural language processing. As shown by the outcomes of various ways, exploiting causality links can improve the accuracy of algorithms' work, although it remains challenging and problematic to manage this duty entirely.

[9] shows that the network architectures built on top of the contextualized language model can learn causal relations in the text using sentence context, event information, and masked event context. For a comprehensive causal comprehension of events stated in natural language text, we must be able to recognize sentences containing causal events, identify those events and their causal linkages, and comprehend the impacts between those events.

Consequently, there is a target to test the other the other language models as XLNet [18], GPT-2 [12], ELECTRA [1], MT5 [17] and to try the other different approaches to fully implement the causality in NLP methods and work on improving the accuracy of algorithms.

Acknowledgments

This research is funded by the Aerospace Committee of the Ministry of Digital Development, Innovations and Aerospace Industry of the Republic of Kazakhstan (BR11265420).

References

1. **Clark, K., Luong, M. T., Le, Q. V., Manning, C. D. (2020).** Electra: Pre-training text encoders as discriminators rather than generators. DOI: 10.48550/ARXIV.2003.10555.

2. **Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019).** BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

3. **Doan, S., Yang, E. W., Tilak, S. S., Li, P. W., Zisook, D. S., Torii, M. (2019).** Extracting health-related causality from twitter messages using natural language processing. BMC Medical Informatics and Decision Making, Vol. 19. DOI: 10.1186/s12911-019-0785-0.

4. **Girju, R., Nakov, P., Nastase, V., Szpakowicz, S., Turney, P., Yuret, D. (2007).** SemEval-2007 task 04: Classification of semantic relations between nominals. Proceedings of the Fourth International Workshop on Semantic Evaluations, Association for Computational Linguistics, pp. 13–18.

5. **Gopalan, S., Lalithadevi, S. (2018).** Cause and effect extraction from biomedical corpus. Computación y Sistemas, Vol. 21. DOI: 10.13053/cys-21-4-2854.

6. **Gurulingappa, H., Rajput, A. M., Roberts, A., Fluck, J., Hofmann-Apitius, M., Toldo, L. (2012).** Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. Journal of Biomedical Informatics, Vol. 45, No. 5, pp. 885–892. DOI: 10.1016/j.jbi.2012.04.008.

7. **Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S. (2010).** SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, pp. 33–38.

8. **Khetan, V., Ramnani, R., Anand, M., Sengupta, S., Fano, A. (2020).** Causal-BERT: Language models for causality detection between events expressed in text.

9. **Khetan, V., Ramnani, R., Anand, M., Sengupta, S., Fano, A. E. (2021).** Causal BERT: Language models for causality detection between events expressed in text. Lecture Notes in Networks and Systems, Springer International Publishing, pp. 965–980. DOI: 10.1007/978-3-030-80119-9$_{6}4$.

10. **Marasovi, A. (2018).** NLPs generalization problem, and how researchers are tackling it. The Gradient.

11. **Maslov, P. (2008).** Extracting causal patterns from natural language texts. Tavrichesky Bulletin of Informatics and Mathematics, Vol. 13, No. 2.

12. **Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019).** Language models are unsupervised multitask learners.

13. **Schölkopf, B. (2019).** Causality for machine learning. DOI: 10.48550/ARXIV.1911.10500.

14. **Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., Bengio, Y. (2021).** Toward causal representation learning. Proceedings of the IEEE, Vol. 109, No. 5, pp. 612–634. DOI: 10.1109/JPROC.2021.3058954.

15. **Solstad, T., Bott, O. (2017).** Causality and causal reasoning in natural language. pp. 619–644.

16. **Tamburini, F. (2017).** Semgrex-plus: a tool for automatic dependency-graph rewriting. Proceedings of the Fourth International Conference on Dependency Linguistics, Linköping University Electronic Press, pp. 248–254.

17. **Xue, L., Constant, N., Roberts, A., Kale, M., Al Rfou, R., Siddhant, A.,**

Barua, A., Raffel, C. (2021). mT5: A massively multilingual pre-trained text-to-text transformer. Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, pp. 483–498. DOI: 10.18653/v1/2021.naacl-main.41.

18. **Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., Le, Q. V. (2019).** Xlnet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems, Curran Associates, Inc., Vol. 32.

Article received on 20/05/2022; accepted on 14/10/2022. Corresponding author is Alexander Gelbukh.

ISSN 2007-9737

Data Integration for the Evaluation of Cancer Evolution in Mexico through Data Visualization

Obdulia Pichardo-Lagunas, Bella Martinez-Seis,
Fernando-de-Jesus Basurto-Carrillo, David Fernández-Flores

Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria en Ingenería y Tecnologías Avanzadas,
Mexico

{opichadola,bcmartinez}@ipn.mx
{fbasurtoc1500, dfernandezf1501}@alumno.ipn.mx

Abstract. Cancer is the third cause of death in Mexico, one of the chronic degenerative diseases that has grown the most throughout the territory. This document describes a system that allows collecting, integrating, and deploying a unified data repository on cancerology in Mexico. Data were obtained from public access sources generated by units specialized in cancer treatment and follow-up. We use data mining techniques for the integration of the data repository. The application developed allows the analysis of the cancer panorama in Mexico. The project involves cleaning processes, integration, selection, and transformation of data in a pre-processing stage, for further analysis and presentation through a graphical interface. The primary objective is to visualize the general behavior and evolution of cancer in Mexico in recent years using data science techniques.

Keywords. Data science, cancer, data repository, transformation of data.

1 Introduction

Data is considered a resource in today's world. According to the World Economic Forum, by 2025, we will generate around 463 exabytes of data globally per day [1].

Data analytics is a process of data cleansing, transformation, and modeling to uncover valuable insights for business decision-making. [2]. There are several types of data analysis techniques that are based on business and technology. However, the main types of data analysis are: Text analysis, Statistic analysis, Diagnostic analysis, Predictive analysis, and Prescriptive analysis.

Much of the data generated is about the health field. Public and private institutions are interested in storing and analyzing this information for diverse reasons; clear and timely diagnosis, designing and developing public health politics or creating marketing campaigns for treatments or medications. The necessary information for this analysis can come from diverse sources and have different characteristics. Hence, the implementation of Data Analytic techniques is required to normalize and study the obtained registers.

In the last years, the collection and analysis of data about the high rate of mortality diseases have been a challenge for governments and the pharmaceutical industry in the world. For example, cancer is the second cause of death worldwide. In 2018, 9.6 million deaths worldwide were due to cancer; one in six deaths was cancer.

The countries with low and middle income have a more number of deaths from cancer. In 2014 Mexico registered a mortality of 71,900 people for causes associated with cancer; in 2017, this number grew to 904 581, and it is estimated that it will reach 1,262,861 in the next decade [3].

Public and private organizations offer diagnosis and treatment of cancer in Mexico. However, no government institution carries out a registry of cases of cancer.

doi: 10.13053/CyS-26-4-4425

In Mexico, does not exist system for the control and monitoring of patients with cancer. This fracture in the national health system means that statistical data on cancer is spread over different sources. The records that do exist may be old, noisy, inconsistent, and sometimes incomplete data.

The process of data preparation is the foundation for practical analysis. Data preparation is not fully automatic, and it is estimated that it consumes 60% to 80% of the time in a data mining project [4]. This procedure includes integrating data collected from various sources, which must be cleaned for subsequent selection, analysis, and transformation.

Integrating high-quality cancer data is essential to help implement public health policies for preventing and treating this disease in Mexico. This paper proposes implementing data science techniques to obtain, analyze, classify and organize data from different public repositories about cancer in Mexico. In addition to a consultation web interface, where the study's results mentioned above will unfold. The platform makes it possible to fully visualize the impact of this disease in Mexico by observing the incidence and mortality of this disease in the national territory.

When data from multiple sources, such as government systems, must be integrated, they are usually developed, implemented, and maintained independently to meet specific needs [5]. Consequently, data cleansing becomes a complex, extensive, and specific task. It is challenging to automate because each source can contain dirty data and be represented differently, overlap, or contradict each other.

Data visualization consist in drawing graphic to show data. Can use different kinds of tools like scatterplots, histograms, or heat maps. The objective of these displays is mainly descriptive, concentrating on simple summaries. Data visualization is useful for data cleaning, exploring data structure, detecting outliers and unusual groups, identifying trends and clusters, spotting local patterns, evaluating modeling output, and presenting results. The main goal is to visualize data and statistics, interpreting the displays to gain information [14].

2 Data Integration and Visualization in Health

Considering the large amount of data generated nowadays, the diversity of sources, and the characteristics of the information obtained, implementing tools that allow the concentration and visualization of data are necessary.

Specifically, about health data, there are some proposals made for the integration and visualization of information. The objective o these works are to facilitate the collection and observation of the data using computational applications.

2.1 Data Integration for Health

The work proposed by Rahi et al. [6] shows the relevance of data unification with different sources. They conjoined data about Malaria coming from public sector agencies, private healthcare providers, defense forces, railways, industry, and independent researchers. They suggest the creation of an integrated digital platform. The platform will provide real-time epidemiological, entomological, and commodity surveillance data that will be of immediate use to all stakeholders and allow the transparent and evidence-based formulation of malaria control policies [6].

Data analysis will facilitate the identification of potential hotspots of malaria and impending outbreaks. The system contains a set of alerts to inform the activities of monitoring and evaluation. The local or remote databases can be selected, guaranteeing continuous reporting. Considering new vision techniques, we seek to create regarding Cancer in Mexico a work similar to that carried out with malaria by Rahi et al.

As part of German Medical Informatics Initiative, Prasser describes the work made for the consortium Data Integration for Future Medicine (DIFUTURE) will establish Data Integration Centers (DICs) at university medical centers [7].

The DIFUTURE Data Integration Centers will implement a three-step process for integrating, harmonizing, and sharing data as well as images from clinical and research environments: First, Data is imported and harmonized using common data and interface standards using IHE profiles, DICOM and HL7 FHIR. Second, data is pre-processed and enriched within a staging and working environment. Third, data is imported into common analytic platforms and data models (including i2b2 and tranSMART) and made accessible in a form compliant with the interoperability requirements defined on the national level.

Same as other research demonstrated the relevance of data integration and visualization considering the future necessities of medical institutions for research and at the point of care as a basis for targeted diagnosis and therapy.

Another project analyzes the problems in collecting medical records to achieve a unified picture of the progression of cancer disease in Greece [8], Varlamis et al. use the death records for cancer cases collected by the Cancer Registry of Crete (CRC) which is member of the European Cancer Registry (ECR).

This institution collects data from private and public hospitals, for six years, between 1998 and 2004. Data like age, sex, place of birth, residence, occupation, and type of cancer diagnosed in Lasithi and Rethymno counties are available.

Feature selection was applied to assess the contribution of each collected feature in predicting patient survival. Several classifiers were trained and evaluated for their ability to predict patient survival. Finally, a statistical analysis of the two regions' cancer morbidity and mortality rates was performed to validate the initial findings.

The data collected was entered into MSExcel and subsequently imported into SPSS and Weka, where the analysis was performed. The data integration phase was more accessible in this project since having a single unified registry provided by the European registry and efforts focused on pre-processing, completing incomplete records, removing duplicate records, and exploratory analysis of the same data.

The analysis of the data was divided into two areas; using the SPSS tool; an exploratory statistical analysis was carried out to achieve a greater understanding of the composition of the data set; after achieving the objective of the exploratory part, it was realized to the specific analysis, using WEKA. The objective of the analysis process in this project is to achieve a classification model to predict the possibility of cancer survival based on the characteristics with which the dataset was built.

2.2 Data Visualization for Health Data Management

The large amount of clinical data generated in medical practice can create complications for health specialists and experts in implementing public policies that try to understand the health status of patients or vulnerable groups. Different approaches have been adopted to solve the problem of data visualization using computational tools. However, many approaches must be considered. Some of the efforts made by specialists to implement practices in designing and evaluating visualization techniques for clinical data are shown below.

As you can see in ”Data Visualization for Chronic Neurological and Mental Health Condition Self-management: Systematic Review of User Perspectives” Polhemus et al. [9] also describe the necessity of visualization of data for users with special characteristics. For this project, the author collects data from mobile health devices and apps. The main idea is to obtain information of users living with chronic neurological and mental health conditions through data visualizations derived from Remote measurement technologies to manage health.

In this review, they search peer-reviewed literature and conference proceedings (PubMed, IEEEXplore, EMBASE, Web of Science, Association for Computing Machinery Computer-Human Interface proceedings, and the Cochrane Library) for original papers published between January 2007 and September 2021 that reported perspectives on data visualization of people living with chronic neurological and mental health conditions.

The articles were examined by two reviewers who screened each paper based on the abstract and full-text article. The extracted data underwent thematic synthesis.

They identified 35 publications from 31 studies representing 12 conditions. Coded data coalesced

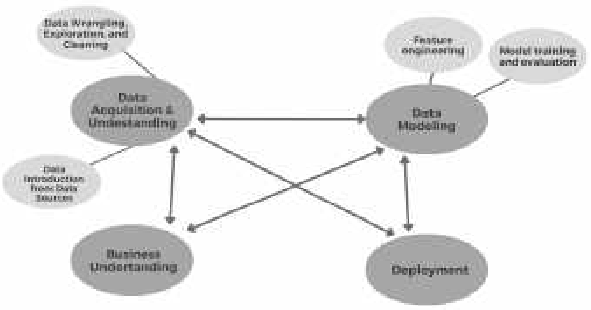

Data Wrangling, Exploration and Cleaning
Feature engineering
Model training and evaluation
Data Acquisition & Understanding
Data Modeling
Data Introduction from Data Sources
Business Undertanding
Deployment

Fig. 1. TDSPs life cycle adapted to the proposed Data Science Process

into 3 themes: desire for data visualization, impact of visualizations on condition management, and visualization design considerations.

The authors can conclude ”When used effectively, data visualizations are valuable, engaging components of RMT. They can provide structure and insight, allowing individuals to manage their own health more effectively. However, visualizations are not “one-size-fits-all”.

In ”Clinical Data Visualization: The Current State and Future Needs”, the authors use the taxonomy proposed by Starren and Johnson to classify the presentation of clinical data [10]. A graphical user interfaces analysis system is proposed in which each interface element is considered an object that can be classified as a list, a table, a generated text, an icon, or a graphic.

The document performs an analysis of the visualization of clinical data from different specialties. The first case analyzes the visualization of cardiac indole data, Blike et al. developed a setup graph to display cardiac parameters, which uses two multi-axis graphs to create geometric objects [11, 12]. These objects were designed to have properties that make it easier to differentiate shock states.

Physicians evaluated physiological data in a computer-based simulation in two separate studies. Using the configuration graph in conjunction with a traditional display improved accuracy and reduced time to diagnose crash conditions compared to the traditional display alone.

In the second example Pulmonary data can be observed, extracting information related to the pulmonary system through the ventilator. Interpreting ventilator data in conjunction with blood gas data allow assessment of a patient’s condition and monitoring of disease processes such as pneumonia and acute respiratory distress syndrome. Integrated lung screens have been developed as an alternative to manual review of individual respiratory parameters.

In 2019 Jiang et al. [13] present a health data visualization system emphasizing geospatial and temporal information integration in healthcare data and focusing on two methods developed specifically for public health data: Spatial Textures and Spiral Theme Plot. The spatial texture technique is used in geospatial visualization that inherently provides additional screen real estate (surface areas) that can be used to encode other data and attributes.

The Spiral Theme Plot technique combines several information visualization methods, including Theme River, Spiral Plot, and Scatter Plot. This combination of public health data with large patient databases satisfies several critical requirements for visualizing time-variant patient records.

Specifically, in the field of public health, the collection and analysis of data from various sources allow for statistical and predictive analyzes that would otherwise be partial or incomplete. Many works are carried out around this topic, focused on different health specialties and with different approaches. Specialists in Mexico require timely and transparent information for decision-making, so systems like the one proposed in this work become necessary.

3 Data Management of Cancer in Mexico

Cancer is a process of uncontrolled growth and spread of cells and can appear practically anywhere in the body, forming a tumor.

Not only prevention and early detection are essential to reduce cancer mortality figures, but also the distribution of resources by location and type of cancer for timely treatment. In this sense, it is important to have an integrated data set to

Table 1. Initial collection of datasets with their description

Data Source	Kind of data	Diagnostic	Dates	Geographic component	Reg. Num.
INEGI	Mortality	Yes	2012-2019	Yes	3301806
Infocancer	Hospital Admissions	Some	2007-2019	Yes	None
CNEGSR (SICAM)	Statistics	Some	2010-2019	Yes	1433
IMSS	Hospital discharges	Yes	2010-2019	Yes (Delegations)	360
CENSIA	Children and youth	Yes	2010-2019	Yes	991
INSP	Nutrition of INSANUT	Some	1994, 2006, 2000	Yes	100
HJM	Laparoscopic surgery	Cancer	2017	Yes	99
INCAN	Admissions and Mortality	Yes	2010-2019	No	2812 and 4013

query, analyze, classify, and predict the evolution of cancer in Mexico.

We adapt the Data Science Lifecycle of Team Data Science Process (TDSP) to this project as you can see in Figure 1; it includes business understanding, deployment, modeling, and data acquisition and understanding. . In the last one, there is the main contribution of this work because it integrates and cleans the data set.

The objective is to lay the foundations and generate a unification of the data sets for future regressions, classifications, groupings, anomaly detection, or recommendations.

3.1 Data Acquisition and Understanding

For the integration process is necessary to have a Data Introduction and a Data Exploration. The first one is to understand where the data came from and its destination for analysis, in this sense we analyze three primary data source platforms: INEGI, Datos Abiertos, and Infocancer. The second one was performed to clean the data, in this sense we normalize data formats, identify missing information, and merged data sets. Those two phases are explained in the next sections.

3.1.1 Data Introduction

Health care is a multidimensional system established for the prevention, diagnosis, and treatment of health-related problems or deficiencies in human beings. In Mexico, there are several health institutions that follow up on Cancer. Some of the data is available through online platforms, in this sense we analyze the content of four of those platforms as Data Sources: INEGI, Datos Abiertos, and Infocancer.

The National Institute of Geography and Statistic (INEGI - Instituto Nacional de Estadística y Geografía) is the main organism in charge of collecting and disseminating of information in Mexico. From this source, we, initially, collected different data sets related to deaths from 2012 to 2019.

Infocancer is a project of the National Institute of Cancer (INCan - Instituto Nacional de Cancerología), from this source we collected statistics related to hospital admissions and mortality from 2007 to 2019.

Open Data (Datos Abiertos) is a public platform from de Federal Government to publish data, from this platform we used data collected by CNEGSR (Centro Nacional de Equidad de Género y Salud Reproductiva), IMSS (Instituto Mexicano del Seguro Social), CENSIA (Centro Nacional para la Salud de la Infancia y la Adolescencia), Goverment of Puebla State, INSP (Instituto Nacional de Salud Pública), HJM (Hospital Juárez de México), INCAN

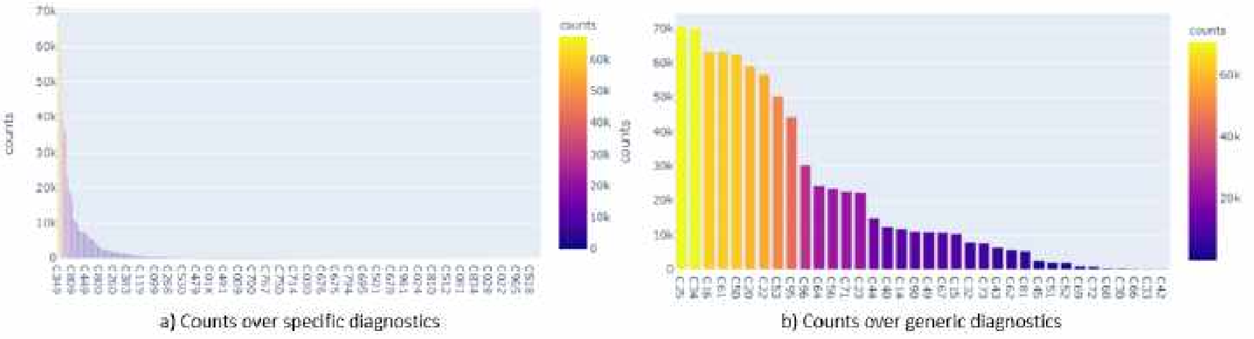


Fig. 2. Number of cancer cases for each kind of cancer

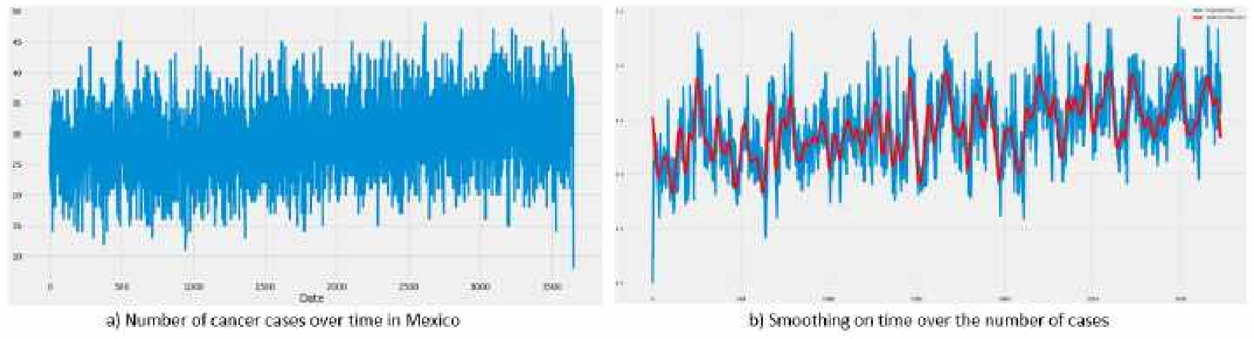


Fig. 3. Original number of cases an smoothing on time by two methods

(Instituto Nacional de Cancerología), Goverment of Jalisco, and Secretaría de Salud.

Table 1 shows the collected datasets with their description of the kind of that they have. It is important to know where the cancer cases are, when did they happen, and what type of Cancer is presented as diagnostics.

We can see that the data sets of INEGI have a big amount of data, with the proper data cleaning, it will be useful for finding tends; on the other hand, Infocancer provides statistics reports that are useful as reference but do not for data mining.

From this initial collection, we got 27 files with 3311614 records.

3.1.2 Harmonized Data Model

Data harmonization refers to combining data from different sources minimizing redundant or conflicting data.

For data merging, three axes were considered on which the data would be combined, so that the records to be considered should have a temporal identifier, a geographic identifier, and a diagnosis.

It is also considered that data display will require elements related to age groups, types of cancer, state of origin of the patient or death, and year of registration. In this sense, the final data set uses two main sources: INEGI and IMSS for a period of 10 years, from 2010 to 2019.

The year 2020 was an atypical year because of the pandemic, so it was no considered. The collection from the first source was extended to have a data frame with 6557201 records with 59 characteristics related to mortality. The collection of the second data source was also expanded up with 43 characteristics related to hospital discharges with a total of 23 files with 6782130 records.

In both cases, the cause of death and the final diagnosis is a good discrimination point for the data, as we can identify only cancer-related fields. We used the international standard with the ICD-10 codes. There are 452 keys to specific cancer diagnostic.

Under the advice of a medical team, the types of cancer that can cover various conditions were selected, reducing this list from 452 specific diagnoses to 36 generic diagnoses. A dictionary was then constructed to identify to which generic cancer type each specific diagnosis belongs. This reduction was significant as it is more manageable for medical personnel.

Figure 2 shows the mapping of the specific diagnostics (Fig. 2a) provided by the original source to a corresponding ICD-10 code (Fig. 2b), where the cause column represents the diagnostics. For example, 17 different malignant tumors located in the external and internal parts of the mouth such as lips and tongue correspond to a single generic key C14 corresponding to Oral Cavity and Pharynx Cancer. This mapping was performed with the support of physicians with various specialties.

For the geographic component, we mapped the data into official geographic keys related to states and municipalities. There are 2475 municipalities. For this cleaning, null data was eliminated and a staggered patron for states and municipalities was done.

Another required data cleaning was related to age, some inconsistencies were detected about the age and some dates. Then an operation was done to calculate the age that should be correct; we detected 244807 (31.12% of the full data) registers with inconsistencies, from those ones, about 244694 had a difference lower than a year; some others were related to babies younger than one-year-old. We selected the proper age and eliminated noisy data.

The original dataset has separate columns containing the information of date, so using lambda expressions to join according to certain columns, daily records, and monthly records can be obtained. We look for errors that consider leap years. For data consistency, we convert daily registers to monthly ones.

Finally, the data set uses CSV and JSON format, it was stored in the no relational database, moreover a documental NoSQL database.

3.2 Data Modeling

For Data Modeling we consider Feature engineering and model training and evaluation. The first one allows us to get better features for the model and for the visualization; and the second one focus on prediction. The harmonization over the data set allows us to use properly the two final data sets: mortality and occurrences.

First, exponential smoothing methods and the Hodrick Prescott method were used to reduce noise and better mark the trend, so this would be the last preprocessing step before feeding the data to the prediction model with LSTM recurrent neural networks.

Then, we focused on obtaining and predicting trends by modeling time series derived from data collected from different data sources. The algorithms used for the analysis stage are based on regression algorithms, applying supervised Deep Learning algorithms in the form of LSTM recurrent neural networks. But first, we used Fig. 4.

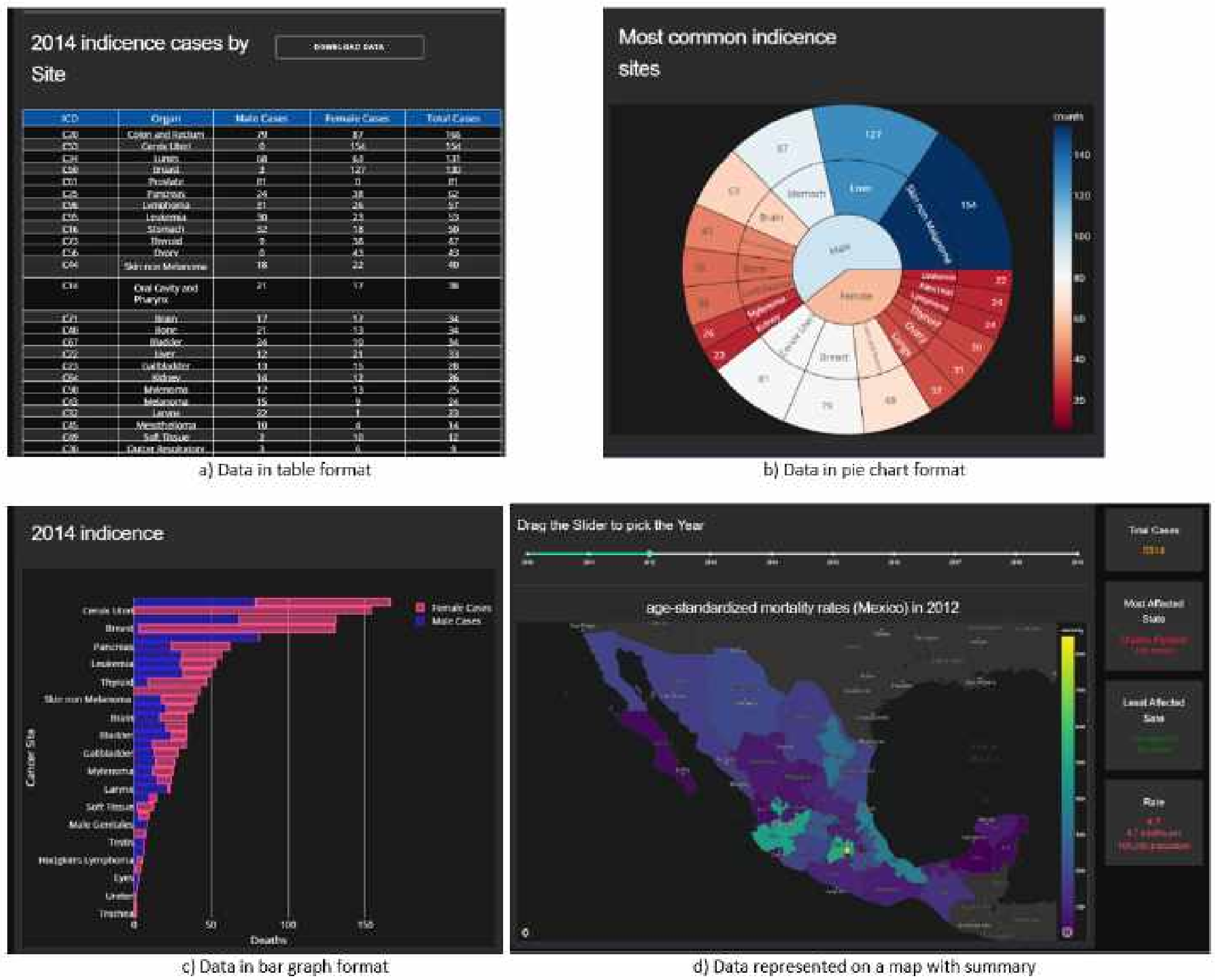
3.2.1 Data Smoothing

We compare two smoothings: exponential smoothing methods and the Hodrick Prescott method. Exponential smoothing is a rule-of-thumb technique for smoothing time series data. It uses the exponential window function. Given the register x_t from the beginning time $t = 0$, and s_t the best estimated value of x, then:

$$s_t = \alpha_t + (1 - \alpha)s_{t-1}, t < 0.$$

where α in the smoothing factor, and $0 < \alpha < 1$, in order to observe the influence of the smoothing on the original set.

On the other hand, the Hodrick-Prescott method identifies the tendency components τ_t in a temporal series y_t. The optimization problem minimizes the deviation of the original series from the trend (the first term of the equation) as



2014 indicence cases by Site

DOWNLOAD DATA

ICD	Organ	Male Cases	Female Cases	Total Cases
C20	Colon and Rectum	79	87	166
C53	Cervix Uteri	0	154	154
C34	Lungs	68	63	131
C50	Breast	3	127	130
C61	Prostate	81	0	81
C25	Pancreas	24	38	62
C96	Lymphoma	31	26	57
C95	Leukemia	30	23	53
C16	Stomach	32	18	50
C73	Thyroid	9	38	47
C56	Ovary	0	43	43
C44	Skin non Melanoma	18	22	40
C14	Oral Cavity and Pharynx	21	17	38
C71	Brain	17	17	34
C40	Bone	21	13	34
C67	Bladder	24	10	34
C22	Liver	12	21	33
C23	Gallbladder	13	15	28
C64	Kidney	14	12	26
C90	Myeloma	12	13	25
C43	Melanoma	15	9	24
C32	Larynx	22	1	23
C45	Mesothelioma	10	4	14
C49	Soft Tissue	2	10	12
C39	Other Respiratory	3	6	9

a) Data in table format

b) Data in pie chart format

c) Data in bar graph format

Drag the Slider to pick the Year
age-standardized mortality rates (Mexico) in 2012
Total Cases
Most Affected State
Least Affected State
Rate

d) Data represented on a map with summary

Fig. 4. Diversification in data visualization

well as the curvature of the estimated trend (the second term).

The trade-off between the two goals is governed by the smoothing parameter λ. The higher the value of λ., the smoother is the estimated trend [15]:

$$min_t(\sum_{t=1}^{T}(y_t - \tau_t) +$$

$$lambda \sum_{t=2}^{T-1}[(\tau_{t+1} - \tau_t) - (\tau_t - \tau_{t-1})]^2).$$

For the present analysis, we used a smoothing factor λ. of 10^3.

Figure 3a shows the number of cases in Mexico between 01-01-2010 and 31-12-2019. Figure 3b shows the smoothing over the data using both algorithms; the blue line in Figure 3b shows the exponential smoothing while the red line in Figure 3b shows the Hodrick-Prescott smoothing. Visually, it is easier to see if the count tends to low or high.

We can see that the tendency is more evident with the Hodrick-Prescott method, then we used it as a preprocessing of the data for the next model.

Table 2. Comparison of the prediction and validation data sets

	Trend	Predictions	Validation
3088	230.830512	231.451614	230.830512
3084	230.886462	231.325180	230.886462
3085	231.051812	231.208832	231.312671
3086	231.312671	231.115479	231.312671
3087	231.639491	231.062851	231.639491

3.2.2 Model Architecture

The objective is to obtain a prediction of changes in the trend of cancer incidence and mortality in Mexico. In this sense, we used a Recurrent Neural Network (RNN) which adapts properly to time series. Specifically, we used a Long Short Term Memory (LSTM). LSTMs can be used to model univariate time series forecasting problems. These are problems composed of a single series of observations and a model is required to learn from the series of past observations to predict the next value in the sequence.

After the normalization process of features, we used a stacked model for the definition of the LSTM. Stacked LSTMs are now a stable technique to challenge sequence prediction problems. A Stacked LSTM architecture can be defined as an LSTM model composed of multiple LSTM layers.

Our network architecture included 3 LSTM layers followed by 3 Dense layers. The first LSTM with 64 neurons and the next 2 with 32 neurons working with 3D matrices. The output of the third LSTM will be a 2D matrix required by the next 3 Dense layers.

The first dense layer was 16 neurons with a ReLu activation function which is a rectified linear activation function, which is useful for stochastic gradient descent with error backpropagation to act as a linear function without being linear. It also provides more sensitivity to the activation sum input and avoids saturation.

The second Dense layer also has a ReLu activation function but with 8 neurons. The third Dense layer has a linear activation function of one neuron since it will be the Output Layer.

3.3 Visualization Process

For the deployment, we consider the information we need to display, in this case, the 3 main pivots that were mentioned before: geographic component, the diagnostic, and time. Other parameters that were considered are gender, age group, and the indicator (incidence or mortality).

We show the data in four different ways. Figure 4a shows data in table format. Figure 4b shows data in pie chart format, where the user can select a subset, for example, he can select Female and the pie chart will change to show just cancer in females. Another way to represent the information is through a bar graph as we can see in Figure 4c.

Finally, a map representation of the data was implemented, the user can select in a control menu that is on the left the gender, indicator, age category, state, and cancer sites.

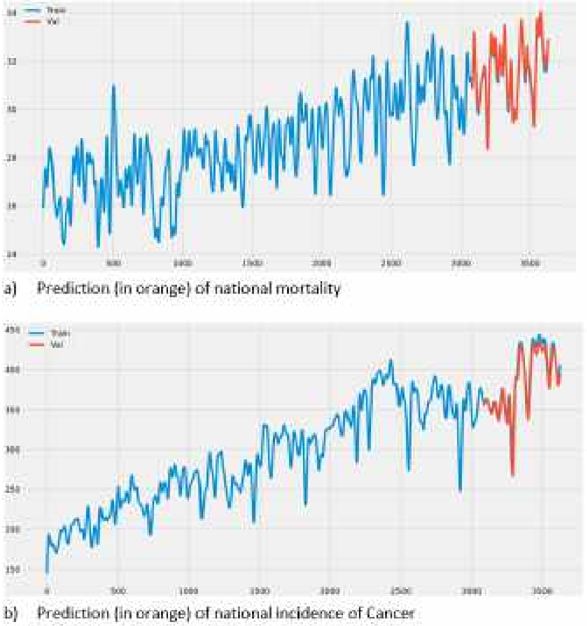
The web application uses Dash-Plotly as a framework.

4 Results

This project focuses in all the data science process, nevertheless an analysis was done by predicting trends when modeling time series. It derived from data collected from different data sources. We used the architecture of the model LSTM recurrent neural networks that was previously presented.

Data sets was divided into test, train, and validation sets, with 85%, 10%, and 5%. All implementations of these models are done by Keras, which is an extremely useful library part of TensorFlow. After defining the model architecture, the model is compiled, using the ADAM optimizer and as an evaluation metric using Mean Absolute Error (MAE).

Table 2 shows some of the predictions and the validation numbers were we can see they are similar. The RMSE obtained is 0.541741, which is a fairly good performance, remembering that these metrics are relative to the data set on which we are working, this is much better represented graphically. Figure 5a shows the prediction of the mortality, in orange, we can see the predicted data, and in blue the validation.



a) Prediction (in orange) of national mortality

b) Prediction (in orange) of national incidence of Cancer

Fig. 5. Graphical comparison of the prediction

Figure 5b shows a similar prediction but in incidences of cancer in Mexico.

5 Conclusions

A repository of clean data on cancer incidence and mortality in Mexico was built over a period of ten years from sources such as the Open Data platform, INEGI, and IMSS.

The data transformation was achieved through smoothing and normalization techniques to be able to use them as input in a prediction model based on neural networks. Obtaining and predicting cancer incidence and mortality trends were validated, obtaining favorable results supported by performance evaluation metrics.

An interface was generated that allowed the visualization of the data obtained and the prediction work carried out on them. The system can be validated through the proposed metrics (MAE, RMSE) and guarantee the availability of the application to simultaneous users.

References

1. **Desjardins, J. (2019).** How much data is generated each day?. World Economic Forum, The Digital Economy, https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/

2. **Johnson, D. (2022).** What is data analysis? Types, process, methods, techniques. Guru99, https://www.guru99.com/what-is-data-analysis.html.

3. **Mohar-Betancourt, A., Reynoso-Noveron, N., Armas-Texta, D., Gutierrez-Delgado, C., Torres-Dominguez, J. A. (2017).** Cancer trends in Mexico: essential data for the creation and follow-up of public policies. Journal of Global Oncology, Vol. 3, No. 6, pp. 740–748.

4. **DataPreparator. (2012).** What is data preparation? https://www.datapreparator.com/what_is_data_preparation.html.

5. **Rahm, E., Do, H. H. (2000).** Data cleaning: Problems and current approaches. IEEE Computer Society Technical Committee on Data Engineering, Vol. 23, No. 4, pp. 3–13.

6. **Rahi, M., Sharma, A. (2020).** For malaria elimination India needs a platform for data integration. BMJ Global Health, Vol. 5, No. 12, pp. e004198. DOI: 10.1136/ bmjgh-2020-004198.

7. **Prasser, F., Kohlbacher, O., Mansmann, U., Bauer, B., Kuhn, K. A. (2018).** Data integration for future medicine (DIFUTURE). Methods of information in medicine, Vol. 57, pp. e57–e65. DOI: 10.3414/ME17-02-0022.

8. **Varlamis, I., Apostolakis, I., Sifaki-Pistolla, D., Dey, N., Georgoulias, V., Lionis, C. (2017).** Application of data mining techniques and data analysis methods to measure cancer morbidity and mortality data in a regional cancer registry: The case of the island of Crete, Greece. Computer Methods and Programs in Biomedicine, Vol. 145, pp. 73–83. DOI: 10.1016/j.cmpb.2017.04.011.

9. **Polhemus, A., Novak, J., Majid, S., Simblett S., Morris, D., Bruce, S., Burke, P., Dockendorf, M. F., Temesi, G., Wykes, Til. (2022).** Data visualization for chronic neurological and mental health condition self-management: Systematic review of user perspectives. JMIR Ment Health, Vol. 9, No. 4., pp. e25249. DOI: 10.2196/25249.

10. **Wanderer, J. P., Nelson, S. E., Ehrenfeld, J. M., Monahan, S., Park, S. (2016).** Clinical data visualization: The current state and future needs. Journal of Medical Systems, Vol. 40, No. 12, pp. 1–9. DOI: 10.1007/s10916-016-0643-x.

11. **Blike, G. T., Surgenor, S. D., Whalen, K., Jensen, J. (2000)**. Specific elements of a new hemodynamics display improves the performance of anesthesiologists. Journal of Clinical Monitoring and Computing, Vol. 16, No. 7, pp. 485–491. DOI: 10.1023/A:1011426226436.

12. **Blike, G. T., Surgenor, S. D., Whalen, K. (1999).** A graphical object display improves anesthesiologists' performance on a simulated diagnostic task. Journal of Clinical Monitoring and Computing, Vol. 15, No. 1, pp. 37–44. DOI: 10.1023/A:1009914019889.

13. **Jiang, S., Fang, S., Bloomquist, S., Keiper, J., Palakal, M., Xia, Y., Grannis, S. (2016).** Healthcare data visualization: Geospatial and temporal integration. Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Vol. 2, pp. 214–221.

14. **Unwin, A. (2020).** Why is data visualization important? What is important in data visualization?. Harvard Data Science Review, Vol. 2, No. 1.

15. **Kenton, W. (2019).** Hodrick-Prescott (HP) filter. Investopedia, https://www.investopedia.com/terms/h/hpfilter.asp.

Article received on 07/07/2022; accepted on 23/09/2022.
Corresponding author is Bella Martinez-Seis.

ISSN 2007-9737

Combined Detection and Segmentation of Overlapping Erythrocytes in Microscopy Images Using Morphological Image Processing

Lariza M. Portuondo-Mallet[1,2], Lyanett Chinea-Valdés[1], Rubén Orozco-Morales[3],
Juan V. Lorenzo-Ginori[1]

[1] Universidad Central "Marta Abreu" de Las Villas,
Centro de Investigaciones de la Informática,
Cuba

[2] Universidad de Oriente,
Centro de Estudios de Neurociencias,
Procesamiento de Imágenes y Señales (CENPIS),
Cuba

[3] Universidad Central "Marta Abreu" de Las Villas,
Centro de Estudio de Métodos Computacionales
y Numéricos en la Ingeniería (CEMNI),
Cuba

lportuondo@uo.edu.cu, {rorozco, juanl}@uclv.edu.cu

Abstract. Segmentation of clusters of erythrocytes into their constituent single cells is a procedure needed in various biomedical applications related to microscopy images. This task is part of the general problem of splitting clumps of objects in images which continues being an open research topic in the Image Processing field. This work presents a unified morphological method to detect and segment clusters of erythrocytes in microscopy images, and proposes two main contributions. The first one is to formulate and evaluate a method to detect clusters as connected components in binary images, obtained from a previous coarse segmentation, which is not capable of further dividing a cluster into its constituent cells. Secondly, to propose the best alternative to split the clusters into their constituent individual cells after evaluating three algorithms based in the combination of the transforms: *H*-maxima, weighted external distance and marker-controlled watershed. Evaluation of the proposed cluster detection methods was made in terms of standard measures of effectiveness. Segmentation accuracy was evaluated comparing the segmented objects obtained to a manually segmented ground truth, by means of the Jaccard index. Then the Friedman test allowed validating the advantages of the proposed method in comparison to the other alternatives studied here.

Keywords. Image segmentation, clusters splitting, watersheds, distance transform.

1 Introduction

1.1 General Background

Segmentation of clusters of overlapping or touching objects in binary images into their single components has been addressed in a variety of practical situations and continues being an open research topic in Image Processing.

Examples can be found for the case of two-dimensional gel electrophoresis overlapping spots [36], segmentation of rocks in images with application to mining industry [4] and rock particles in general for their recognition [39].

doi: 10.13053/CyS-26-4-3893

Other examples are applications related to nanotechnology [43] and to agriculture and food [3], general automated size analysis in multi-flash imaging [21] as well as numerous applications in the biomedical field, among which segmentation of overlapped or touching erythrocytes in microscopy images, to which this work is devoted, is an important example.

A classification of the segmentation methods used in a specific biomedical application is presented in [19] where various approaches like methods based in concave point detection, blob detection, clustering and morphological processing are recognized and discussed.

Other examples are splitting of clumped or overlapped cells based on template matching strategy [7] and a method called Recursive Water Flow (RWF) [8] for cell splitting in histological images. The problem of segmenting touching cells in a 3D framework is addressed in [23].

Segmentation of histopathological images including overlapped or touching cells was addressed in [13] using deep learning algorithms and spatial relationships. Splitting of 3D cell clusters for the case of volumetric confocal images is presented in [15].

A combined method for overlapped cell detection and segmentation based in features obtained from the skeleton and the contour of the cells is showed in [16]. A semi-automatic approach for detection and segmentation of cell nuclei based on graph-cuts and Laplacian of Gaussian (LoG) filtering is proposed in [1].

A method based on concave points extraction through polygonal approximation and ellipse fitting bubbles with average distance deviation criterion and two constraint conditions was addressed in [45]. Reference [44] employed a modified version of curvature scale space method to extract corner points and then recognize the concave points by evaluating angular changes.

These concave points and the centroid points are then used to characterize the structure of the cell clump and to construct the split line by using the corresponding splitting strategy. Other approach proposed recently to split overlapped cells based on elliptical shape models appers in [29].

Various approaches to segment clusters in images from the Papanicolaou test are presented in [31, 32, 33, 38] and other diverse microscopy image applications using methods not based in mathematical morphology were reported in [28, 27, 34, 41].

The method proposed in this work uses an approach to segment clusters based in morphological image processing techniques and under such view, we will comment about methods of this kind in more detail. A method to split cell clumps based in the use of different morphological scales after iterative erosion to find cell-specific markers is developed in [37].

In spite of the good results they obtained, the authors point out that at the time of their publication a comprehensive benchmark using a database of cell clumps or clumped objects was not available. It is worth to notice, however, that to our knowledge such benchmark does not exist yet.

A morphological method is presented in [17] based in the use of an adaptive *H*-minima transform together with an external distance and marker-controlled watershed transform to segment cell clusters, with good results in terms of percentages of correctly segmented clusters.

Reference [18] followed this line of work and it was introduced there a parameterization using an ellipsoidal modeling of contours to perform a more appropriate analysis. The authors expressed their results there in terms of percentages of correctly split clumps.

Various alternatives of the use of markers considering minima imposition were studied in [6] where a relative equivalence was found between different approaches, to represent the markers used to control the watershed transform in order to split the clumps.

Other morphological approach using the watershed transform complemented with a corner detection algorithm appears in [26]. The classical watershed and distance transforms are used in [40], specifically to segment chromosomes showing overlapping.

An improved ultimate erosion process (UECS) together with an edge-to-marker association is proposed in [30] to separate the overlapping convex objects in electron micrographs.

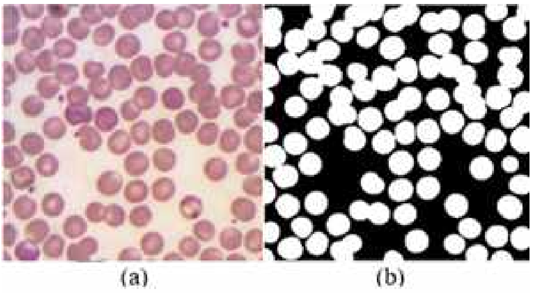


Fig. 1. Microscopy image and coarse segmentation (a) original image, (b) coarse segmentation from image (a)

In this work, the authors used a noise-robust measure of convexity (or concavity) based on the sensitivity to the coarseness of digital grids as the stopping criterion for erosion. The missing contours of the occluded particles are inferred using a Gaussian mixture model on B-splines.

In reference [42] the gradient-barrier watershed algorithm is proposed, in which the gradient in the overlapping region is used directly as the barrier to the water flow. A cluster segmentation method based in the use of structural features and morphological image processing is showed in [20], again obtaining high accuracy in terms of performance measures (sensitivity, specificity) of the cluster detection process as well as accuracy of segmentation.

A review of the use of mathematical morphology techniques in malaria studies, which includes the segmentation of overlapped cells is presented in [25]. Reference [22] presents a method based in a watershed algorithm that iteratively identifies markers, considering a set of different *h* values in the *H*-minima transform.

This method showed good results, but it is oriented to the specific case of wide-field fluorescence microscopy images and requires calculating a fair gradient map from the original image as well as defining heuristically some parameters.

Recently, deep learning algorithms, in particular convolutional neural networks (CNN) have been also applied in medical image analysis [24]. The fully convolutional neural network U-Net [35] has significantly influenced the field of cell segmentation.

This network model was designed to work with few training images and to obtain accurate segmentation. In [2] deep learning was applied to predict cell nuclei and combined with thresholding and watershed transform to segment different types of cells.

Their approach was developed only for fluorescent images with stained cytoplasm. A modified version of U-Net called MultiResUnet is proposed in [14] and obtained better results than using the classical U-Net.

In reference [12] is proposed a method called BubCNN which employs a Faster region-based CNN (RCNN) detector module to locate bubbles and a shape regression CNN to predict bubble shape parameters.

A great future can be foreseen for deep learning based models in this kind of applications, however training deep networks tends to be computationally expensive and might require large numbers of annotated data, which is a time-consuming process. This implies that other conventional image processing techniques like those presented here can be still a valuable choice for the task addressed in this work.

1.2 Unified Framework for Detection and Segmentation of Clusters

We introduce in this work a unified method oriented to segment with high effectiveness clusters having up to medium complexity, which means roughly less than 30% overlapping which could be considered to allow a useful individual cell analysis after splitting. We mention also that erythrocytes consist usually in round-like objects of moderately variable sizes.

The algorithm used to segment the clusters operates by means of a combination of the conventional distance transform, the *H*-maxima transform, morphological operations and a weighted external distance transform combined with marker-controlled watershed segmentation, as will be described in detail later. This allowed using the information obtained during the clusters detection to facilitate their subsequent segmentation (split).

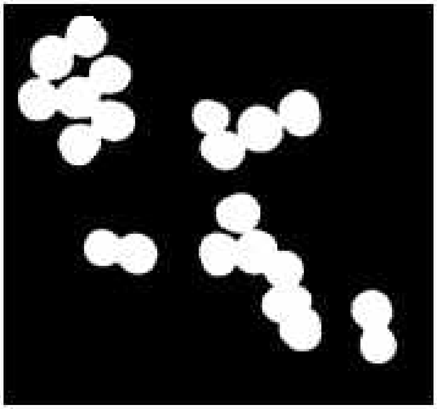


Fig. 2. Connected regions forming clusters

The method presented here showed a high effectiveness in detecting the clusters in terms of performance measures like sensitivity, specificity, accuracy, precision and F-measure, as well as a high segmentation accuracy. The latter was measured in terms of the Jaccard index obtained when comparing the computer-segmented objects to a manually segmented ground truth.

2 Materials and Methods

2.1 Images Dataset

The whole detection and segmentation process begins with a coarse segmentation, which produces a binary image in which the touching or overlapping erythrocytes remain as connected components. The binary images used in our experiments contain clusters of various sizes and were obtained through coarse segmentation of microscopy images, which correspond to mice peripheral blood smears stained with Giemsa.

Other components of the blood smears as leukocytes and platelets were eliminated from the image using image processing techniques, not described here as our interest resides in the segmentation of the remaining clusters of erythrocytes.

A Zuzi microscope model 148 was used to acquire the images, equipped with a plan-achromatic lens having 1.25 numerical aperture and a 0.5 magnification of the camera adaptor, with a 319CU digital camera of 3.2 megapixel and 8-bit RGB uncompressed output, obtaining a resolution of 2048 $\times$ 1536 pixels. The objective power used was 100$\times$ with immersion oil, obtaining a total magnification of 50$\times$, which results roughly in around 140 pixels per cell diameter for the images employed in the experimental work.

The images were saved in .tiff (tagged image file) format. Then, the images were segmented by thresholding to obtain the set of binary image containing independent, single objects as well as clusters of various sizes and complexity.

Other steps in this process included conversion to grayscale prior to thresholding and then, morphological area-opening filters are used to remove items smaller than a red blood cell and to fill the holes left after thresholding. We stress the fact that this primary ”coarse” segmentation is not of concern to this research and its role was only to obtain images containing appropriate clusters to perform the experimental work.

The dataset created consists of 43 images containing in total 4265 binary objects, 1081 of which can be considered as clusters and 3184 as individual cells. Fig. 1 shows an original image and its corresponding binary image after coarse segmentation and Fig. 2 exhibits four examples of connected regions forming clusters.

2.2 Detection of the Clusters Contained in the Binary Images

To detect the clusters contained in the binary images that were obtained as described in the previous section, we followed a method that uses both the conventional (inner) distance transform, the external distance transform (*EDT*) and a weighted version of it (*WEDT*) as well as the *H*-maxima transform and some morphological processing operations, in a process described in detail in what follows.

This approach was used because it produces the inner markers needed afterward in the splitting process. The distance transform $DT(A)$ described in [9] is defined in the following manner: for any point x in A, $DT(A)(x)$ is the distance from x to the complement of A:

$$DT(A)(x) = \min\{d(x,y), y \in A^C\}. \tag{1}$$

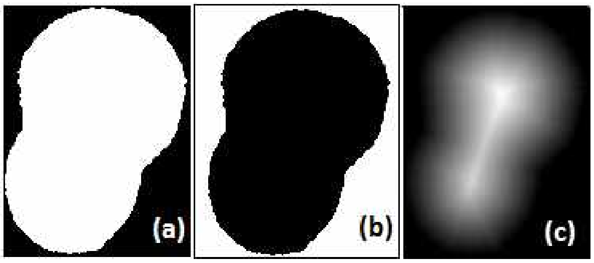


Fig. 3. (a) Binary image from overlapping cells. (b) Complemented binary image. (c) Distance transform

To calculate $DT(A)(x)$, firstly the binary image from the coarse segmentation, which has one-valued foreground pixels, is complemented. Then $DT(A)(x)$ is calculated as the distance from each zero-valued pixel to the nearest one-value pixel.

As the inner distance transform is applied here to the complement of a connected component, its result is a grayscale image exhibiting its highest intensity in a point or patch, which is in general a regional maximum, located farthest from the background. This process is depicted in Fig. 3.

The eventual appearance of spurious maxima will be addressed later. To define the external distance transform, consider the set B of pixels in the background (binary level 0) of the binary image under analysis.

Then for any point $x \in B$, $EDT(B)(x)$ is the distance from x to the nearest pixel pertaining to a marker point (binary level 1), usually taken as a regional maximum as described in the previous paragraph:

$$EDT(B)(x) = \min\{d(x,y), y \in B^C\}. \qquad (2)$$

The proposed methodology followed a sequence of steps to determine whether a connected component in the binary image (obtained from the previous coarse segmentation) corresponds to a cluster or to a single object and then, split those that are considered as clusters. These steps were:

1. Labelling the connected components and calculating the inner distance transform map for each one.
2. Obtaining the valid regional maxima of the distance transform (*DT*) for each binary object present in the image).
3. Classify as clusters all the objects having more than one of these maxima.
4. Build the skeleton by influence zones (*SKIZ*) [9] which correspond to the regional maxima for each cluster, using the *weighted EDT* (*WEDT*), which is the *EDT* with its values divided (weighted) by a factor obtained during the selection of the valid regional maxima described in the next section.
5. Segment the clusters into their constituent components by means of the marker controlled watershed transform [9], using the *SKIZ* lines as external markers and the regional maxima as inner markers.

When building the *EDT* map in setp 4 to obtain the *SKIZ*, the distances from a background pixel to each regional maximum were weighted by a coefficient, which depends on the magnitude (height) of the regional maximum, previously normalized to the interval [0, 1]. Segmentation by means of the watershed transform followed the previous steps.

We point out, however, that obtaining valid regional maxima corresponding to the clustered binary objects is not a trivial task. The clusters may have a moderately irregular contour, and therefore several spurious maxima can appear after calculating the distance transform.

These spurious maxima are usually deemed as noise and can lead to over segmentation when used as markers for segmenting using the marker-controlled watershed transform.

2.3 Determining the Valid Regional Maxima in *DT(A)(x)*

In this work, three methods were applied and compared in order to determine the valid regional maxima present in the binarized clusters.

— Method 1: Iterative *H*-maxima transform. This method apply iteratively the *H*-maxima transform to the distance transform map of each complemented binary objects and afterwards counting the number of remaining regional maxima.

— Method 2: Morphological filtering. This method has the purpose of transforming the set of spurious regional maxima formed around the center of a single (and perhaps part of a cluster) object into one valid, unique maximum.

In this case, an alternating open-close sequential filter [9] with two stages and a disk structuring element is applied to the distance transform map. Then, the algorithm extracts the regional maxima and the magnitude (height) of these resulting maxima is considered representative of that of the individual merged maxima.

— Method 3: Radon transform. This method is described in [11] where the Radon transform and morphological operations are used to find the markers for the erythrocytes.

A detailed description of these methods is presented in the next section.

2.4 Detailed Description of the Methods Used to Detect Clusters

The *H*-minima and *H*-maxima transforms are powerful tools to suppress undesired minima or maxima in a grayscale image.

In this case, we applied the *H*-maxima transform to the distance transform map corresponding to the complemented binary image, obtained from the coarse segmentation step. The *H*-maxima transform *HMAX* is defined in [9] as:

$$\mathrm{HMAX}_{h,D}(f) = f \triangle_D (f - h), \tag{3}$$

where $\triangle_D$ is the morphological operation of geodesic reconstruction, *f* is the intensity image, *h* is a height parameter and *D* is the structuring element. The *HMAX* transform removes any intensity dome in the image having height less than *h* and decreases the height of the other domes by *h*. Calculation of *HMAX* tends to eliminate successively the spurious maxima of different heights as the parameter *h* increases by iterative steps. Once the spurious maxima are eliminated or merged, if the parameter *h* continues increasing, at some moment the regional maxima pertaining two adjacent clustered objects will also merge.

This fact is used in as stop criterion in [17], where the dual *H*-minima transform is used in an analogous way. The algorithm in this reference goes back one step to keep isolated the regional minima pertaining to different adjacent merged objects.

However, increasing *h* in small steps until merging the maxima from adjacent objects implies in our case an unnecessary computing burden, because actually there is only the need to suppress the spurious maxima, which will occur after only some few steps.

In order to find a practical solution to this problem, experimental work with a large number of diverse clusters was performed, testing the results of iterations increasing the parameter *h*.

It was found experimentally that the number of maxima stabilizes in the desired value after at least five successive iterations in practically all cases, without further decrements in the number of maxima until the merging phenomenon previously mentioned occurs.

This determined the use as stop criterion for the iterative *H*-maxima transform the constancy of the number of detected regional maxima during five successive iterations. If after this convergence more than one maximum remain present in a connected component being analyzed, it is possible to say that we are in presence of a cluster, given that a single erythrocyte would show only one maximum.

Then, the maxima obtained for the different components in the image are saved. These maxima will be used later as internal markers to be used in the watershed segmentation, together with the last height value obtained from the *HMAX* transform, which will be also used for separating the clusters into individual objects. On the basis of the previous discussion, three methods to detect clusters were implemented and compared, whose algorithms are summarized as follows:

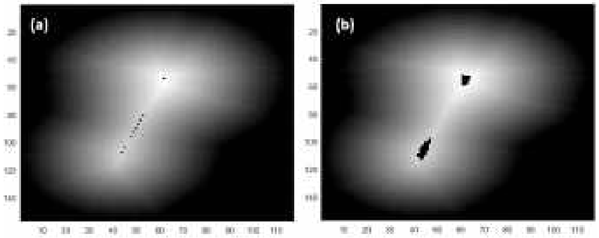


Fig. 4. (a) Regional maxima superimposed to the distance transform map in Fig.2, notice the presence of multiple spurious maxima. (b) Final regional maxima after the iterative search, where the spurious maxima have been merged into two single ones, as expected

2.4.1 Method 1: Iterative *H*-maxima Transform for Detecting Clusters:

1. Perform the coarse segmentation of the image using a standard method and label the resulting binary connected components, which can be either single objects or clusters.

2. For each labeled object *i* do:

   a) Compute the distance transform (Euclidean) on the complement of the *i*th connected component and normalize the obtained grayscale image *Dmap* to the range [0,1].

   b) Count the number of regional maxima in *Dmap* for each labeled connected component; let this number be *N*.

   c) Guess an initial parameter value $h = 0.01$.

   d) While $N > 1$, successive calculations of the *HMAX* transform incrementing *h* in small steps (experimentally set to 0.05) begins until the calculated number *N* of regional maxima repeats its value a number of times, reaching a count heuristically set to five, or *N* reaches the value 1.

      This was the criterion of convergence for the calculation of the number of maxima and the suppression of spurious extrema. Here in each iteration the new value of *N* is saved and compared with the previous one, to allow counting the number of repeats of it. Every time *N* changes, the counter is reset to one and counting re-starts.

```
Input: Ibw: binary image obtained from
       initial coarse segmentation
Output: Iseg: binary image with split
        cell clusters
dh ← $0.005$ ;                          /* initialize dh */
Iseg ← create a matrix of zeros of size of
  $Ibw$;
$L$ ← set of all connected components of $Ibw$;
foreach $c \in L$ do
    cont ← $1$ ;                        /* initialize cont */
    $h \leftarrow 0.01$ ;                      /* initialize h */
    $cc$ ← complement($c$) ;
    DMap ← DT($cc$) ;  /* distance transform */
    DMapn ← Normalize(Dmap) ;
      /* normalize Dmap to the range $[0,1]$ */
    RegMax ← RegionalMaxima(DMapn) ;
    $N$ ← # connected components of
      RegMax;
    Nprev ← $N$ ;
    while $N > 1$ and cont $\leq 5$ do
        Hmap ← H-Maxima(DMapn, $h$);
        RegMax ← RegionalMaxima(Hmap);
        $N$ ← # connected components of
          RegMax;
        $h \leftarrow h + dh$ ;
        if $N$ = Nprev then
            cont ← cont +1;
        else
            cont ← 1 ;
            Nprev ← $N$ ;
    if $N > 1$ then  /* c is a nucleus cluster */
        $S$ ←
          SplitClusterWEDT($c$, Hmap, RegMax);

        Iseg ← Iseg $\vee$ $S$;
    else                 /* c is an isolated cell */
        Iseg ← Iseg $\vee$ $c$;
return Iseg;
```

   e) If $N > 1$ after convergence, the labeled binary object is classified as a cluster and the algorithm, as will be seen, calls the method *SplitClusterWEDT* in order to split it. This function would receive two additional parameters, these are the final calculation of the *H*-maxima transform, which contains the information about the heights of its regional maxima, as well as the regional maxima map, stored respectively in *Hmap* and *RegMax*, that were obtained in step (d).

The pseudo code illustrates this algorithm for the Iterative *H*-maxima transform method. Fig. 4 shows a binary object corresponding to a cluster of 2 erythrocytes and the regional maxima obtained for it during its processing. Notice that in this case $N = 9$ initially and at the end of the algorithm run $N = 2$ as it should be.

2.4.2 Method 2: Morphological Filtering

This method applies a morphological approach to detect clusters and extracting markers for both the clusters and the single cells. The steps are as follows:

1. Perform the coarse segmentation of the original image in the same way as in Method 1.

2. Determine the distance transform (Euclidean) of the complement of this binary image and then normalize it. Let be *Idt* the resulting image.

3. Compute a two-stages open-close alternating sequential filtering (ASF), using a disk structuring element *g* with radius 1 and 2 in the first and second filtering stages respectively, in order to eliminate the spurious maxima. We call the resulting image *Ioc*. The general expression for this filtering process is:

$$\mathrm{ASF}^2_{CO,g}(f) = ((((f \circ g) \bullet g) \circ 2g) \bullet 2g). \quad (4)$$

   For which in this case *f* is the *Idt* image. Here ∘ and • mean respectively morphological opening and closing.

4. Determine the regional maxima on *Ioc* and call the resulting image *Irm*.

5. For each labeled connected component present in the binary image:

   a) Compute a logical AND operation between the binary image of the connected component and *Irm*. We call the resulting image *Imark*.

   b) Count the numbers of regional maxima on *Imark* with the aid of labeling the connected components contained in it.

   c) If the number calculated in (b) is greater than one the object is classified as a cluster and its division is carried out using the *SplitClusterWEDT* method, which receive as arguments the binary image of the cluster, the regional maxima map of the cluster (*Imark*) and the distance transform image after the open close filtering (*Ioc*). In other cases, the object pertaining this connected component is classified as a single erythrocyte.

2.4.3 Method 3: Radon Transform (*RT*)

This method uses the Radon transform to find the markers for the cells as described in [11]. The search for markers is performed based on the ability of the *RT* to detect shape parameters and their behavior with circular structures.

The circular structure edge was determined previously in order to apply the direct *RT* and after that the sinogram projections were filtered using a matched filter having a horseshoe-shaped impulse response. This filter was used to enhance the projections of all circular structures with radius *r*, which is computed from the median cell area in each image.

Then, an image with peaks close to the circular structures centers is obtained by means of the inverse *RT*. After this, a threshold is applied which is calculated by means of histogram analysis of the reconstructed grayscale image.

Finally, a morphological closing was performed in order to identify the final markers of each cell. Once the image containing the markers is obtained, we proceed to determine which connected components within the coarse segmentation image can be considered as clusters for their subsequent division by means of the *SplitClusterWEDT* method.

Similarly, to the previous method, for each connected component of the binary image obtained by means of the coarse segmentation, a logical AND operation of it with the whole markers image is performed to obtain the final markers that correspond to the specific connected component that is being analyzed. The resulting markers are labeled and if their number is greater than one the corresponding object is considered as a cluster.

The *SplitClusterWEDT* algorithm needs three arguments, which in this case are the binary image of the cluster, the markers corresponding to this cluster and the normalized distance transform of the logical complement of the cluster binary image.

As a final comment concerning the last step in the previous descriptions, e.g. calling the method to split the clusters, we emphasize the fact that aside from the cited *SplitClusterWEDT* method, splitting by means of the classical marker controlled watershed transform as well as using the *EDT* were also tested and compared, as described in the following section.

2.5 Segmentation of Clusters Into Their Constituent Objects

The algorithm devoted to segment the connected components identified as clusters into their constituent parts takes three inputs. The first one *C* is the binary image of the cluster. The second parameter *RegMax* is the binary image of the valid regional maxima identified during cluster *C* detection.

The third one *Hmap* depends upon the clusters detection method employed. The output of this algorithm is the binary image *Cseg* of the cluster, divided into its constituent components.

The algorithm begins by labeling and counting the connected components of the regional maxima contained in *RegMax* and setting their values in the variables *LRM* and *Num* respectively. Then follows a loop having as many iterations as regional maxima are present in *C*.

This loop starts initializing a binary matrix *S* to zero and then setting to one the elements of *S* whose positions match with the elements labeled *i* in the *LRM* matrix. The described loop can be implemented instead through vector operations for the sake of computational efficiency.

Then the algorithm computes an element-wise multiplication (Hadamard product) between matrices *S* and *Hmap* to obtain a new matrix called *HeightRM*, whose values correspond, for Method 1, to those of the *h*-maxima transform in the region occupied by each regional maximum and are zero in the rest of the matrix. In this case, for methods 2 and 3 the values of the distance transform are used instead of the *h*-maxima transform values.

```
Input: $C_{m\times n}$: binary mask of the cluster;
       $\mathrm{RegMax}_{m\times n}$: inner markers;
       $\mathrm{HMap}_{m\times n}$: height of the maxima and
       depend of the cluster detection
       method selected
Output: $Cseg_{m\times n}$: split cluster
LRM ← label matrix for connected
  components of RegMax;
num ← # of connected components of
  RegMax;
dtarray ← array of m × n × num
  dimensions;
for $i \leftarrow 1$ to num do
    $S \leftarrow$ matrix of size $m \times n$ initially set to
      zero;
    $S_{j,k} \leftarrow 1$ for all $j \leftarrow 1 : m$, $k \leftarrow 1 : n$,
      such that $\mathrm{LRM}_{j,k} = i$ ;
    $\mathrm{HeightRM} \leftarrow S \circ \mathrm{Hmap}$ ;      /* Hadamard
      product */
    $\mathrm{index} \leftarrow \mathrm{find}(\mathrm{HeightRM}, 1)$ ;   /* index of
      the first non-zero element in HeightRM */
    $\mathrm{divfact} \leftarrow \mathrm{HeightRM}(\mathrm{index})$;
    $\mathrm{dtarray}^i \leftarrow \frac{DT(S)}{\mathrm{divfact}}$ ; /* obtain the WEDT for
      the ith regional maximum */
for $j \leftarrow 1 : m, k \leftarrow 1 : n$ do
    // minimum of all matrices
    $\mathrm{im4ws}_{j,k} \leftarrow min\{\mathrm{dtarray}^i_{j,k}, i \leftarrow 1 :
      num\}$;
$\mathrm{SKIZ} \leftarrow \mathrm{watershed}(\mathrm{im4ws})$ ;
$\mathrm{Cseg} \leftarrow C$ ;
$\mathrm{Cseg}_{j,k} \leftarrow 0$ for all $j \leftarrow 1 : m$, $k \leftarrow 1 : n$,
  such that $\mathrm{SKIZ}_{j,k} = 0$ ;
return Cseg;
```

Then for each regional maximum its height value called *divfact*, is used to weight (divide) the *EDT* value associated to this maximum.

A three-dimensional array called *dtarray* is built in which its *i*th level is a matrix that contains the weighted *EDT* (*WEDT*), which is the *EDT* with its values divided (weighted) by *divfact*.

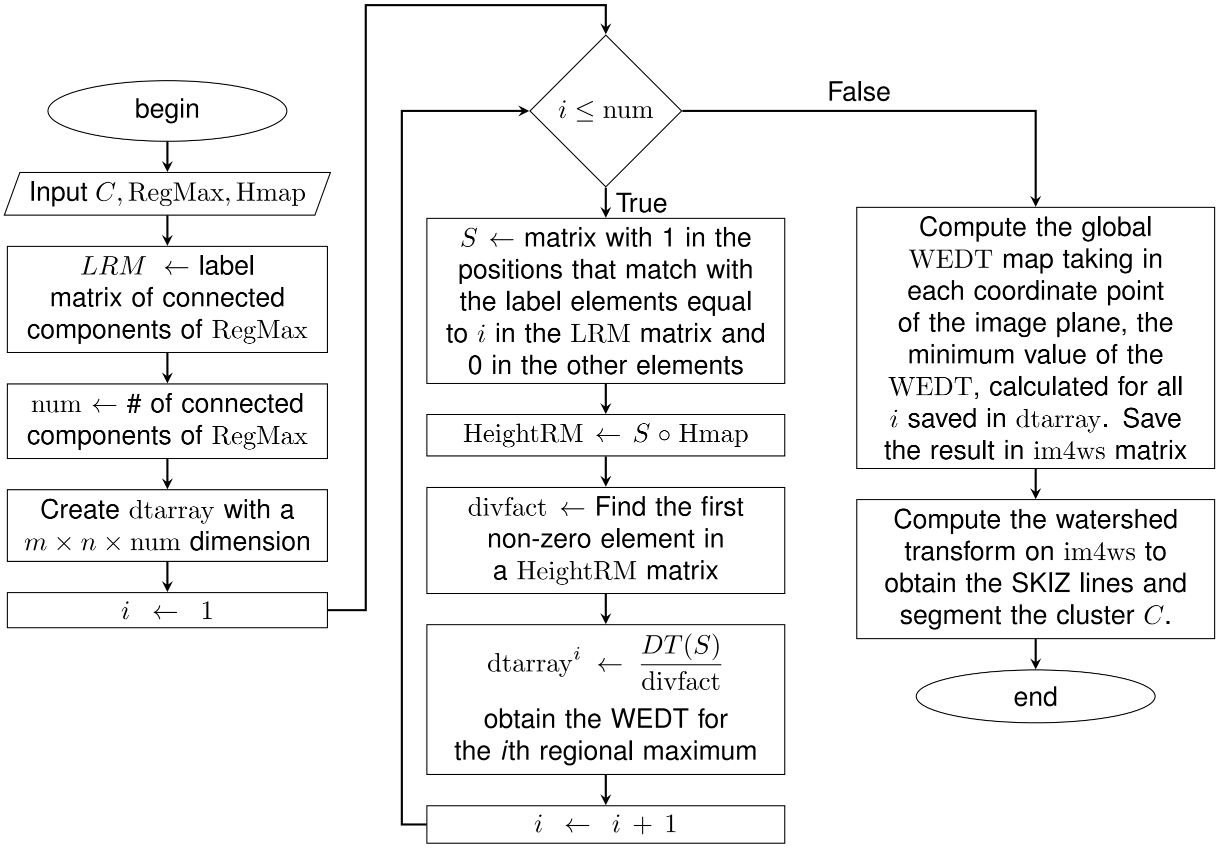

begin
Input $C, \mathrm{RegMax}, \mathrm{Hmap}$
$LRM \leftarrow$ label matrix of connected components of RegMax
$\mathrm{num} \leftarrow$ # of connected components of RegMax
Create $\mathrm{dtarray}$ with a $m \times n \times \mathrm{num}$ dimension
$i \leftarrow 1$
$i \leq \mathrm{num}$
True
False
$S \leftarrow$ matrix with 1 in the positions that match with the label elements equal to i in the LRM matrix and 0 in the other elements
$\mathrm{HeightRM} \leftarrow S \circ \mathrm{Hmap}$
$\mathrm{divfact} \leftarrow$ Find the first non-zero element in a $\mathrm{HeightRM}$ matrix
$\mathrm{dtarray}^i \leftarrow \frac{DT(S)}{\mathrm{divfact}}$ obtain the WEDT for the *i*th regional maximum
$i \leftarrow i+1$
Compute the global WEDT map taking in each coordinate point of the image plane, the minimum value of the WEDT, calculated for all i saved in $\mathrm{dtarray}$. Save the result in $\mathrm{im4ws}$ matrix
Compute the watershed transform on $\mathrm{im4ws}$ to obtain the SKIZ lines and segment the cluster C.
end

Fig. 5. Block diagram of the algorithm to segment the clusters using the Weighted External Distance Transform

The reasoning behind this procedure is that the *WEDT* value calculated in some specific point tends to be lower for a larger height of the maximum and viceversa.

This fact determines that the *SKIZ* lines tend to separate from higher maxima and come closer to lower maxima, and this leads to a better location of the *SKIZ* lines (equal distance) to segment clustered objects having different size.

The remaining *i* values give rise to matrices corresponding to each regional maximum, each one of them with its respective weight. The algorithm saves in dtarray the *WEDT* for each regional maximum in a cluster.

Then it computes the global *WEDT* map taking in each coordinate point of the image plane, the minimum value of the *WEDT*, calculated for all i saved in *dtarray* and saving it in *im4ws* matrix.

Then the marker controlled watershed transform is applied to this matrix to obtain the *SKIZ* lines which will be used to segment the binary cluster *C*.

Fig. 5 shows a block diagram illustrating the described algorithm, the pseudo code for it is shown above. Two alternative methods were compared with the proposed algorithm in other to explore their accuracy.

These methods were the marker controlled watershed transform using the inner distance transform (*CW*) and the marker controlled watershed transform using the external distance transform (*EDT*).

The combination of the three methods implemented for the detection of clusters (Iterative *H*-maxima transform, Morphological filtering and Radon transform) with the three methods to split them into their constituent objects form nine combined methods.

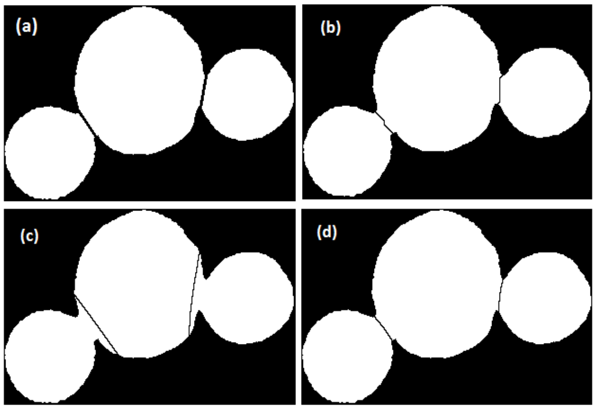


Fig. 6. Watershed lines in the segmentation result after detecting the clusters by means of iterative *H*-maxima algorithm. (a) Ground truth. (b) Using inner distance transform. (c) Using external distance transform. (d) Using the weighted external distance transform

Fig. 6 shows the result of the segmentation using the Iterative *H*-maxima method to detect markers and the three ways to split the cluster: the inner distance transform, the external distance transform and the proposed weighted external distance transform.

In this figure, we can notice the difference in terms of the watershed lines. In (a) the ground truth lines (b) broken lines can be observed, in (c) the line is somewhat displaced from the right position and in (d) the splitting line appears in a right place.

2.6 Evaluating the Effectiveness of Clusters Detection

A comparison between the three methods to detect clusters allowed determining the most appropriate alternative.

This comparison considered the detection of clusters in terms of true positives (TP) or clusters classified as such, false positives (FP) single objects classified as clusters, true negatives (TN) single objects correctly classified, and false negatives (FN) as clusters classified as single objects. From these data, the indexes of effectiveness: sensitivity, specificity, accuracy, F-measure and precision were calculated.

These measures are defined as follows:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{5}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{6}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{7}$$

$$\text{F-measure} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}}, \tag{8}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{9}$$

2.7 Evaluating the Segmentation Accuracy

The segmentation accuracy was tested using a ground truth composed by 500 binary clusters obtained from a first coarse segmentation, from which a careful, manually segmented version was built by digitally drawing an appropriate straight line between the vertices of the concavities that appear just at the points where the overlapping region of the roundish erythrocytes begin, as shown in Fig. 6a.

These clusters comprised two to eight single touching or overlapping objects with low to moderately different shapes, sizes and spatial orientations, up to 1220 single objects. The metric used to evaluate the accuracy of the segmentation was the Jaccard similarity index [10], which measures the coincidence between the segmentation result and the ground truth and is defined as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \quad 0 \leq J \leq 1, \tag{10}$$

where *A* and *B* are the binary sets to be compared and $| * |$ means the cardinality of sets. *A* result $J = 1$ means perfect coincidence between the binary images while $J = 0$ indicates total lack of coincidence. In our case, *A* would be the manually segmented object and *B* the object obtained from the automated segmentation method.

Table 1. Indexes of effectiveness in the detection of clusters

Indexes	Iterative Hmax	Morph Filtering	Radon Transform
TP	1057	1068	945
TN	3182	3181	3187
FP	2	3	0
FN	24	13	133
Sensitivity	97.78%	98.8%	87%
Specificity	99.94%	99.91%	100%
F-measure	98.79%	99.26%	93.43%
Accuracy	99.39%	99.62%	96.88%
Precision	99.81%	98.72%	100%

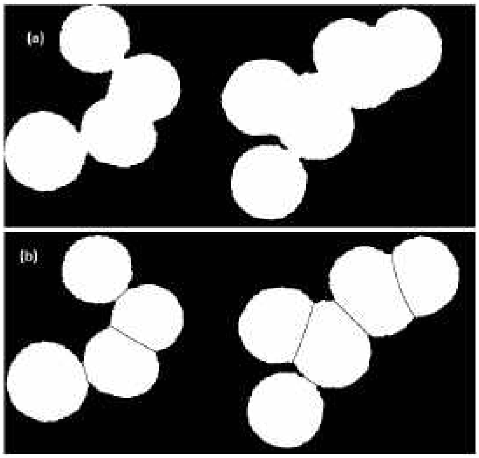


Fig. 7. (a) Binary image with two clusters, (b) Segmentation result using the *HmaxWEDT* method

The analysis and interpretation of the results when evaluating the Jaccard coefficients was performed applying statistical tests.

We compared the nine methods using the Friedman's non-parametric rank test with a Bergmann and Hommel's correction for the post-hoc analysis. These tests were computed using the public R scmamp package [5].

3 Results and Discussion

Fig.7 shows two segmentation results using the *HmaxWEDT* method. Here the Iterative *H*-maxima transform is used to detect clusters and extract the inner markers and the weighted external distance transform (*WEDT*) to split the clusters into their constituent objects.

We stress the fact that this combination of methods obtained the best results. The effectiveness in the detection of clusters was measured in terms of sensitivity and specificity. We analyzed 43 images containing 4265 binary objects. Table 1 shows the indexes of effectiveness in the detection of clusters, for the three methods analyzed: Iterative *H*-maxima transform, Morphological filtering and Radon transform.

The numbers in the tables were rounded to two decimal places. Table 2 shows the descriptive statistics of the Jaccard coefficients calculated for the nine methods analyzed, for which 1220 objects were used.

Here *Hmax*, *Morph* and *Radon* stand for the Iterative *H*-maxima transform, Morphological filtering and Radon transform respectively, and *CW*, *EDT* and *WEDT* for the classical watershed transform, external distance transform and weighted external distance transform.

This table shows that the method *HmaxWEDT* exhibited better results than the others, in terms of mean, median and standard deviation. Similar results were obtained with the other methods when using the *WEDT*.

The Friedman test found statistically significant differences in results among the compared algorithms with a *p-value* of 2.2e-16 (test statistic = 6234.5). Then, the Bergmann and Hommel *post-hoc* procedure was carried out in order to find which combination of methods showed a statistically significant difference.

As a further description in order to have a better understanding of the possible similarities and differences among the tested algorithms, we plotted and show in Fig. 8 a critical difference plot with the corrected *p-value* and $\alpha = 0.05$.

In this plot, each algorithm is placed on an axis according to its average ranking. Then, those algorithms that do not show significant differences are grouped together using a horizontal line. The rankings in the plot assume that larger values have a poorer rank.

In our case, the plot shows that, in general, the *HmaxWEDT* combination method ranked

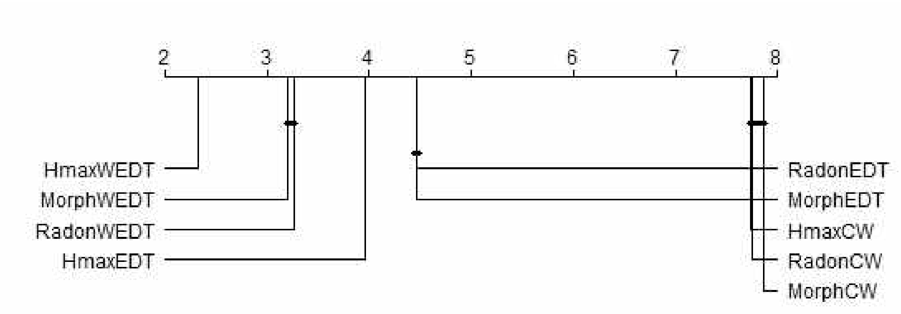


Fig. 8. Cross-comparison for the nine algorithms tested using the Friedman test and the Bergmann and Hommel *post hoc* correction. Groups of methods that are not significantly different appear connected by a horizontal line

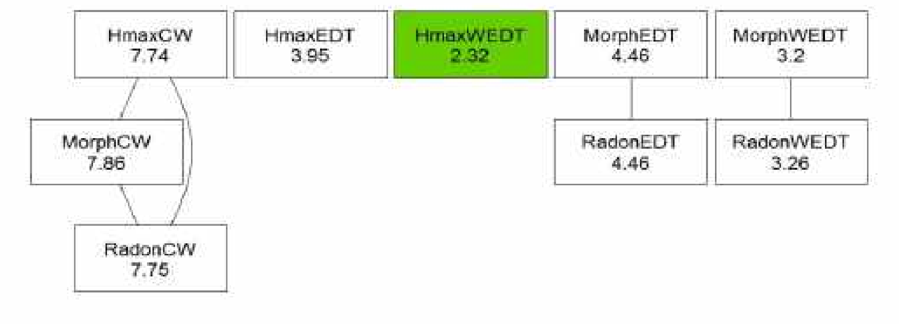


Fig. 9. Friedman test with the Bergmann and Hommel *post hoc* correction for the nine algorithms tested. Groups of methods that are not significantly different appear as connected nodes

significantly better than the other combined algorithms, showing as well statistically significant differences in comparison with the others.

Another representation of the results of this test is shown in Fig. 9, where in this graph each node represents an algorithm and shows its name and the computed Friedman's test statistic.

A node with a filled background in green indicates the best ranked algorithm after this comparison. Lines between nodes indicate that the differences between connected algorithms are not found to be significant for $\alpha = 0.05$, according to the Bergmann-Hommel *post-hoc* procedure.

There are no significant differences between the algorithms *HmaxCW*, *MorphCW* and *RadonCW* for which their mean ranks are very similar.

These three algorithms in spite of the way they use to detect the markers for the objects -using the methods Iterative *H*-maxima transform, morphological approach and Radon transform respectively- have in common the way used to split the clusters, e.g. using the classical watershed transform.

The same occurs with pairs *MorphEDT* and *RadonEDT* which use the external distance transform; and *MorphWEDT* and *RadonWEDT* which use the proposed weighted external distance transform.

3.1 Comparative Study

In order to make a comparative assessment of our proposed method, experimental results were compared to other state-of-art methods cited in the present article.

In spite that these works do not use the same database or even the same type of cells, the global results may provide an idea about how the figures obtained in the experiments reported in this article compare with those obtained in other works in this field.

In reference [18] the results are expressed in terms of percentages of correctly segmented

Table 2. Descriptive statistics of the Jaccard coefficient for the nine methods

Method	Mean	Median	St.Dev.	Max	Min
HmaxCW	0.946	0.948	0.01	0.965	0.853
HmaxEDT	0.985	0.991	0.021	1	0.749
HmaxWEDT	**0.993**	**0.996**	0.01	1	**0.892**
MorphCW	0.94	0.948	0.057	0.965	0.332
MorphEDT	0.977	0.991	0.055	1	0.157
MorphWEDT	**0.987**	**0.994**	0.042	1	0.393
RadonCW	0.946	0.948	0.017	0.965	0.561
RadonEDT	0.98	0.99	0.04	1	0.258
RadonWEDT	**0.987**	**0.994**	0.034	1	0.408

Table 3. Detection of cell clusters for one image

Method	CC	TP	TN	FP	FN
Iterative *H*-maxima	117	27	89	0	1
Morph filtering	117	27	89	0	1
Radon transform	117	23	89	23	5

Table 4. Mean runtime for each method (in seconds)

Method	Radon	Morph Filtering	Iterative *H*-maxima
Classical Watershed transform (CW)	8.66	1.82	46.41
External distance transform (EDT)	8.59	2.11	46.86
Weighted external distance transform (WEDT)	23.11	18.87	62.73

clusters obtaining a 96.43% accuracy on cervical and breast cancer images.

The accuracy results obtained in our works are higher compared with this reference in spite that the image are from different types of cells. Reference [20] showed their results in terms of performance measures of overlapped cells detection as well as accuracy of splitting.

They achieved 97.4% accuracy in the overlapped cells detection on the test set. In our work, we obtain better results in the cluster detection process achieving 99.39% and 99.69% accuracy with the Iterative *H*-maxima transform and Morphological filtering methods respectively.

Reference [44] obtained high results in terms of sensitivity, precision and F-measure where true positive (TP) is the number of correctly split objects. Three datasets were used to evaluate the performance of the method and average values of sensitivity = 98.29%, precision = 99.02% and F-measure = 98.65% were obtained.

In our work, the TP is the number of objects classified correctly as clusters, and in this sense, we obtained 99.81% precision and 98.79% F-measure by the Iterative *H*-maxima transform method as well as 99.72% precision and 99.26% F-measure by the Morphological filtering method which are slightly better.

3.2 Runtime Analysis

This study was carried out using MATLAB (2016a version) on a computer with an Intel Core i3-2310M processor clocked at 2.10 GHz and with 4 GB of RAM and 64 bits Windows 10 Pro operating system. To reduce the computational load, the binary image obtained from the coarse segmentation was resized to resolutions of 1024 × 768 pixels.

Table 3 shows for one resized binary image the total of connected components (CC) and the indexes of TP, TN, FP and FN detected by the three methods. For this image, the Iterative *H*-maxima transform and Morphological filtering obtained the same results.

These two methods exhibited the best results obtaining the TP and consequently the best results in terms of sensitivity, F-measure and accuracy showed earlier in Table 1.

Table 4 shows a comparison of the mean running times for the three clusters detection algorithms combined with the three methods used to split the clusters in their constituent parts.

The morphological filtering method combined with the three cluster-splitting methods showed the best performance in terms of speed, which is a very important factor when analyzing large numbers of images.

The Iterative *H*-maxima method was the most time consuming. This result is a consequence of the need to perform a number of iterations calculating the *H*-maxima transform, which has a relatively high computational cost, in order to obtain the appropriate *h* values.

The same occurs with the *WEDT* method used to split the clusters. In this case, each detected cluster is to be analyzed to compute the weighted distance transform, which has a higher computational cost.

4 Conclusion

This research explored various alternatives to detect and split connected components in binary images, which appear in segmentation processes of microscopy images having touching or overlapping erythrocytes. The scope of this approach was constrained to blood smear images containing erythrocytes having moderate differences in size as well as a moderate degree of overlapping.

Three methods to detect connected components associated to clusters, named Iterative *H*-maxima transform, Morphological filtering and Radon transform were used, as well as three methods to split these connected components in their constituent parts, named in this case external distance transform (*EDT*), the classical watershed transform (*CW*) and the weighted distance transform (*WEDT*) which result in nine possible combinations.

References

1. **Al-Kofahi, Y., Lassoued, W., Lee, W., Roysam, B. (2010).** Improved automatic detection and segmentation of cell nuclei in histopathology images. IEEE Transactions on Biomedical Engineering, Vol. 57, No. 4, pp. 841–852. DOI: 10.1109/TBME.2009.2035102.

2. **Al-Kofahi, Y., Zaltsman, A., Graves, R., Marshall, W., Mirabela, R. (2018).** A deep learning-based algorithm for 2-D cell segmentation in microscopy images. BMC Bioinformatics, Vol. 19, No. 365. DOI: 10.1186/s12859-018-2375-z.

3. **Brosnan, T., Sun, D. W. (2002).** Inspection and grading of agricultural and food products by computer vision systems—a review. Computers and Electronics in Agriculture, Vol. 36, No. 2-3, pp. 193 – 213. DOI: 10.1016/S0168-1699(02)00101-1.

4. **Cabello, E., Sánchez, M., Delgado, J. (2002).** A New Approach to Identify Big Rocks with Applications to the Mining Industry. Real-Time Imaging, Vol. 8, No. 1, pp. 1–9. DOI: 10.1006/rtim.2000.0255.

5. **Calvo, B., Santafé Rodrigo, G. (2016).** scmamp: Statistical comparison of multiple algorithms in multiple problems. The R Journal, Vol. 8, No. 1.

6. **Chinea-Valdés, L., Lorenzo-Ginori, J. (2011).** Evaluation of distance transform based alternatives for image segmentation of overlapping objects. Scientific Conference on Computer Science and Informatics.

7. **Diaz, G. (2008).** Automatic clump splitting for cell quantification in microscopical images. In Progress in Pattern Recognition, Image Analysis and Applications Lecture Notes in Computer Science, Vol. 4756. pp. 763–772.

8. **Dorfer, M., Mattes, J. (2016).** Recursive water flow: A shape decomposition approach for cell clump splitting. IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 811–815. DOI: 10.1109/ISBI.2016.7493390.

9. **Dougherty, E. R., Lotufo, R. A. (2003).** Hands-on Morphological Image Processing. SPIE. DOI: 10.1117/3.501104.

10. **Ge, F., Wang, S., Liu, T. (2007).** New benchmark for image segmentation evaluation. Journal of Electronic Imaging, Vol. 16, No. 3. DOI: 10.1117/1.2762250.

11. **González-Betancourt, A., Rodríguez Ribalta, P., Meneses Marcel, A., Sifontes Rodríguez, S., Lorenzo Ginori, J. V., Orozco Morales, R. (2016).** Automated marker identification using the Radon transform for watershed segmentation. IET Image Processing, Vol. 11, No. 3, pp. 183–189.

12. **Haas, T., Schubert, C., Eickhoff, M., Pfeifer, H. (2020).** BubCNN: Bubble detection using faster RCNN and shape regression network. Chemical Engineering Science, Vol. 216. DOI: 10.1016/j.ces.2019.115467.

13. **Hatipoglu, N., Bilgin, G. (2017).** Cell segmentation in histopathological images with deep learning algorithms by utilizing spatial relationships. Medical; Biological Engineering; Computing, Vol. 55, No. 10, pp. 1829–1848. DOI: 10.1007/s11517-017-1630-1.

14. **Ibtehaz, N., Rahman, M. S. (2020).** MultiResUNet : Rethinking the u-net architecture for multimodal biomedical image segmentation. Neural Networks, Vol. 121, pp. 74–87. DOI: 10.1016/j.neunet.2019.08.025.

15. **Indhumathi, C., Cai, Y. Y., Guan, Y. Q., Opas, M. (2011).** An automatic segmentation algorithm for 3D cell cluster splitting using volumetric confocal images. Journal of Microscopy, Vol. 243, No. 1, pp. 60–76. DOI: 10.1111/j.1365-2818.2010.0382.x.

16. **Jie, D., Jing-feng, L., Jing-yu, Y. (2010).** Combined technologies in analysis of overlapping cells. 2010 International Conference on E-Business and E-Government, IEEE, pp. 1608–1612. DOI: 10.1109/icee.2010.407.

17. **Jierong, C., Rajapakse, J. C. (2009).** Segmentation of Clustered Nuclei With Shape Markers and Marking Function. Biomedical Engineering, IEEE Transactions on, Vol. 56, No. 3, pp. 741–748.

18. **Jung, C., Kim, C. (2010).** Segmenting clustered nuclei using H-minima transform-based marker extraction and contour parameterization. Biomedical Engineering, IEEE Transactions on, Vol. 57, No. 10, pp. 2600–2604.

19. **Jung, C., Kim, C. (2014).** Impact of the accuracy of automatic segmentation of cell nuclei clusters on classification of thyroid follicular lesions. Cytometry Part A, Vol. 85, No. 8, pp. 709–718. DOI: 10.1002/cyto.a.22467.

20. **Khodadadi, V., Fatemizadeh, E., Setarehdan, S. K. (2015).** Overlapped cells separation algorithm based on morphological system using distance minimums in microscopic images. 2015 22nd Iranian Conference on Biomedical Engineering (ICBME), IEEE, pp. 263–268.

21. **Koh, T., Miles, N., Morgan, S., Hayes-Gill, B. (2007).** Image segmentation of overlapping particles in automatic size analysis using multi-flash imaging. IEEE Workshop on Applications of Computer Vision (WAC V '07), IEEE. DOI: 10.1109/wacv.2007.37.

22. **Koyuncu, C. F., Akhan, E., Ersahin, T., Cetin-Atalay, R., Gunduz-Demir, C. (2016).** Iterative h-minima-based marker-controlled watershed for cell nucleus segmentation. Cytometry Part A, Vol. 89, No. 4, pp. 338–349. DOI: 10.1002/cyto.a.22824.

23. **Li, G., Liu, T., Nie, J., Guo, L., Chen, J., Zhu, J., Xia, W., mara, A., Holley, S., Wong, S. (2008).** Segmentation of touching cell nuclei using gradient flow tracking. Wiley, Vol. 231, No. 1, pp. 47–58. DOI: 10.1111/j.1365-2818.2008.02016.x.

24. **Litjens, G., Kooi, T., Bejnordi, B. E., Setio Adiyoso, A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., Sánchez, C. I. (2017).** A survey on deep learning in medical image analysis. Medical Image Analysis, Vol. 42, pp. 60–88. DOI: 10.1016/j.media.2017.07.005.

25. **Loddo, A., Ruberto, C. D., Kocher, M. (2018).** Recent advances of malaria parasites detection systems based on mathematical morphology. Sensors, Vol. 18, No. 2, pp. 513. DOI: 10.3390/s18020513.

26. **Nasr-Isfahani, S., Mirsafian, A., Masoudi-Nejad, A. (2008).** A New Approach for Touching Cells Segmentation. Vol. 1, pp. 816–820.

27. **Neves, J. C., Castro, H., Tomás, A., Coimbra, M., Proença, H. (2014).** Detection and separation of overlapping cells based on contour concavity for Leishmania images. Cytometry Part A, Vol. 85, No. 6, pp. 491–500.

28. **Nguyen, N. T., Duong, A. D., Vu, H. Q. (2011).** Cell splitting with high degree of

overlapping in peripheral blood smear. International Journal of Computer Theory and Engineering, pp. 473–478. DOI: 10.7763/ijcte.2011.v3.352.

29. **Panagiotakis, C., Argyros, A. (2020).** Region-based fitting of overlapping ellipses and its application to cells segmentation. Image and Vision Computing, Vol. 93. DOI: 10.1016/j.imavis.2019.09.001.

30. **Park, C., Huang, J. Z., Ji, J. X., Ding, Y. (2013).** Segmentation, inference and classification of partially overlapping nanoparticles. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 3. DOI: 10.1109/tpami.2012.163.

31. **Phoulady, H. A., Goldgof, D. B., Hall, L. O., Mouton, P. R. (2016).** A new approach to detect and segment overlapping cells in multi-layer cervical cell volume images. IEEE 13th International Symposium on Biomedical Imaging (ISBI), pp. 201–204. DOI: 10.1109/isbi.2016.7493244.

32. **Plissiti, M. E., Nikou, C. (2012).** Overlapping cell nuclei segmentation using a spatially adaptive active physical model. Image Processing, IEEE Transactions on, Vol. 21, No. 11, pp. 4568–4580. DOI: 10.1109/tip.2012.2206041.

33. **Plissiti, M. E., Vrigkas, M., Nikou, C. (2015).** Segmentation of cell clusters in Pap smear images using intensity variation between superpixels. International Conference on Systems, Signals and Image Processing (IWSSIP), IEEE, pp. 184–187.

34. **Romero Rondón, M. F., Sanabria Rosas, L. M., Bautista Rozo, L. X., Mendoza Castellanos, A. (2016).** Algoritmo para la detección de glóbulos rojos superpuestos en imágenes microscópicas de extendidos de sangre periférica. DYNA, Vol. 83, No. 198, pp. 187–194. DOI: 10.15446/dyna.v83n198.47177.

35. **Ronneberger, O., Fischer, P., Brox, T. (2015).** U-Net: Convolutional networks for biomedical image segmentation. Lecture Notes in Computer Science, Springer International Publishing, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

36. **Savelonas, M., Maroulis, D., Mylona, E. (2009).** Segmentation of two-dimensional gel electrophoresis images containing overlapping spots. 9th International Conference on Information Technology and Applications in Biomedicine, pp. 1–4.

37. **Schmitt, O., Hasse, M. (2009).** Morphological multiscale decomposition of connected regions with emphasis on cell clusters. Computer Vision and Image Understanding, Vol. 113, No. 2, pp. 188–201. DOI: 10.1016/j.cviu.2008.08.011.

38. **Song, Y., Tan, E. L., Jiang, X., Cheng, J. Z., Ni, D., Chen, S., Lei, B., Wang, T. (2017).** Accurate cervical cell segmentation from overlapping clumps in pap smear images. IEEE Transactions on Medical Imaging, Vol. 36, No. 1, pp. 288–300. DOI: 10.1109/tmi.2016.2606380.

39. **Wang, W. (2008).** Rock particle image segmentation and systems. Pattern Recognition Techniques, Technology and Applications. I-Tech, Vienna, Austria, pp. 197–226. DOI: 10.5772/6242.

40. **Wenzhong, Y., Xiaohui, F. (2010).** A watershed based segmentation method for overlapping chromosome images. 2010 Second International Workshop on Education Technology and Computer Science, IEEE. DOI: 10.1109/etcs.2010.107.

41. **Xu, W., Sang, N. (2015).** Urine sediment overlapped cells segmentation based on hough transform and geometrical feature. International Symposium on Bioelectronics and Bioinformatics (ISBB), IEEE, pp. 211–214.

42. **Yang, H., Ahuja, N. (2014).** Automatic segmentation of granular objects in images: Combining local density clustering and gradient-barrier watershed. Pattern Recognition, Vol. 47, No. 6, pp. 2266–2279. DOI: 10.1016/j.patcog.2013.11.004.

43. **Zafari, S., Eerola, T., Sampo, J., Kälviäinen, H., Haario, H. (2015).** Segmentation of partially overlapping nanoparticles using concave points. Advances in Visual Computing, Springer International Publishing, pp. 187–197. DOI: 10.1007/978-3-319-27857-5_17.

44. **Zhang, Q., Wang, J., Liu, Z., Zhang, D. (2020).** A structure-aware splitting framework for separating cell clumps in biomedical images. Signal Processing, Vol. 168. DOI: 10. 1016/j.sigpro.2019.107331.

45. **Zhang, W. H., Jiang, X., Liu, Y. M. (2012).** A method for recognizing overlapping elliptical bubbles in bubble image. Pattern Recognition Letters, Vol. 33, No. 12, pp. 1543–1548. DOI: 10.1016/j.patrec.2012.03.027.

Article received on 14/01/2021; accepted on 16/03/2022. Corresponding author is Lariza Portuondo Mallet.

ISSN 2007-9737

Comprehensive Survey: Approaches to Emerging Technologies Detection within Scientific Publications

Amir Yelenov[1,2], Alexandr A. Pak[1,2], Atabay A. Ziyaden[1,2],
Iskander Akhmetov[1,2], Alexander Gelbukh[3], Irina Gelbukh[4]

[1] Institute of Information and Computational Technologies,
Almaty,
Kazakhstan

[2] Kazakh-British Technical University,
Almaty,
Kazakhstan

[3] Instituto Politécnico Nacional,
Mexico City,
Mexico

[4] Independent Researcher,
Mexico City,
Mexico

{greamdesu, aa.pak83, iamdenay,
iskander.akhmetov, gelbukh,ir.gelbukh}@gmail.com

Abstract. The identification of breakthrough topics and emerging technologies has been of interest to the governments of many countries and the scientific community since the last century. This study presents the status and trend of the research field through a comprehensive review of relevant publications, a new look at the problem of defining the term "emergent technologies," defining boundaries between similar terms; and a modern baseline method on the citation prediction subtask for the discovery of emergent technologies. The outcomes of this technique have demonstrated the significance of features that characterize the preceding 1-year, 2-year, and 3-year citation counts, as well as their impact on the quality of neural network and random forest models. Our hypothesis, however, that author-specific measures may enhance prediction results was not supported. We ascribe this difficulty to the dimensionality curse. The authors examined methodological elements of research and technological development; consequently, it is important to note that, from a technical viewpoint, theoretical research is far from complete due to the vast variety of projects, outstanding challenges, research questions, and market assumptions. Finding more input characteristics to improve the quality of predictions and switching from classification to regression may also improve the precision of the suggested baseline model.

Keywords. Citation prediction, emergent technology, neural networks, scientometrics.

1 Introduction

For a number of years, governments, companies and individual scientists have been interested in tracking science and technology trends, which means the development of topics in science and technology that can significantly affect the socio economic sphere around the world.

Identifying and analyzing breakthrough themes is a time-consuming, expert-intensive process.

doi: 10.13053/CyS-26-4-4424

Automated techniques have time limits, too much or too little data, inadequate validation and bias control, and time-consuming, human-intensive validation against real behavior.

Globalizing science and technology enhances the possibility of high-performance technical solutions in varied socio economic and geographical places. This area's sponsored research has come in waves. In the United States, the NSF program of the 1960s attempted to track important developments in the R&D process.

It should be noted that breakthrough research can be searched in the scientific literature or the media, Topic Detection and Tracking program (TDT) considered the task of searching in the media. Thus, the tasks can be broken down into three main steps:

(1) segmenting the stream of recognized speech into individual stories; (2) identifying those news stories that are the first to discuss a new event occurring in the news; and (3) giving a small number of news examples about the event, finding all subsequent articles in the feed [55].

In 2011, the IARPA Foresight and Understanding from Scientific Exposition (FUSE) program was funded to "develop automated methods that assist in the systematic, continuous, and comprehensive evaluation of technical developments".

The fundamental hypothesis of the FUSE program is that real processes of technological development leave visible traces in the public scientific and patent literature.

FUSE creates a system that can (1) handle a massive, multidisciplinary, growing, noisy, and multilingual body of scientific and patent literature; (2) automatically generate and prioritize technical terms in emerging technical fields and provide compelling evidence of emergence; (3) provide this capability for literature in English and Chinese.

The relevance of this study is signaled by the fact that the Competition Act of America was passed in 2021, which explicitly mentions the identification of new and innovative areas as a specific goal [3].

Today, there are conferences and societies dedicated to exploring new and breakthrough technologies. Despite this broad interest in the issue of breakthrough technologies - a Scopus search for "breakthrough technologies" yields over 13,000 articles - identifying emerging topics in science and technology remains a challenge.

A recent review of definitions and methods [11] reports that most studies of emerging technologies are retrospective analyses of predefined areas rather than methodological studies designed to identify new technologies. For example, [48] identified nanobiotechnology as a new and important area in nanotechnology and then used bibliometric methods to characterize the structure of topics in the field.

While characterizing recent work is important and helps current participants in the technology understand its history and landscape, these types of studies cannot identify currently emerging topics of interest to funders and practitioners around the world. Few studies offered methods for identifying emerging themes, and even fewer offered a list of emerging themes from the literature.

The main challenge in identifying emergent technologies is coming from the problem of definition of the word. It is important to understand the process which leads technology from being new or innovative to emergent. Moreover, there are no widely agreed definitions for emergent technologies and/or research.

2 Defining "Emergent Technologies"

First of all, it is necessary to define the concept of emergence and breakthrough research and technology. According to the Oxford Advanced American Dictionary, the term "emergent" means "starting to exist, grow, or become known".

The Cambridge Advanced Learner's Dictionary & Thesaurus interprets this definition as "to become known, especially as a result of examining something or asking questions about it" or "to become known or develop as a result of something". The authors' definitions in the scientific literature of emergent technology have evolved throughout time.

In 1985, Harris et al. [23] mentions that molten salt reactors are an emerging technology, indicating that they have significant promise [for waste disposal]. In 2000, P.J. Cullen, in his paper about the food industry and food processing [12]

Table 1. Usage of "emergent technology" definition are cited

Year	Author	Author's definition	Intersection with our definition
1995	Ben Martin [37]	"promising research" that has the potential to yield the highest advantages	Influence potential
1999	Goldstein [21]	set of qualities that they must possess, namely, radical novelty, coherence, correlation, completeness, global or macro scale, dynamism, recognition	Radical novelty, consistency
2005	Mick P. Couper [9]	offer many opportunities to expand the way we think [of survey data collection]	Influence potential
2010	Cozzens [10]	rapid growth, novelty, untapped market potential, and high technology base	Radical novelty, relatively fast growth
2015	Rotolo [41]	relatively fast growth, radical novelty, consistency, influence potential, uncertainty in applicability	Radical novelty, relatively fast growth

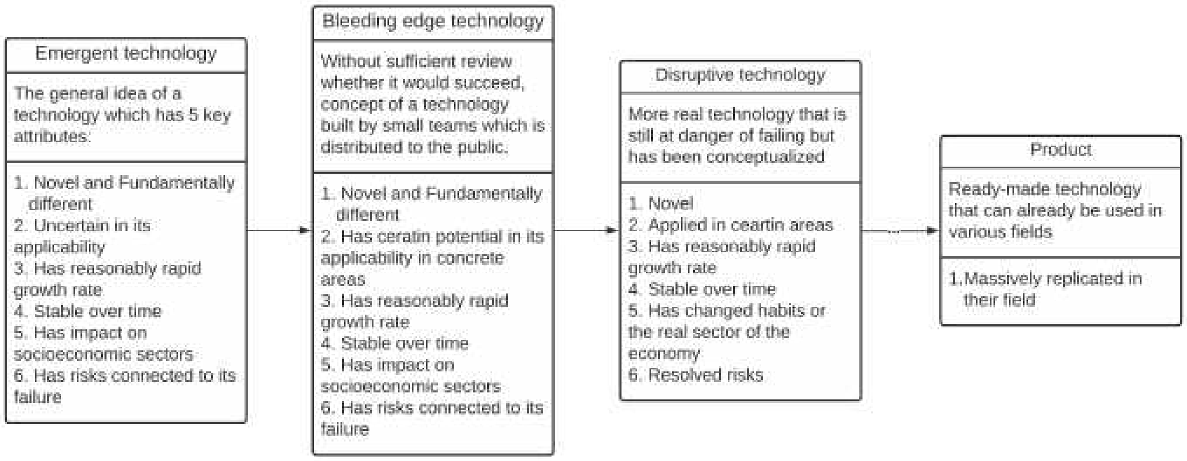

Emergent technology
The general idea of a technology which has 5 key attributes:
1. Novel and Fundamentally different
2. Uncertain in its applicability
3. Has reasonably rapid growth rate
4. Stable over time
5. Has impact on socioeconomic sectors
6. Has risks connected to its failure
Bleeding edge technology
Without sufficient review whether it would succeed, concept of a technology built by small teams which is distributed to the public.
1. Novel and Fundamentally different
2. Has ceratin potential in its applicability in concrete areas
3. Has reasonably rapid growth rate
4. Stable over time
5. Has impact on socioeconomic sectors
6. Has risks connected to its failure
Disruptive technology
More real technology that is still at danger of failing but has been conceptualized
1. Novel
2. Applied in ceartin areas
3. Has reasonably rapid growth rate
4. Stable over time
5. Has changed habits or the real sector of the economy
6. Resolved risks
...
Product
Ready-made technology that can already be used in various fields
1.Massively replicated in their field

Fig. 1. Life-cycle of a technology

has defined the term as "technology, which has potential [within the food industry]".

In the same year, Waksman, in his study [53] in the medical field, states that "The ultimate test for any emerging technology to become a standard [of care depends on the outcome of the clinical trials."

In his study of Emerging Technology for Detection of DNA Binding Proteins, Dummitt et al. (2006) state that such a technology "attempts to overcome [such] limitations" [13].

In 2011 study [40] of dienamine catalysis author defines his emergent technology in organic synthesis as "a powerful technology can address these critical issues." However, there are a number of complementary opinions, namely:

Ben Martin in his study of science and technology in 1995 [37] characterizes emerging technologies as "promising research" that has the potential to yield the highest advantages.

Mick P. Couper, in his study [9] of technology trends in 2005, explores current technological advancements in survey research and associates the term with technologies that "offer many opportunities to expand the way we think [of survey data collection]."

Emergent technologies are significant advancements in technology, such as quantum computing, artificial intelligence, robots, and additive manufacturing, that generate new competitive risks and commercial opportunities in the near and long term.

The author [21] defines emergent technologies through a set of qualities that they must possess, namely, radical novelty, coherence, correlation, completeness, global or macro scale, dynamism, recognition. Cozzens et al. [10] described emergent technologies using terms such as rapid growth, novelty, untapped market potential, and high technology base.

Another author [41] defined emergent technologies using the following key attributes: Radical novelty: the technology is significantly different from previously used methods in the field; it has grown in popularity relatively quickly to a certain extent; it has demonstrated consistency in its application and use; it has the potential to be influential but is still uncertain in its applicability.

Based on the above discussion, we have summarized the data in Table 1. We tend to introduce our own definition. The resason behind that is disunity in the definition of this term in the scientific community, as shown above.

An emergent technology is a novel technology that is fundamentally different, has a reasonably rapid growth rate, endures over time, and has a major impact on socioeconomic sectors in both local and global markets. An emergent technology is a technology, which has following characteristics:

1. Novel and fundamentally different,
2. Uncertain in its applicability,
3. Has reasonably rapid growth rate,
4. Stable over time,
5. Has impact on socioeconomic sectors.

2.1 Difference between Emergent, Disruptive and Bleeding Edge Technologies

There are numerous definitions of new and promising technologies used in the scientific and popular science communities, and our goal in this section is to distinguish them from the definition of emergent technology, as well as to identify similarities and areas in which they are used to aid in a better understanding of the concept that lies beneath its definition.

Disruptive technology is a word that is frequently used in conjunction with developing technology. According to Lingfei Wu et al., disruptive technology is frequently found or produced by small teams of scientists, and this process entails risks that they typically incur while pursuing new ideas and prospects that may succeed in the future.

The author proposed a novel statistic dubbed the "disruption index" in his study [56]. In his work, the author summarizes technology that has been tested and thus justified the risk taken by small teams; this means that, based on the technology life-cycle depicted in Figure 1, it has already passed the earlier stages of being emergent and bleeding edge technology, which were more susceptible to failure. That is not to say that his technology is without risk; nevertheless, by looking back on previously successful technologies, the author indicates that earlier steps yielded more risk.

Additionally, the term "bleeding edge" technology refers to a form of technology that has been released to the public without having sufficient reliability evidence and may therefore be unpredictable. The risk and price associated with bleeding edge technologies are typically carried by the end user - in the majority of situations, the customer.

3 Analysis of the Research Interest Among Scientific Community

The growing interest in "emerging technology" research that has been indexed by the Web of Science database during the past two decades is seen in Figure 2. The most recent few

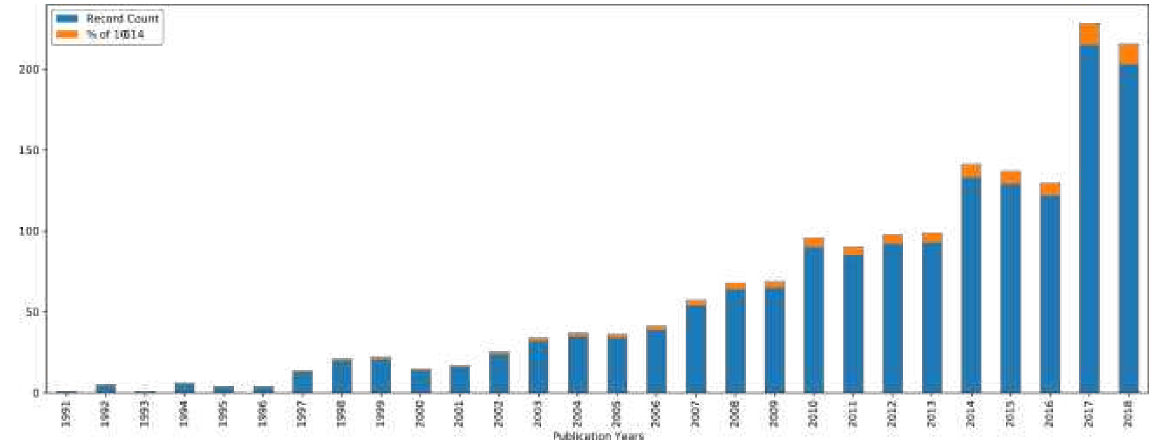


Fig. 2. Change in number of publications using the term "emergent technology" from 1991 to 2018 year according to Web Of Science

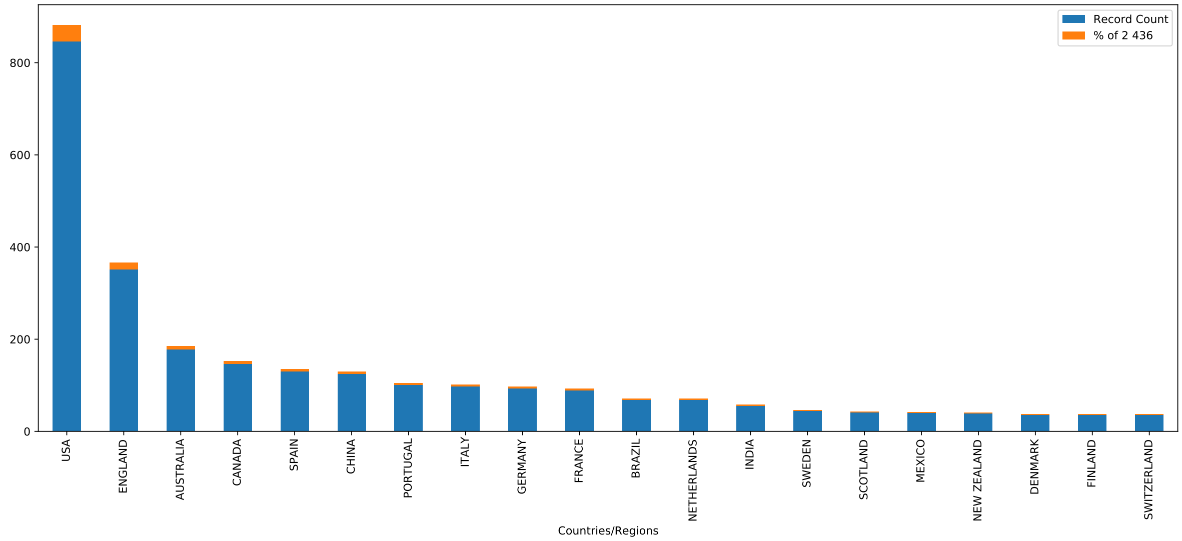


Fig. 3. Research interest among different countries on the topic of "emergent technology" as of 2018 year according to Web Of Science

years have shown very significant growth. The countries that have contributed the most to the body of knowledge about the concept of disruptive technologies are presented in Figure 3.

The data from the Web of Science indicates that the majority of articles were written by writers based in the United States. We assume that it is possible to make a connection between the rise in interest in 1997 and the implementation of initiatives such as DARPA's TDT in the United States in the late 1990s if one considers the information presented in this article.

4 Overview of Emergent Technology Detection Approaches

The authors [46] proposed a method to identify emerging topics from a broad citation database. The authors applied the proposed method and identified more than 70 topics prior to 2014 as emerging topics. These topics are characterized in terms of their key source events and drivers, applications, and various indicators.

The paper presents evidence that these themes and their key researchers are unusual in many ways. Additionally, related papers are discussed, followed by descriptions of emerging technologies, emergent themes put forward and their characteristics, the evidence associated with these themes, and a discussion of results in the context of science policy.

The methodological issue regarding methods of finding and detecting breakthrough research is the idea of searching for these topics in publicly available digital libraries of scientific literature.

The mathematical apparatus of natural language processing arises naturally here and the main flagship idea underlying trend analysis for identifying breakthrough research is the rapidly increasing number of publications and scholars working on a particular topic, which is a necessary condition for search but not yet sufficient.

Additional conditions that are imposed on candidate topics are the requirements of coherence, partial independence from their "main topic" and other disciplines, and self-sustaining, i.e., the candidate topic should not just be a subsidiary entity supported by other research, but should be an independent field.

The question of when such a field is considered "re-emerging" is not only related to its "age" but also to the time when its literature has reached the critical mass necessary for it to exist and be widely recognized as an independent and self-sustaining entity.

4.1 Identifying the Emerging Research Areas

Various approaches have been proposed to identify emerging research areas [34, 47], their level of maturity [43, 54] and their dynamics [5, 27, 51]. The first, and perhaps the most difficult task, is to segment the areas of research activity in the global research and technology environment.

In the field of scientific research, approaches have been proposed by a number of authors [5, 58]. On the other hand, in the direction of technological development, an alternative approach has been proposed [14].

There are different ways to automatically build a research activity classifier; the essential difference for any approach is only the input data. Such as grouping specialized journals, matching authors, analyzing thematic queries according to the terminology of the field, etc.

Moreover, most online services have developed their own classifiers to facilitate access to scientific information. Thus, a lot of work has been done on the surface structuring of scientific and technical activities, using a set of established keywords such as "low Earth orbit space systems", "remote sensing data", or highly cited "groundbreaking" articles to search based on terms or cited references. Sometimes all articles published in one or more journals are analyzed.

None of these approaches covers all the work published on a particular topic, but their results can be a reasonable indicator for analysis. It is worth noting that tracking the number of publications in a particular line of research involves taking into account the frequency of use of certain terms in that area of research.

However, textual similarity based on common terms is also associated with strong citation references. Thus, [26, 27] for example, analyzed the correlation between the distribution of terms and the distribution of citations in the citation graph and found that citation coherence correlated with the textual coherence of the term representing the topic.

The authors [45] tracked the evolution of clusters by selecting characteristic terms for each cluster and showed that similar topics are strongly related through cross-referencing, while articles on different topics are weakly related. They concluded that separating this subject into strongly related clusters is necessary to discover breakthrough research.

They also introduced two topological measures to determine the role of each article in the citation network, so that nodes with the same role are in similar topological positions. These measures were used to determine the presence of new clusters.

Another dynamic approach uses sub-classification of subject areas in the natural sciences, social sciences, and humanities, which

Table 2. Example of a data used from dataset

index	text embedding	title	title embedding	year	1-year	2-year	3-year	first author h index	first author paper count	first author citation count	sum h index	sum paper count	sum citation count
1	[-7.3826470375061035, -0.3564034700393677, 0.8...	Modules for Experiments in Stellar Astrophysic...	[-0.02009905, -0.022373319, 0.03170862, 0.0149...	[2013]	[45]	[121]	[216]	[27]	[51]	[8317]	[131]	[447]	[27096]
2	[-4.855892181396484, 1.4361735582351685, 2.138...	MODULES FOR EXPERIMENTS IN STELLAR ASTROPHYSIC...	[-0.020886937, 0.044551138, -0.0065947664, 0.0...	[2015]	[24]	[126]	[282]	[27]	[51]	[8317]	[117]	[443]	[29620]
3	[-5.774590015411377, -1.2082645893096924, 2.93...	Modules for Experiments in Stellar Astrophysic...	[-0.026900753, -0.035499975, 0.03234886, 0.029...	[2010]	[6]	[29]	[102]	[27]	[51]	[8317]	[190]	[813]	[58313]
...	...	...	...	...	...	...	...	...	...	...	...	...	...
6348	[-3.6669158935546875, -4.581037998199463, -1.9...	Spectr-W3 Online Database On Atomic Properties...	[-0.032371737, -0.030360237, -0.03526009, -0.0...	[2002]	[0]	[0]	[0]	[5]	[33]	[91]	[42]	[288]	[1933]
6349	[-3.8591768741607666, -2.6366443634033203, 1.3...	Performance of Magnetic Penetration Thermomete...	[-0.024682185, -0.046158567, 0.029100196, 0.03...	[2012]	[1]	[4]	[4]	[88]	[638]	[27971]	[227]	[1482]	[63066]
6350	[-1.6990535259246826, 0.5203409790992737, 2.11...	Wide Field X-ray Telescope: Mission Overview	[-0.022060653, -0.0299272, -0.04176955, 0.0037...	[2010]	[0]	[2]	[5]	[64]	[714]	[21367]	[160]	[1809]	[37241]

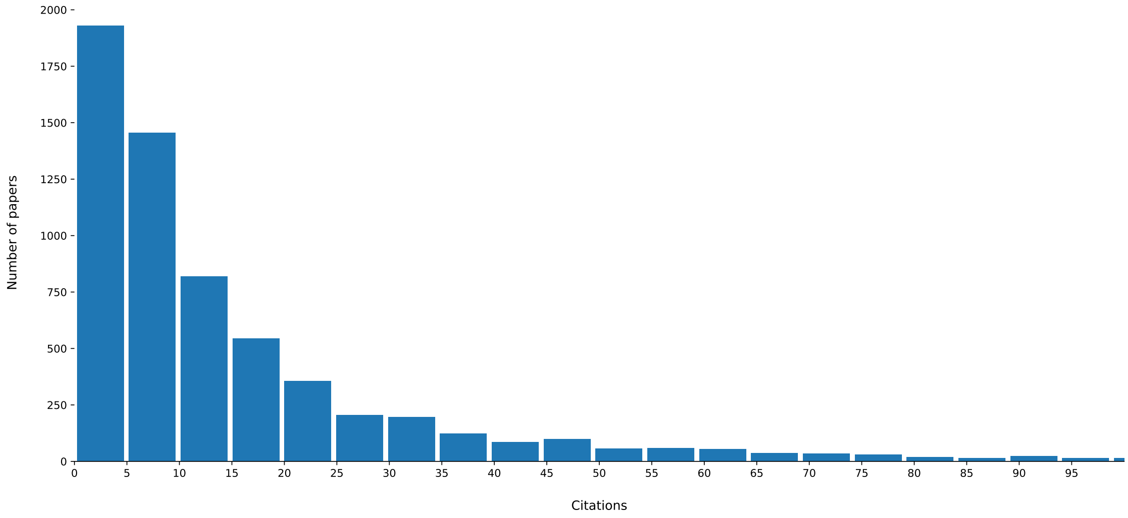


Fig. 4. Distribution of papers by amount of citations

can be performed based on text mining and textual similarities between documents; extracted terms can also be used in complex ways to label and describe the resulting clusters. [33].

One possibility to monitor the structural changes and evolution of the number of clusters is certainly the application of complementary methods. For example, the diachronic thematic communities approach is of significant interest in the present

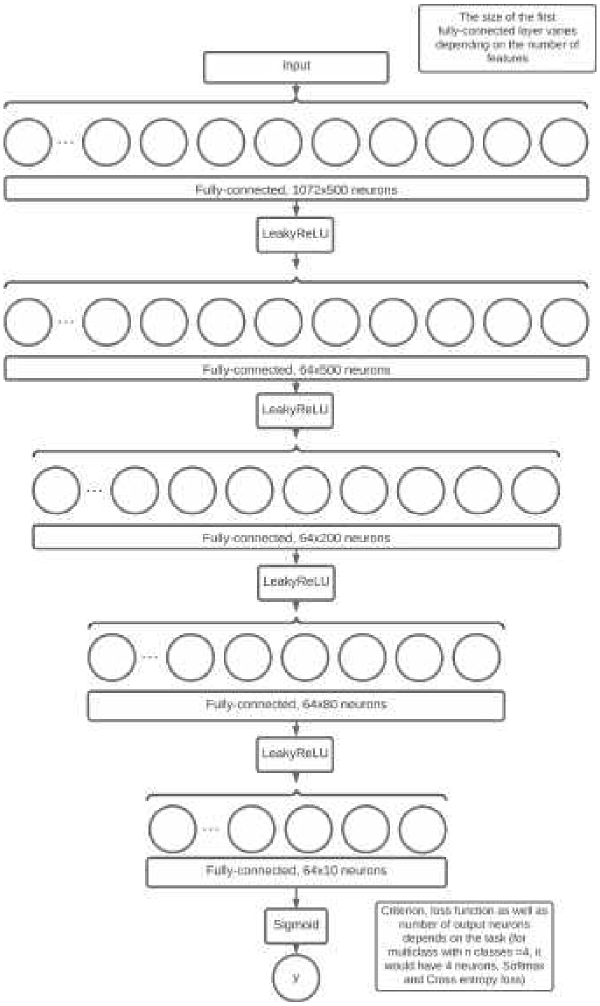

The size of the first fully-connected layer varies depending on the number of features
Input
Fully-connected, 1072x500 neurons
LeakyReLU
Fully-connected, 64x500 neurons
LeakyReLU
Fully-connected, 64x200 neurons
LeakyReLU
Fully-connected, 64x80 neurons
LeakyReLU
Fully-connected, 64x10 neurons
Sigmoid
y
Criterion, loss function as well as number of output neurons depends on the task (for multiclass with n classes =4, it would have 4 neurons, Softmax and Cross entropy loss)

Fig. 5. Neural network architecture for binary classification

program, its main idea being to find communities (temporary research teams, VNCs) of people working on semantically related topics at the same time.

These communities are interesting because their analysis allows us to understand the pattern of dynamics in the global research world, i.e., the birth of new communities or the rebirth of old ones; the migration of researchers from one topic to another; the splitting, merging, and discontinuation of communities; and more.

To this end, we are interested in developing graph-dynamics methods that are able to properly handle the dynamic aspects of topic community formation, giving priority to the relationships between researchers who seem to follow the same research trajectories.

The present approach was formulated in [32] with multiple sources for mining research. In another paper [39], it was extended in the form of Temporal Semantic Topic-Based Clustering (TST), which uses a new metric to cluster researchers according to their research trajectories, defined as the distribution of semantic topics over time.

4.2 Lexical Approach

Another method consists of the lexical approach proposed in [16]. It has several limitations, namely the relatively low ability to differentiate research topics, leading to "overestimation" of relations between documents.

Furthermore, with it there is a dimensionality problem in the context of big data, but the most significant problem is related to the fact that aggregating information at different levels using thesauri and ontology at different time periods leads to different readings of results.

Based on all of the above, a hybrid approach based on different architectures and solutions seems promising. Indeed, a number of authors [8, 25, 49] have proposed a similar strategy of combinatorial methods in related problems.

The question arises how such combinations of text-based and reference-based components can be applied to diachronic analysis, if each of the components causes some problems in the application to long-term analysis.

The inconsistency in the application of both bibliographic linking and joint citation analysis over periods of, say, more than 10 years is due to the aging literature and the genealogy of citations is evident.

Moreover, the use of collaborative citation analysis in the context of new emerging topics is questionable, as it takes time before a "critical mass" of articles on a new research topic is reached, which is necessary to produce the highly cited publications needed for collaborative work citation-based clustering [24]. In turn, the use of bibliographic linkage for long-term analysis is mostly limited to the impact of citation genealogy.

4.3 Scientific Indicators

Another approach considers methods of using specific indicators to refine a new field of study. Science indicators have been used to study the emergence or growth of scientific fields, such as the price index, the immediacy index, and the currency index [1, 29]. In [15], they investigate the emergence of knowledge as a result of scientific discovery and its disruptive effect on the structure of scientific communication.

They apply network analysis to illustrate this emergence in terms of journals, words, and citations. The paper [35] investigates changes in journal citation patterns during the emergence of a new field. Paper [28] discovers new technologies through citation network analysis, finding that fuel cell and solar cell research are fast-growing fields.

In [42] proposed approaches based on historiography and field mobility to trace the impact of a specific article. In their study, the citation historiographer generated by the 1968 Merton article shows the emergence of a new field of science and technology research in the 1970s.

They demonstrate that there is an algorithmic determinism in the origin of the new research field based on the formalization of academic trajectories of scholars, which in turn are conceptualized as pragmatic modes of knowledge dissemination. Many researchers use quantitative models to study how ideas spread in academic communities and how academic fields evolve over time.

Goffman has conducted several studies [17–20] to mathematically model the temporal development of scientific fields. In his work, he first formalizes the term Scientific Discovery (SD), the development of his idea is continued by Vityaev [52], in which knowledge is considered as a certain set of formal-logical statements with their corresponding probabilities.

Due to such a technique, the problems of ambiguity and ambiguity, as well as the problem of sharp divergence of truth estimates in the deductive-logical inference procedure are solved. Another interesting observation in Goffman's work is the assertion that there is a strong link between models of epidemic spread and the spread of knowledge in a particular field of research [20].

A development of the epidemiological approach has evolved in [4] analyzing the temporal evolution of new fields within several scientific disciplines by the number of authors and publications, using models of contagion developed in epidemiology. An alternative way to detect thematic trends is based on the idea of an algorithm from digital signal processing, namely the spike detection algorithm [31].

It uses a Markov automaton whose states correspond to the frequencies of individual words, and state transitions correspond to points in time around which the word frequency changes significantly.

Then, over a given set of time-stamped text, e.g., abstracts and years of article publication, the algorithm identifies those abstract words whose frequency increases dramatically, and outputs a list of these words along with the beginning and end of the sample period. The strength with which changes occur may indicate emergent properties, i.e., the potential for a study to be a breakthrough in its field. Ideas from digital signal processing have also found application in [36], in which a packet data algorithm was used to define breakthrough research terms with a high degree of time lag.

Its output can be used as indicators to identify future trends in the research field. The work [36] covers biomedical and other research from 1982 - 2001. An interesting idea is to combine different indicators to identify spikes in temporal structures.

4.4 Data Availability for Detection Methods

The main requirement for the above-mentioned approaches is to require the existence of a central direction that links all possible variations and alternatives into a unified structure. A S&T (Science and Technology) document or a set of such documents may serve as such a central direction. S&T documents can be understood as technology patents or research publications.

In turn, various online services of digital libraries, patent agencies, specialized news portals, etc.

Table 3. Features used in the experiment with their description

Feature	Description	Example data
Paper Embedding	Embedding of a full-text using SPECTRE	[-7.3826470375061035, -0.3564034700393677, 0.8...
Title Embedding	Embedding of the paper's title using FastText [38]	[-0.02009905, -0.022373319, 0.03170862, 0.0149..
Year of Publication	Year of the publication of the paper	2013
1-year total citations	Total citation for a given paper for the one year after it was published	45
2-year total citations	Total citation for a given paper for the two years after it was published	121
3-year total citations	Total citation for a given paper for the three years after it was published	216
First author h-index	H-index of the first authors of a publication	27
First author paper count	Paper count of the first author of the publication	51
First author citation count	Citation count of the first author of the publication	8317
Sum of first three and last author's h-index	Total sum of h-indexes of the first three and the last author of the publication	131
Sum of first three and last author's paper count	Total sum of paper count of the first three and the last author of the publication	447
Sum of first three and last author's citation count	Total sum of citation count of the first three and the last author of the publication	27096

Table 4. Results of classifiers with different feature inputs. For all models, the same experiments were carried out, the table shows only noteworthy examples. For multi-class classification we use 4 neurons on last layer and Cross-Entropy loss with Softmax

Model	Features	F1 score	Task
Random forest	3-year	0.93	binary
Random forest	1-year, 2-year, 3-year	0.92	binary
Random forest	full-text embedding, title embedding, year, 1-year, 2-year, 3-year	0.90	binary
Random forest	all features	0.89	binary
Neural network	all features	0.60	binary
Neural network	full-text embedding, title embedding, year, 1-year, 2-year, 3-year	0.88	binary
Neural network	full-text embedding, title embedding, year, 1-year, 2-year, 3-year	0.71	multi-class

can serve as data sources for all the above approaches. For example, since 1976, the United States Patent and Trademark Office (USPTO) has provided a full-text patent and search engine that can be used free of charge.

The USPTO services are used by R&D (Research and Development) policy makers, R&D managers, technology developers, and R&D planners and creators [2, 50, 57]. As a result, technology process agents can study trends, form R&D strategies in view of high competition.

Taking patents as a source of data can reduce the amount of information processed in the task of finding breakthrough technologies, but patents themselves represent a very large volume.

In 2012, more than 253,000 utility model patents were granted in the US system alone, bringing the total number of granted patents in the US to over eight million. It is worth noting that more than 75% of the information contained in patents is no longer being reused.

When looking for new technologies, most of these patents are of little interest, perhaps because they describe the gradual development of mature technologies or because they describe technologies with relatively low potential [6]. Another important source of data is online library services.

For example, Elsevier's Scopus is an open online database index that contains full-text materials and citation links for scientific publications. In addition, an important quality of this index is the fact that it only includes articles that have undergone double-blind review, which certainly has a positive impact on the quality of scientific material.

The name Scopus was taken from the Latin name of the Hammercope bird (Scopus umbretta), which has excellent navigation skills. The Scopus database was founded in 1966. The collection contains over 40,000 titles from approximately 11,678 international publishers, of which nearly 35,000 journals are peer-reviewed in top-level subject areas.

Scopus covers a variety of formats (books, journals, conference proceedings, and more). The fields of science covered by Scopus are technology, medicine, social sciences, arts, and humanities [22].

The purpose of this program is to obtain data on the fly to generate relevant analytics, but specially prepared datasets exist for solving applied research questions.

For example, BIGPATENT is a dataset consisting of 1.3 million US patent documents collected from publicly available Google Patents datasets using BigQuery. It contains patents filed after 1971 in nine different technology areas.

Compared to other datasets, BIGPATENT has the following properties: the summaries contain a richer discourse structure with more recurring entities; the terminology is evenly distributed in the input data; and the gold standard of this dataset has a large variety in text length [44].

5 Modern Perspective on the Discovery of Emergent Technologies Through the Use of Deep Learning Techniques

5.1 Dataset for Testing Emerging Technology Classifiers

The data used to generate the citation and co-authorship graphs came from Semantic Scholar's public API on the topic of astophysics. This dataset provides tables including abstract embeddings, author lists, paper ids, article titles, published year, number of citations and references, and fields of study.

Additionally, you can get citations per year for each paper and information about the authors such as their total paper count, h-indexes, and total number of citations. The data was acquired via this site's official API and assembled into a dataset, an example of which is provided in Table 2.

The dataset initially had 10,000 records; however, after deleting empty and duplicated records, we received 6350 records, which were then separated into train and test subsets in a 4:1 ratio, yielding 5080 records in the test and 1270 records in the validation datasets.

5.2 Methodology

We tackled the topic via the lens of classification, namely binary classification. Articles having fewer than ten citations are classified as class 0, whereas those with more than ten citations are classified as class 1. Using the Semantic Scholar API, we obtain the article's full-text vector representation.

The term "full-text embedding" refers to the vector representation of the complete text of a scientific publication generated using the SPECTRE language model [7]. SPECTRE is a transformer-based language model for the production of document-level embeddings of scientific documents.

It is a pre-trained SciBert on a specific aim that modifies weights depending on the full-texts of publications and their citation relationships. To obtain the vector representation of the title, we utilize FastText's pre-trained model Common Crawl (600B tokens) with 2 million word vectors trained with subword information [38].

Our model is implemented in PyTorch. The model is a feed-forward neural network with five fully connected hidden layers; the signal is passed through the LearkyReLU function between the hidden levels. Figure 5 illustrates a more thorough construction.

The model is fed a vector of length 1078 bytes, where the full-text embedding is 768 bytes, the title embedding is 300 bytes, and the other characteristics are integers. The full descriptions of all features are listed in Table3.

At the output, there is a single neuron, after which sigmoid is applied and the result is regarded binary cross-entropy. We utilized the Adam optimizer [30] with a learning rate of 1e-4 and the ReduceLROnPlateau scheduler for training. We train our model on two RTX 2080Ti cards (11 GB each) for seven epochs with a batch size of 64 and no gradient accumulations.

To compare, we trained our model using both multi-class classification (class 0: n = 3, class 1: n = 8, class 2: n = 20, and class 3: n > 20) and Random Forest.

5.3 Results

Following a series of experiments employing a variety of approaches, the following results were obtained, as shown in Table 4. As shown in the table, both the neural network and decision trees perform better with a subset of features (full-text embedding, title embedding, year, 1-year, 2-year, and 3-year) than with all features; therefore, our hypothesis that author-specific metrics may improve prediction results was not supported.

We attribute this to the curse of dimensionality because the training set contains somewhat more than 5000 samples and the breadth of the model increases as a result of the usage of embeddings. Despite this, the decision trees performed well, particularly when compared to the neural network. However, studies with a different set of characteristics have revealed that the number of citations for 1, 2, and 3 years is the most disproportionately significant factor.

6 Conclusion and Future Work

It must be appealing to create models of automatic identification of ground-breaking research and technology for a variety of human endeavors; this is the focus of the present work.

The writers studied methodological aspects of research and technological progression; thus, it should be mentioned that from a technical standpoint, theoretical research is far from complete due to the wide diversity of projects, outstanding challenges, research questions, and market assumptions.

We believe it is conceivable to advocate the creation of distinct open data sets for each technological industry as the next step. In addition, we should observe the shift from statistical to intellectual scientometrics, as indicated by the abundance of works devoted to forecasting and predicting various research performance indicators.

Using the space sector as an example, the authors of this research also analyzed several machine learning algorithms for forecasting citation indices. The outcomes of recent numerical studies are positive. Searching for numerous extra input features to increase the quality of the prediction and shifting from classification to regression constitutes additional development.

Acknowledgments

This research was funded by Aerospace Committee of the Ministry of Digital Development, Innovations and Aerospace Industry of the Republic of Kazakhstan grant number BR11265420.

References

1. **Almeida, J., Pais, A., Formosinho, S. (2009).** Science indicators and science patterns in europe. Journal of Informetrics, Vol. 3, pp. 134–142. DOI: 10.1016/j.joi.2009.01.001.

2. **Altuntas, S., Dereli, T. (2015).** Forecasting technology success based on patent data. Technological Forecasting and Social Change, Vol. 96, pp. 202–214. DOI: 10.1016/j.techfore.2015.03.011.

3. **Atkinson, R. D. (2021).** Why the united states needs a national advanced industry and technology agency. Why the United States Needs a National Advanced Industry and Technology Agency.

4. **Bettencourt, L., Kaiser, D., Kaur, J., Castillo-Chávez, C., Wojick, D. (2008).** Population modeling of the emergence and development of scientific fields. Scientometrics, Vol. 75, No. 3, pp. 495–518. DOI: 10.1007/s11192-007-1888-4.

5. **Braun, T., Schubert, A., Zsindely, S. (1997).** Nanoscience and nanotecnology on the balance. Scientometrics, Vol. 38, pp. 321–325. DOI: 10.1007/BF02457417.

6. **Breitzman, A., Thomas, P. (2015).** The emerging clusters model: A tool for identifying emerging technologies across multiple patent systems. Research Policy, Vol. 44, No. 1, pp. 195–205. DOI: 10.1016/j.respol.2014.06.006.

7. **Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D. (2020).** SPECTER: Document-level representation learning using citation-informed transformers. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. DOI: 10.18653/v1/2020.acl-main.207.

8. **Correia, A., Jameel, S., Schneider, D., Paredes, H., Fonseca, B. (2020).** A workflow-based methodological framework for hybrid human-ai enabled scientometrics. 2020 IEEE International Conference on Big Data (Big Data), IEEE, pp. 2876–2883. DOI: 10.1109/BigData50022.2020.9378096.

9. **Couper, M. P. (2005).** Technology trends in survey data collection. Social Science Computer Review, Vol. 23, No. 4, pp. 486–501. DOI: 10.1177/0894439305278972.

10. **Cozzens, S., Gatchair, S., Kang, J., Kim, K.-S., Lee, H., Ordonez-Matamoros, G., Porter, A. (2010).** Emerging technologies: Quantitative identification and measurement. Technology Analysis & Strategic Management, Vol. 22, No. 3, pp. 361–376. DOI: 10.1080/09537321003647396.

11. **Cozzens, S., Gatchair, S., Kang, J., Kim, K. S., Lee, H. J., Ordóñez, G., Porter, A. (2010).** Emerging technologies: quantitative identification and measurement. Technology Analysis & Strategic Management, Vol. 22, No. 3, pp. 361–376. DOI: 10.1080/09537321003647396.

12. **Cullen, P., Duffy, A., O'Donnell, C., O'Callaghan, D. (2000).** Process viscometry for the food industry. Trends in Food Science & Technology, Vol. 11, No. 12, pp. 451–457. DOI: 10.1016/S0924-2244(01)00034-6.

13. **Dummitt, B., Chang, Y. H. (2006).** Molecular beacons for dna binding proteins: An emerging technology for detection of dna binding proteins and their ligands. ASSAY and Drug Development Technologies, Vol. 4, No. 3, pp. 343–349. DOI: 10.1089/adt.2006.4.343.

14. **Fall, C., Törcsvári, A., Benzineb, K., Karetka, G. (2003).** Automated categorization in the international patent classification. ACM SIGIR Forum, Vol. 37, No. 1, pp. 10–25. DOI: 10.1145/945546.945547.

15. **Froyland, G. (2001).** Extracting dynamical behavior via markov models. Nonlinear Dynamics and Statistics, pp. 281–321. DOI: 10.1007/978-1-4612-0177-9_12.

16. **Glänzel, W., Thijs, B. (2012).** Using 'core documents' for detecting and labelling new emerging topics. Scientometrics, Vol. 91, No. 2, pp. 399–416. DOI: 10.1007/s11192-011-0591-7.

17. **Goffman, W. (1966).** Mathematical approach to the spread of scientific ideas—the history of mast cell research. Nature, Vol. 212, No. 5061, pp. 449–452. DOI: 10.1038/212449a0.

18. **Goffman, W. (1971).** A mathematical method for analyzing the growth of a scientific discipline. Journal of the ACM, Vol. 18, No. 2, pp. 173–185. DOI: 10.1145/321637.321640.

19. **Goffman, W., Harmon, G. (1971).** Mathematical approach to the prediction of scientific discovery. Nature, Vol. 229, No. 5280, pp. 103–104. DOI: 10.1038/229103a0.

20. **Goffman, W., Newill, V. (1964).** Generalization of epidemic theory: An application to the transmission of ideas. Nature, Vol. 204, No. 4955, pp. 225–228. DOI: 10.1038/204225a0.

21. **Goldstein, J. (1999).** Emergence as a construct: History and issues. Emergence, Vol. 1, No. 1, pp. 49–72. DOI: 10.1207/s15327000em0101_4.

22. **Guz, A., Rushchitsky, J. (2009).** Scopus: A system for the evaluation of scientific journals. International Applied Mechanics, Vol. 45, No. 4, pp. 351–362. DOI: 10.1007/s10778-009-0189-4.

23. **Harris, R. H., English, C. W., Highland, J. H. (1985).** Hazardous waste disposal: Emerging technologies and public policies to reduce public health risks. Annual Review of Public Health, Vol. 6, No. 1, pp. 269–294. DOI: 10.1146/annurev.pu.06.050185.001413.

24. **Hicks, D. (1987).** Limitations of co-citation analysis as a tool for science policy. Social Studies of

Science - SOC STUD SCI, Vol. 17, No. 2, pp. 295–316. DOI: 10.1177/030631287017002004.

25. **Janssens, F., Glänzel, W., De Moor, B. (2008).** A hybrid mapping of information science. Scientometrics, Vol. 75, No. 3, pp. 607–631. DOI: 10.1007/s11192-007-2002-7.

26. **Jo, Y., Lagoze, C., Giles, C. (2007).** Detecting research topics via the correlation between graphs and texts. ACM Press, pp. 370–379. DOI: 10.1145/1281192.1281234.

27. **Jones, B., Weinberg, B. (2011).** Age dynamics in scientific creativity. Proceedings of the National Academy of Sciences of the United States of America, Vol. 108, No. 47, pp. 18910–18914. DOI: 10.1073/pnas.1102895108.

28. **Kajikawa, Y., Yoshikawa, J., Takeda, Y., Matsushima, K. (2008).** Tracking emerging technologies in energy research: Toward a roadmap for sustainable energy. Technological Forecasting and Social Change, Vol. 75, No. 6, pp. 771–782. DOI: 10.1016/j.techfore.2007.05.005.

29. **King, J. (1987).** A review of bibliometric and other science indicators and their role in research evaluation. Journal of Information Science, Vol. 13, No. 5, pp. 261–276. DOI: 10.1177/016555158701300501.

30. **Kingma, D. P., Ba, J. (2014).** Adam: A method for stochastic optimization. DOI: 10.48550/ARXIV.1412.6980.

31. **Kleinberg, J. (2003).** Bursty and hierarchical structure in streams. Data Mining and Knowledge Discovery, Vol. 7, No. 4, pp. 373–397. DOI: 10.1023/A\%3A1024940629314.

32. **Lamirel, J. C., Ta, A. P., Attik, M. (2008).** Novel labeling strategies for hierarchical representation of multidimensional data analysis results, pp. 169–174.

33. **Lee, J., Lee, D. (2005).** An improved cluster labeling method for support vector clustering. IEEE transactions on pattern analysis and machine intelligence, Vol. 27, No. 3, pp. 461–464. DOI: 10.1109/TPAMI.2005.47.

34. **Lee, W. H. (2008).** How to identify emerging research fields using scientometrics: An example in the field of information security. Scientometrics, Vol. 76, No. 3, pp. 503–525. DOI: 10.1007/s11192-007-1898-2.

35. **Leydesdorff, L., Schank, T. (2009).** Dynamic animations of journal maps: Indicators of structural changes and interdisciplinary developments. Journal of the American Society for Information Science and Technology, Vol. 59, No. 11, pp. 1810–1818. DOI: 10.1002/asi.20891.

36. **Mane, K., Borner, K. (2004).** Mapping topics and topic bursts in pnas. Proceedings of the National Academy of Sciences of the United States of America, Vol. 101, No. 1, pp. 5287–5290. DOI: 10.1073/pnas.0307626100.

37. **Martin, B. R. (1995).** Foresight in science and technology. Technology Analysis & Strategic Management, Vol. 7, No. 2, pp. 139–168. DOI: 10.1080/09537329508524202.

38. **Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., Joulin, A. (2017).** Advances in pre-training distributed word representations. arXiv. DOI: 10.48550/ARXIV.1712.09405.

39. **Osborne, F., Scavo, G., Motta, E. (2014).** Identifying diachronic topic-based research communities by clustering shared research trajectories. Lecture Notes in Computer Science, pp. 114–129. DOI: 10.1007/978-3-319-07443-6_9.

40. **Ramachary, D. B., Reddy, Y. V. (2011).** Dienamine catalysis: An emerging technology in organic synthesis. European Journal of Organic Chemistry, Vol. 2012, No. 5, pp. 865–887. DOI: 10.1002/ejoc.201101157.

41. **Rotolo, D., Hicks, D., Martin, B. (2015).** What is an emerging technology? Research Policy, Vol. 44, No. 10, pp. 1827–1843. DOI: 10.1016/j.respol.2015.06.006.

42. **Scharnhorst, A., Garfield, E. (2010).** Tracing scientific influence. Dynamics of Socio-Economic Systems, Vol. 2. DOI: 10.48550/ARXIV.1010.3525.

43. **Serenko, A., Bontis, N., Booker, L. D., Sadeddin, K. W., Hardie, T. (2010).** A scientometric analysis of knowledge management and intellectual capital academic literature. Journal of Knowledge Management, Vol. 14, No. 1, pp. 3–23. DOI: 10.1108/13673271011015534.

44. **Sharma, E., Li, C., Wang, L. (2019).** BIGPATENT: A large-scale dataset for abstractive and coherent summarization. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. DOI: 10.18653/v1/p19-1212.

45. **Shibata, N., Kajikawa, Y., Takeda, Y., Matsushima, K. (2008).** Detecting emerging research fronts based on topological measures in citation networks of scientific publications. Technovation, Vol. 28, No. 11, pp. 758–775. DOI: 10.1016/j.technovation.2008.03.009.

46. **Small, H., Boyack, K. W., Klavans, R. (2014).** Identifying emerging topics in science and technology. Research Policy, Vol. 43, No. 8, pp. 1450–1467. DOI: 10.1016/j.respol.2014.02.005.

47. **Takeda, Y., Kajikawa, Y. (2008).** Optics: A bibliometric approach to detect emerging research domains and intellectual bases. Scientometrics, Vol. 78, No. 3, pp. 543–558. DOI: 10.1007/s11192-007-2012-5.

48. **Takeda, Y., Mae, S., Kajikawa, Y., Matsushima, K. (2009).** Nanobiotechnology as an emerging research domain from nanotechnology: A bibliometric approach. Scientometrics, Vol. 80, No. 1, pp. 23–38. DOI: 10.1007/s11192-007-1897-3.

49. **Thijs, B., Glänzel, W. (2018).** The contribution of the lexical component in hybrid clustering, the case of four decades of “scientometrics”. Scientometrics, Vol. 115, No. 1, pp. 21–33. DOI: 10.1007/s11192-018-2659-0.

50. **Thorleuchter, D., Van den Poel, D., Prinzie, A. (2010).** A compared R&D-based and patent-based cross impact analysis for identifying relationships between technologies. Technological Forecasting and Social Change, Vol. 77, No. 7, pp. 1037–1050. DOI: 10.1016/j.techfore.2010.03.002.

51. **van Raan, A. F. J. (2000).** On growth, ageing, and fractal differentiation of science. Scientometrics, Vol. 47, No. 2, pp. 347–362. DOI: 10.1023/a:1005647328460.

52. **Vityaev, E., Kovalerchuk, B. (2004).** Discovery of empirical theories based on the measurement theory. Minds and Machines, Vol. 14, No. 4, pp. 551–573. DOI: 10.1023/B:MIND.0000045991.67908.13.

53. **Waksman, R. (2000).** Vascular brachytherapy: update on clinical trials. The Journal of invasive cardiology, Vol. 12, pp. 18–28.

54. **Watts, R. J., Porter, A. L. (2003).** R&D cluster quality measures and technology maturity. Technological Forecasting and Social Change, Vol. 70, No. 8, pp. 735–758. DOI: 10.1016/S0040-1625(02)00355-4.

55. **Wayne, C. L. (1998).** Topic detection & tracking (TDT) overview & perspective. pp. .

56. **Wu, L., Wang, D., Evans, J. A. (2019).** Large teams develop and small teams disrupt science and technology. Nature, Vol. 566, No. 7744, pp. 378–382. DOI: 10.1038/s41586-019-0941-9.

57. **Zhang, L., Zhao, J., Lu, H., Gong, L., Li, L., Zheng, J., Zhu, Z. (2011).** High sensitive and selective formaldehyde sensors based on nanoparticle-assembled zno micro-octahedrons synthesized by homogeneous precipitation method. Sensors and Actuators B Chemical, Vol. 160, No. 1, pp. 364–370. DOI: 10.1016/j.snb.2011.07.062.

58. **Zitt, M., Bassecoulard, E. (2008).** Challenges for scientometric indicators: Data demining, knowledge-flow measurements and diversity issues. Ethics in Science and Environmental Politics, Vol. 8, pp. 49–60. DOI: 10.3354/esep00092.

Article received on 09/08/2022; accepted on 21/10/2022.
Corresponding author is Irina Gelbukh.

ISSN 2007-9737

Similarity Correlation of Frequency Distributions of Categorical Data in Analysis of Cognitive Decline Severity in Asthmatics

Ildar Z. Batyrshin[1], Imre J. Rudas[2], Nailya Kubysheva[3], Svetlana Akhtyamova[4]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] Obuda University, Budapest,
Hungary

[3] Kazan Federal University, Kazan,
Russia

[4] Kazan National Research Technological University, Kazan,
Russia

batyr1@gmail.com, rudas@uni-obuda.hu, aibolit70@mail.ru, ahtjamovasve@yandex.ru

Abstract. The paper presents the method of measuring the similarity and difference in the frequency distributions of one categorical variable for different levels of another variable. This method calculates the similarity and correlation between the rows of the contingency table. In this work, we use it for the analysis of associations of cognitive indicators. The proposed categorical data association analysis method can be used as an addition to the classical chi-square test.

Keywords. Correlation, frequency distribution, categorical data, cognitive impairment.

1 Introduction

Recently it was proposed a general approach to the representation and construction of correlation and association coefficients [1-3]. They are considered as (correlation or association) functions of two arguments defined on a set with involutive operation, taking values in $[-1,1]$ and satisfying several properties. The involutive operation is mapping elements of the domain into "opposite" elements, such that the correlation between mutually opposite elements equals -1. It was shown [2-3] that most of the traditional correlation and association coefficients considered in statistics [4-5] and taking values in the interval $[-1,1]$ are correlation functions.

Correlation functions can be generated by similarity and dissimilarity functions satisfying suitable properties [1-3,6]. Moreover, a one-to-one correspondence exists between correlation and bipolar similarity functions [2]. For this reason, correlation functions are also referred to as invertible similarity correlations.

These results pave the way to construct correlation functions on almost any set if one can define an involutive operation and suitable similarity or dissimilarity function on this set. This approach was used in [7] to introduce a bipolar dissimilarity function and corresponding correlation function on the set of probability and relative frequency distributions. This proposal used the involutive negation of probability distributions [8]. The constructed correlation function surprisingly coincided with the Pearson product-moment correlation coefficient [4,5]. The authors of [7] used the introduced correlation function for calculating the correlation between frequency distributions in contingency tables.

doi: 10.13053/CyS-26-4-4439

Usually, these frequency distributions are defined on a set of categories of categorical variables presented in contingency tables [9-12].

In this paper, we apply the dissimilarity and correlation functions proposed in [7] to analyze the cognitive decline severity in asthmatics.

Currently, the manifestation of cognitive impairment in various diseases, including bronchial asthma, is being actively investigated [13]. Cognitive dysfunctions affect the control and development of asthma, which determines the relevance of more detailed studies of the role of cognitive impairment in the course of the disease. Various factors, including age, disease duration, education, lifestyle, etc., determine the degree of cognitive decline in asthmatics.

As a rule, existing tests for assessing cognitive impairment are analyzed by traditional statistical analysis methods allowing us to determine the significance of associations between the studied indicators [9,10]. In this paper, we analyze new data about relationships between the sociodemographic data of patients and the severity of cognitive decline in asthmatics published in [14].

The paper has the following structure. Section 2 discusses the basics of the theory of invertible similarity correlations.

The dissimilarity and correlation functions introduced in [7] for the analysis of relationships between frequency distributions and categorical data are considered in Section 3. Section 4 presents the data from [14]. Section 5 describes the results of the analysis of these data using the proposed method. The last Section discusses obtained results, a conclusion, and future work.

2 Basics of Correlation Functions

Consider the basic definitions and results related with correlation functions [1-3]. Let Ω be a set with more than one element. A function $N: \Omega \rightarrow \Omega$ is referred to as a *reflection* or a *negation* on Ω if, for all x in Ω, it satisfies the *involutivity* property:

$$N(N(x)) = x, \qquad (1)$$

and N is not an identity function, i.e., $N(x) \neq x$ for some x in Ω. An element x in Ω satisfying the property:

$$N(x) = x$$

is called a *fixed point* of the negation N. The set of all fixed points of the negation N in Ω is denoted as $FP(\Omega)$. This set can be empty.

Denote V a non-empty subset of $\Omega \setminus FP(\Omega)$ closed under N. The set V does not contain fixed points of N, and if x in V, then $N(x)$ is also in V.

For any element x in V, its negation $N(x)$ can be considered as an element *opposite* to x. From the involutivity property (1), it follows that the element x is also opposite to $N(x)$. Hence for every x in V the elements x and $N(x)$ are mutually opposite, and $N(x)$ is also in V.

A correlation function (association measure) on V is a function $A: V \times V \rightarrow [-1,1]$ satisfying for all x, y in V the following properties [1]:

Symmetry:

A1. $A(x,y) = A(y,x)$,

Reflexivity:

A2. $A(x,x) = 1$,

Inverse relationship:

A3. $A(x,N(y)) = -A(x,y)$.

Usually, correlation and association coefficients used in statistics are calculated between real variables, dichotomous variables, rankings, etc., without considering some involutive operation on the corresponding set of variables [4, 5]. For such coefficients taking values in the interval $[-1,1]$, only the properties A1 and A2 are considered. But it was shown [1-3] that involutive operation can be introduced on the domains of traditional correlation coefficients like Pearson, Spearman, Kendall correlation, etc., and they will satisfy the inverse relationship property A3. For this reason, correlation functions satisfying property A3 also referred to as *invertible correlation functions* [2].

From properties A1-A3, it follows that the correlation function satisfies for all x in V the property:

Opposite elements:

$$A\big(x,N(x)\big) = -1.$$

Hence the correlation between opposite elements equals -1.

Correlation functions can be constructed from similarity and dissimilarity functions [1-3].

A function $D: V \times V \rightarrow [0,1]$ is a *dissimilarity function* on V if for all x, y in V, it is *symmetric*:

$$D(x,y) = D(y,x)$$

and *irreflexive*:

$$D(x,x) = 0.$$

A function $S: V \times V \rightarrow [0,1]$ is a *similarity function* on V if for all x, y in V, it is *symmetric*:

$$S(x,y) = S(y,x),$$

and *reflexive*:

$$S(x,x) = 1.$$

Similarity S and dissimilarity D functions are *dual* if for all x, y in V it is fulfilled:

$$S(x,y) = 1 - D(x,y), \quad (2)$$

$$D(x,y) = 1 - S(x,y). \quad (3)$$

These functions are called *bipolar* if, for all x, y in V, they satisfy the following properties [2]:

$$S(x,y) + S\big(x, N(y)\big) = 1,$$

$$D(x,y) + D\big(x, N(y)\big) = 1.$$

There exists a one-to-one correspondence between invertible correlation functions and bipolar similarity (dissimilarity) functions [2]:

$$A(x,y) = 2S(x,y) - 1, \quad (4)$$

$$A(x,y) = 1 - 2D(x,y). \quad (5)$$

These three functions compose complementary triplet (S, D, A) and are also related as follows:

$$A(x,y) = \mathrm{S(x,y)} - D(x,y). \quad (6)$$

As it follows from (4), the invertible correlation function is nothing else but a rescaled bipolar similarity function. For this reason, the correlation function is also referred to as a *similarity correlation function*. From (4), we see that the similarity values from interval $[0,1]$ are linearly transformed into correlation values in the interval $[-1,1]$. For example, we have:

$$A(x,y) = 1 \quad \text{if} \quad S(x,y) = 1,$$

$$A(x,y) = 0 \quad \text{if} \quad S(x,y) = 0.5,$$

$$A(x,y) = -1 \quad \text{if} \quad S(x,y) = 0.$$

As we can see, the correlation is positive if $S(x,y) > 0.5$ (similarity value is high) and negative if $S(x,y) < 0.5$ (similarity value is low). Dually, we see from (5) that the correlation is positive if $D(x,y) < 0.5$, and negative, if $D(x,y) > 0.5$.

In the following Section, we show how the correlation between frequency distributions is constructed using the involutive negation of probability distribution and suitable dissimilarity function between distributions.

3 Correlation of Frequency and Probability Distributions

The correlation of frequency distributions was introduced in [7]. We give a short description of the steps described in the previous Section and used for constructing the corresponding correlation function. Suppose, $F = \{f_1, \dots, f_n\}$ is a frequency distribution of n categories, where f_i is a non-negative integer value of the frequency of appearance of the i-th category in some experiments. Transform F in relative frequency distribution $P = \{p_1, \dots, p_n\}$, where:

$$p_i = \frac{f_i}{\sum_{i=1}^{n} f_i}, \ i = 1, \dots, n.$$

We can consider P as a probability distribution, where p_i is a probability that the result of a random experiment will fall in i-th category. We have:

$$0 \leq p_i \leq 1, \qquad \sum_{i=1}^{n} p_i = 1.$$

Let Ω be a set of possible probability distributions with n elements. General methods of construction of negations of probability distributions are considered in [15]. The involutive negation of probability distributions is defined as follows [8]:

$$N(p_i) = \frac{MP - p_i}{nMP - 1},$$

where $MP = \max(P) + \min(P)$, and $\max(P) = \max\{p_1, \dots, p_n\}$, $\min(P) = \min\{p_1, \dots, p_n\}$.

The uniform probability distribution

$$P_U = \left(\frac{1}{n}, \dots, \frac{1}{n}\right)$$

is a unique fixed point of the negation N.

The bipolar dissimilarity function D on the set of probability distributions $V = \Omega \backslash \{P_U\}$ is defined as follows [7]:

$$D(P,Q) = \frac{1}{4}\sum_{i=1}^{n}\left[\frac{np_i - 1}{\sqrt{\sum_{i=1}^{n}(np_i-1)^2}} - \frac{nq_i - 1}{\sqrt{\sum_{i=1}^{n}(nq_i-1)^2}}\right]^2. \qquad (7)$$

It defines by (5) the invertible correlation function [7]:

$$A(P,Q) = \frac{\sum_{i=1}^{n}(np_i-1)(nq_i-1)}{\sqrt{\sum_{i=1}^{n}(np_i-1)^2}\sqrt{\sum_{i=1}^{n}(nq_i-1)^2}}. \qquad (8)$$

This correlation function coincides with Pearson product-moment correlation coefficient.

In the following Section, we use this correlation function to analyze cognitive decline severity in asthmatics.

4 Data of Cognitive Declines in Asthmatics

When analyzing the literature devoted to assessing the effect of cognitive impairment on various characteristics in patients with asthma, we were attracted by the article of Haq Satti et al. [14]. These authors studied associations between sociodemographic factors and cognitive decline severity in asthmatics. Table 1 presents the results obtained in this work.

This study showed [14] that the Duration of Illness and the use of Poly-Pharmacy were closely associated with the presence and severity of cognitive decline (p=0.005 and p=0.019, respectively).

In this paper, we analyzed the presented data using the similarity and correlation of frequency distributions considered in the previous Section.

5 Results

Since the number of severe cognitive declines in Table 1 is small, we combined the last two columns into one column.

Table 1. Characteristics of the asthmatic patients and their cognitive decline severity. Adapted from [14]

Factors	No Cog. Decline	Mild Cog. Decline	Moder. Cog. Decline	Severe Cog. Decline
Total	N (%) 68 (50.4%)	N (%) 45 (33.3%)	N (%) 16 (11.8%)	N (%) 6 (4.4%)
Age				
25-40	30 (44.1%)	17 (37.8%)	6 (37.5%)	2 (33.3%)
>40	38 (55.9%)	28 (62.2%)	10 (62.5%)	4 (66.7%)
Education				
10 or less	53 (77.9%)	32 (71.1%)	12 (75%)	4 (66.7%)
>10	15 (22.1%)	13 (28.9%)	4 (25%)	2 (33.3%)
Duration of Illness				
<5 years	63 (92.6%)	32 (71.1%)	13 (81.2%)	3 (50%)
>5 years	05 (7.4%)	13 (28.9%)	3 (18.8%)	3 (50%)
Tobacco Smoking				
Non Smoker	34 (50%)	16 (35.5%)	5 (31.2%)	2 (33.3%)
Smoker	34 (50%)	29 (64.5%)	11 (68.2%)	4 (66.7%)
Poly-Pharmacy				
No	36 (52.9%)	12 (26.6%)	4 (25%)	3 (50%)
Yes	32 (47.1%)	33 (73.4%)	12 (75%)	3 (50%)

Note: Cog. – cognitive; Moder. – moderate

In addition, we transformed the frequency of patients in each cell of the table into relative frequency such that their sum in every string equals to 1 (see Table 2).

As a result, we obtain for each of the five factors two relative frequency (probability) distributions of the categorical variable Cognitive Decline Severity containing three levels (categories): No Cognitive Decline, Mild Cognitive Decline, and Moderate or Severe Cognitive Decline.

Denoting the first of each pair of distributions by P and the second by Q we calculate relationships between them as follows: dissimilarity $D(P,Q)$ by (7), similarity $S(P,Q)$ by (2) and correlation $A(P,Q)$ by (8). The results are presented below:

Age:

25-40: P = (0.545, 0.31, 0.145),
>40: Q = (0.475, 0.35, 0.175),
$D(P,Q)$: 0.0100,
$S(P,Q)$: 0.9900;
$A(P,Q)$: 0.9800;

Education:

10 or less: P = (0.52, 0.32, 0.16),
>10: Q = (0.44, 0.38, 0.18),
$D(P,Q)$: 0.0372,
$S(P,Q)$: 0.9628,
$A(P,Q)$: 0.9256;

Duration of Illness:

<5 years: P = (0.57, 0.29, 0.14),
>5 years : Q = (0.21, 0.54, 0.25),
$\boldsymbol{D(P,Q)}$**:** **0.6464,**
$\boldsymbol{S(P,Q)}$**:** **0.3536,**
$\boldsymbol{A(P,Q)}$**:** **-0.2928**;

Tobacco Smoking:

Non Smoker: P = (0.60, 0.28, 0.12),
Smoker: Q = (0.44, 0.37, 0.19),
$D(P,Q)$: 0.0513,
$S(P,Q)$: 0.9487,
$A(P,Q)$: 0.8973;

Poly-Pharmacy:

No: P = (0.65, 0.22, 0.13),
Yes: Q = (0.40, 0.41, 0.19),
$D(P,Q)$: 0.2030,
$S(P,Q)$: 0.7970,
$A(P,Q)$: 0.5941.

One can see that relative frequency distributions P and Q of the categorical variable Cognitive Decline Severity for both levels of factors Age, Education, and Tobacco Smoking are very similar. The similarity between them is more than 0.94, and the correlation is more than 0.89. A change in levels of factors does not cause a significant change in distributions. For this reason, one can conclude that Cognitive Decline Severity is not associated with these factors or that these associations are insignificant.

Table 2. Characteristics of the asthmatic patients and their cognitive decline severity (modified Table 1)

Factors	No Cog. Decline	Mild. Cog. Decline	Moder. + Severe Cognitive Decline
Total	N=68	N=45	N=22
Age			
25-40 (n=55)	30 (0.545)	17 (0.31)	8 (0.145)
>40 (n=80)	38 (0.475)	28 (0.35)	14 (0.175)
Education			
10 or less (n=101)	53 (0.52)	32 (0.32)	16 (0.16)
>10 (n=34)	15 (0.44)	13 (0.38)	6 (0.18)
Duration of Illness			
<5 years (n=111)	63 (0.57)	32 (0.29)	16 (0.14)
>5 years (n=24)	5 (0.21)	13 (0.54)	6 (0.25)
Tobacco Smoking			
Non Smoker (n=57)	34 (0.60)	16 (0.28)	7 (0.12)
Smoker (n=78)	34 (0.44)	29 (0.37)	15 (0.19)
Poly-Pharmacy			
No (n=55)	36 (0.65)	12 (0.22)	7 (0.13)
Yes (n=80)	32 (0.4)	33 (0.41)	15 (0.19)

Note: Cog. – cognitive; Moder. - moderate

On the contrary, the frequency distributions P and Q of the two Duration of Illness factor levels have considerable differences.

The similarity is less than 0.5, the dissimilarity is greater than 0.5, and the correlation is negative (see also (6) for the relationship between these three functions).

These results allow us to conclude that Cognitive Decline Severity is associated with the factor Duration of Illness because the change in the levels of this factor causes a considerable change in corresponding distributions. This result is consistent with the results of the work [14].

Although the difference between distributions of Poly-Pharmacy is more considerable than for the first three factors, the similarity between distributions is high, and correlation has a high positive value.

For this reason, we can conclude that the association between Cognitive Decline Severity and Poly-Pharmacy is not very high.

6 Discussion and Conclusion

The method presented in this paper allows us to measure the similarity and difference in the frequency distributions of one categorical variable for different levels of another variable. Our method is based on calculating the similarities and correlations between the rows of the contingency table.

The proposed categorical data association analysis method can be used as an additional relationship assessment to the classical chi-square analysis method.

A comparative analysis of the results obtained in our work and [14], in which the Pearson chi-square test was used, showed the same associations for four factors. At the same time, our calculations revealed a significant similarity and positive correlation (r=0.6) between the degree of cognitive decline for different levels of Poly-Pharmacy, indicating a not large association between the considered variables.

The differences obtained require more detailed further research on the relationship between our and classical methods used to analyze the association of categorical data.

Frequency distributions appear in social-behavioral sciences, biology, medicine, marketing, business, etc. [9-12, 16-18]. We plan to apply the proposed method to data analysis in some of these areas. Another possible application of the considered methods is an analysis of relationships between subjective probability distributions and subjective weight distributions in models of probability reasoning and multicriteria decision-making [19].

Acknowledgments

This work was supported by the Strategic Academic Leadership Program of Kazan Federal University, the program for developing the Scientific-Educational Mathematical Center of Volga Federal District, and the project IPN SIP 20220857.

References

1. **Batyrshin, I. Z. (2015).** On definition and construction of association measures. Journal of Intelligent & Fuzzy Systems, Vol. 29, No. 6, pp. 2319–2326. DOI: 10.3233/IFS-151930.
2. **Batyrshin, I. Z. (2019).** Constructing correlation coefficients from similarity and dissimilarity functions. Acta Polytechnica Hungarica. Vol. 16, No. 10, pp. 191–204.
3. **Batyrshin, I. Z. (2019).** Data science: Similarity, dissimilarity and correlation functions. In: Artificial Intelligence, Springer, Cham, pp. 13-28. DOI: 10.10 07/978-3-030-33274-7_2.
4. **Chen, P. Y., Popovich, P. M. (2002).** Correlation: Parametric and nonparametric measures. Sage, Thousand Oaks, CA.
5. **Gibbons, J. D., Chakraborti, S. (2003).** Nonparametric statistical inference. 4th ed. Dekker, New York.
6. **Batyrshin, I. (2019).** Towards a general theory of similarity and association measures: similarity, dissimilarity, and correlation functions. Journal of Intelligent and Fuzzy Systems, Vol. 36, No. 4, pp. 2977–3004. DOI: 10.3233/JIFS-181503.
7. **Rudas, I. J., Batyrshin I.Z. (2023).** Explainable correlation of categorical data and bar charts. Recent Developments and the New Directions of Research, Foundations, and Applications. Springer Cham. vol. 1.
8. **Batyrshin, I. Z. (2021).** Contracting and involutive negations of probability distributions. Mathematics, Vol. 9, No. 19, p. 2389. DOI:10.3390/math9192389.

9. **Agresti, A. (2002).** Categorical data analysis. 2nd ed. John Wiley & Sons, Hoboken, New Jersey.
10. **Tang, W., He, H., Tu, X. M. (2012).** Applied categorical and count data analysis. CRC Press.
11. **Simonoff, J. S. (2003).** Analyzing categorical data. Springer, New York. Vol. 496.
12. **Azen, R., Walker, C. M. (2021).** Categorical data analysis for the behavioral and social sciences. 2nd ed. Routledge, New York. Pp. 296, DOI: 10.4324/97 80203843611.
13. **Irani, F., Barbone, J. M., Beausoleil, J., Gerald, L. (2017).** Is asthma associated with cognitive impairments? A metaanalytic review. Journal of clinical and experimental neuropsychology, Vol. 39, No. 10, pp. 965–978. DOI: 10.1080/13803395.20 17.1288802.
14. **Haq Satti, R. R. U., Rasheed, S. A., Gul, R., Athar, M.H. (2022).** Frequency of cognitive decline in asthma patients and associated socio-demographic factors. PAFMJ, Vol. 72, pp. S114– S117.
15. **Batyrshin, I. Z., Kubysheva, N. I., Bayrasheva, V. R., Kosheleva, O., Kreinovich, V. (2021).** Negations of probability distributions: A survey. Computación y Sistemas, Vol. 25, No. 4, pp. 775–781. DOI: 10.13053/cys-25-4-4094.
16. **Hancock, J.T., Khoshgoftaar, T. M. (2020).** Survey on categorical data for neural networks. Journal of Big Data, Vol. 7, No. 1, pp. 1–41. DOI: 10.1186/s40537-020-00305-w.
17. **Albright, S. C., Winston, W. L. (2019).** Business analytics: Data analysis & decision making. 7th ed, Cengage Learning.
18. **Camm, J., Cochran, J., Fry, M., Ohlmann, J., Anderson, D. (2019).** Business Analytics: descriptive, predictive, prescriptive. 3rd ed. Cengage Learning.
19. **Batyrshin, I.Z. (2022).** Fuzzy Distribution Sets. Computación y Sistemas, Vol. 26, No. 3, pp. 1411–1416. DOI: 10.13053/CyS-26-3-4360.

Article received on 08/07/2022; accepted on 25/10/2022.
Corresponding author is Ildar Z. Batyrshin.

ISSN 2007-9737

Distributed Geometric Multigrid Method: Analysis of a V Cycle Truncation Level Criteria

Matías Valdés, Sergio Nesmachnow

Universidad de la República,
Uruguay

{mvaldes,sergion}@fing.edu.uy

Abstract. This article presents the analysis of a V cycle truncation level criteria in a parallel implemention of a geometric multigrid method for solving partial differential equations, developed over a distributed memory system. The proposed system is implemented in C, using the Message Passing Interface library. A theoretical analysis of the proposed truncation level criteria is presented, and its evaluation is reported for the Poisson problem. The experimental analysis indicates that the proposed method achieves accurate speedup and computational efficiency, and shows a good scalability behavior to solve large problems by properly using more processing units.

Keywords. Multigrid, distributed memory, truncated V cycle, MPI.

1 Introduction

Multigrid methods are efficient numerical algorithms for solving partial differential equations by applying several discretization formulae, hierarchically organized [8, 16]. They are recognized within the most efficient techniques for solving partial differential Eqs. [16]. Furthermore, they can be expanded to employ higher level data structures to implement powerful multilevel methods, to address complex problems in various research domains by properly handling different matrix patterns, lattices, and other useful abstract mathematical structures for arranging the set of points that discretize a given problem domain.

Multigrid methods are also suitable for parallelism. A typical approach is to take advantage of the inherent parallel computations on the multiple grid components [10]. However, several important considerations must be taken into account to achieve a proper computational efficiency, including the sequential processing of elements in each grid level and the granularity of the parallel computations, which may be different for different levels.

One of the most useful applications of multigrid numerical methods is as solvers for elliptic partial differential equations. In fact, they are characterized as the fastest methods for elliptic problems [16]. One of the most notorious elliptic partial differential equation is Poisson equation. This equation usually appears as part of dynamical models for different physical phenomena. For example, it is related to the Navier-Stokes equations for incompressible flows, as it appears as a sub-problem in the discretization of these equations by using the MAC method, the projection method, and the fractional-step method [7]. Also, the incompressible Navier-Stokes equations may be reformulated into the Poisson pressure equation [9]. Poisson equation is also widely used as a test problem in numerical analysis and high performance computing [5].

In this line of work, this article proposes and analyzes the performance of a distributed memory implementation of a goemetric multigrid method, using truncated V cycles, and applied to a Poisson problem. The proposed approach consists in truncating each V cycle at the deepest possible level l that also guarantees that each processing unit is assigned a given constant number of vertices N. This truncation criteria was mentioned by Linden et al. [11], although it was not considered for V cycles.

doi: 10.13053/CyS-26-4-4438

We propose a theoretical analysis of the aforementioned truncation criteria, which characterizes the level reached in the truncated V-cycle, and its number of vertices, in terms of the problem size and number of processing units. We then present an experimental analysis of the computational performance of the implemented multigrid method on a distributed memory infrastructure.

The article is organized as follows. Section 2 describes the test problem and the main concepts about geometric multigrid methods. Section 3 presents a review of related works. Section 4 describes the proposed truncated V cycle criteria and the implementation details of the multigrid solver. The experimental evaluation is reported in Section 5. Finally, Section 6 presents the conclusions and formulates the main lines for future work.

2 Background and Theoretical Foundations

This section presents the test problem and the theoretical foundations for the proposed approach.

2.1 Test Problem: Poisson Equation

The Poisson problem was selected to test the performance of the implemented multigrid method. Poisson equation is an elliptic partial differential equation with several applications in science, which is commonly used for the evaluation of multigrid solvers (see the review of related works in Section 3).

The considered Poisson problem is defined on the unit square as domain, and Dirichlet boundary conditions are assumed, as expressed in Eq. 1. There, $\Delta u(x,y) = u_{xx}(x,y) + u_{yy}(x,y)$ is the Laplacian operator; $v(x,y)$ is a known *source* function; and $g(x,y)$ is also a known function that determines the boundary conditions. The Poisson equation is discretized by using centered finite differences, in a regular grid of $n+1$ vertices per dimension. The discretization generates a sparse linear system, with $(n-1)^2$ unknowns (one for each interior vertex) [2]. The resulting linear system is solved with multigrid to compute an estimation of $u(x,y)$ in the points of the grid:

$$\text{find } u : \Omega = [0,1]^2 \subseteq \mathbb{R}^2 \to \mathbb{R} \, / \quad \begin{cases} -\Delta u(x,y) = v(x,y), & \forall\,(x,y) \in \text{int}\,(\Omega)\,, \\ u(x,y) = g(x,y), & \forall\,(x,y) \in \partial\Omega. \end{cases} \tag{1}$$

2.2 Multigrid Methods

Multigrid methods work under the idea of accelerating an iterative solver by using information provided by a global correction procedure, which operates on coarse grids to solve a fine grid problem which in turn is easier to solve. The recursive process, called the multigrid cycle, is applied until a direct solver can be applied without additional computation cost on the coarsest grid. V-cycle is one of the most popular type of multigrid cycles, characterized by computing the coarse grid correction down levels (the *restriction phase*), until finding a level where the direct method is applied.

Numerical computations on steps performed in coarse grids are quicker, as few vertices are considered, and the numerical error converges faster as the method goes down levels [8]. Then, an interpolation is applied (the *prolongation* phase) to determine values on upper level grids, until the finer grid is again reached. Multigrid methods are recursive in nature and a V-cycle can be extended to any number of levels.

In this article, geometric multigrid, implemented with cycles of type V, is used as a solver for the discretized Poisson problem. The problem domain is first partitioned into a uniform grid, in which an initial estimation is find by applying few iterations from an iterative *smoother* method. The residual of this initial estimation is then restricted into a a coarser grid, where it is corrected by applying few iterations of the smoother.

This process is repeated by taking successively coarser grids. When the coarsest grid is reached, estimations from the different grid levels are combined by interpolation to the finer grids, until the initial grid is reached. This process is illustrated in Fig. 1, where h is the width of the initial grid, n is the number of unknowns in each dimension of this grid, and R and I represent the restriction and interpolation operations, respectively.

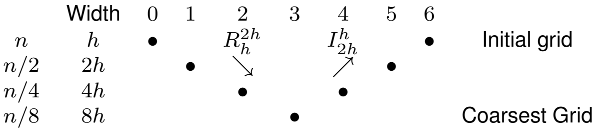


Fig. 1. A non-truncated V cycle for MG

Algorithm 1 Recursive multigrid V cycle: $V^h(A_h, r^h, u^0)$

```
Require: $h = 1/(m-1)$, $A_h \in \mathbb{R}^{m\times m}$, $r^h \in \mathbb{R}^{m\times 1}$, $\theta \in \mathbb{N}^+$
if actual grid is not the coarsest then
    $x^h$ = gs_rb($A_h, r^h, \theta$)    ▷ GS-RB from $u^0$
    $r^{2h} = R_h^{2h}(r^h - A_h x^h)$    ▷ Restrict to coarser grid
    $x^{2h} \leftarrow V^{2h}(A_{2h}, r^{2h}, u^0 = \vec{0})$    ▷ Recursion
    $x^h = x^h + I_{2h}^h(x^{2h})$    ▷ Interpolate and correct
    $x^h$ = gs_rb($A_h, r^h, \theta$)    ▷ GS-RB with $u^0 = x^h$
else
    $x^h \leftarrow$ sor($A_h, r^h$)    ▷ SOR until convergence
end if
return $x^h$
```

A recursive implementation for a multigrid V cycle is presented in Algorithm 1, following the idea by Briggs et al. [2]. The implementation uses Gauss-Seidel (GS) as smoother method. GS applies a Red-Black update order strategy and executes a fixed number of iterations θ. R_h^{2h} represents the restriction of the grid vertices to the next coarser grid; which is done by full-weighting of adjacent vertices. I_{2h}^h denotes the interpolation to the finer grid, which is performed by bi-linear interpolation of adjacent vertices. Finally, the Successive Over-Relaxation (SOR) method is applied as the coarse grid solver.

Gauss-Seidel Red-Black (GS-RB) is selected as smoother method, since it is a more parallelizable version of the traditional Gauss-Seidel solver. GS-RB is obtained by modifying the order in which grid vertices are updated. Vertices are first separated into two colors: red and black, intercalated. Then, they are updated with the usual GS expression, first applied to all red vertices, and then to the black ones [15]. This way, vertices of the same color may be updated concurrently. An important part of the smoother is the residual, as it is used as the independent term of the linear system solved by the multigrid method in each grid. In the case of GS-RB, the residual for black vertices is always null, as they are updated last. For the red vertices update, the residual coordinates associated to u^{k+1}, are given by Eq. 2 (for $i + j =$ even):

$$r_{i+j}^{k+1} = v_{i,j} + \frac{u_{i,j-1}^{k+1} + u_{i,j+1}^{k+1} - 4u_{i,j}^{k+1}}{h^2} + \frac{u_{i+1,j}^{k+1} + u_{i-1,j}^{k+1}}{h^2}. \quad (2)$$

When solving a discretized Poisson linear system, GS-RB requires $O(n^4)$ operations in order to estimate a solution satisfying $\|r^k\| < \epsilon\|r^0\|$, $\epsilon > 0$. More efficient methods are Conjugate Gradient or Successive Over-Relaxation (SOR); both with $O(n^3)$ operations [4]. For the proposed implementation, SOR was chosen as coarse grid solver, as it may be obtained easily from the GS implementation. Specifically, SOR with Red-Black order (SOR-RB) is used. The linear system of interest is symmetric and positive definite, which implies that SOR-RB is convergent for any $w \in (0, 2)$ [4]. The value of w was chosen to maximize the convergence speed, as $w_{\text{opt}} = 2/(1 + \sin(\pi h))$. The coordinates of SOR residual, associated to u^{k+1}, are given by Eq. 3 (valid for red and black vertices):

$$r_{i,j}^{k+1} = v_{i,j} + \frac{u_{i,j-1}^{k+1} + u_{i,j+1}^{k+1} - 4u_{i,j}^{k+1}}{h^2} + \frac{u_{i+1,j}^{k+1} + u_{i-1,j}^{k+1}}{h^2}. \quad (3)$$

Each multigrid iteration applies one V cycle, with initial condition taken as the previous V cycle final estimation: $u^{k+1} = V^h(A, b, u^k)$.

When developing a parallel implementation of a multigrid method in a distributed memory system, the number of vertices assigned to each processing unit decreases exponentially while descending in a V cycle. Thus, the cost of message passing may start to prevail over the

doi: 10.13053/CyS-26-4-4438

cost of computations, producing a degradation of the overall computational efficiency. A technique to overcome this problem is, instead of reaching the coarsest grid, using *truncated* V cycles [16], as illustrated in Fig. 2. When using a truncated V cycle, Algorithm 1 iterates until reaching the coarsest grid, which is determined by the truncation criteria.

The truncated V cycle strategy has a specific drawback: if the cycle is truncated too soon, the (truncated) coarse grid may have too many vertices, and solving the coarse grid linear system may become a bottleneck, affecting the overall computational efficiency of the solver. Thus, selecting an appropriate truncation level that takes into account the resulting trade-off is one of the most important challenges when implementing distributed memory multigrid methods [10].

3 Related Works

Sterk and Trobec [14] presented a parallel implementation of a multigrid method applied to solve the 3D Poisson equation, within a fluid flow simulation. The proposed algorithm was implemented in MPI and executed on a cluster of workstations. The authors performed several configuration experiments, including finding the grid size for switching from parallel to sequential execution. A comparison with a SOR method was reported. Both methods achieved similar sub-linear speedup values (i.e., below 0.8), but the results confirmed that the parallel multigrid method had a better scalability behavior than parallel SOR.

Gradl and U. Rüde [6] studied multigrid implemented over the Hierarchical Hybrid Grids framework for finite element problems and using the Metis mesh partitioning software. A specific algorithm design was proposed to lower the communication overheads, by reducing the number of messages exchanged between parallel processes.

The time per V cycle and the time to the overall solution were analyzed in weak scalability experiments. Results showed a good parallel scalability of the implemented multigrid method, but authors acknowledged that further improvement were needed for a full optimization of the communication patterns. The data structures used for calculation also allow applying adaptive mesh refinement methods, e.g., using hanging nodes or red-green refinement, to expand the applicability of the proposed solver.

Daley et al. [3] proposed two parallel mapping algorithms for addressing the scalability limitations of multigrid methods due to excessive communication costs in the coarser grids. Improved communication algorithms were conceived to map the mesh back and forth between an uniform grid and an adaptive mesh refinement procedure. The performance of the proposed parallel mapping algorithms was analyzed for a case study involving a multigrid Poisson solver using octrees block structures in FLASH, a multiphysics/multiscale method for flows simulations.

Experiments were performed in a Virtual Node with 4 MPI tasks per node and 512 MB of memory per MPI task, on a IBM BG/P platform. Results indicated that the proposed implementation allowed obtaining an increase on performance when increasing the number of computing resources, depending on the level used for switching from a refined mesh in the octree to a uniform grid (the higher the refinement level, the better the performance gain).

Müller and Scheichl [12] studied the scalability of numerical methods for solving elliptic partial differential equations for atmospheric fluid dynamics. Solvers based on Krylov subspaces and multigrid algorithms were categorized as the most efficient methods, after an experimental evaluation performed in the national supercomputer from the UK. Several algorithms were evaluated, already implemented in two well known libraries of routines for scalable parallel linear systems resolution: Distributed and Unified Numerics Environment (DUNE) and the Parallel High Performance Preconditioners (hypre).

In turn, two custom implementations were developed: a Conjugate Gradient solver using vertical line relaxation preconditioner and a tensor-product geometric multigrid algorithm. Results demonstrated that the overall computational efficiency of the tensor-product geometric multigrid method was better

	Width	0	1	2	3	4	5	6	
n	h	●		R_h^{2h}		I_{2h}^h		●	Initial grid
$n/2$	$2h$		●	$\searrow$		$\nearrow$	●		
$n/4$	$4h$			●	$\rightarrow$	●			Coarse grid
$n/8$	$8h$				○				Coarsest Grid

Fig. 2. A truncated V cycle for MG

that standard algebraic multigrid methods. Furthermore, the implemented multigrid solver was more robust than one-level methods when considering parameter variations.

The scalability of multigrid solvers available in hypre was also studied by Baker et al. [1]. Several multigrid algorithms were compared, including PFMG, SMG, SysPFMG, BoomerAMG, and AMS, to solve three benchmark problems: a 3D Laplace (i.e., Poisson with $v(x,y) = 0$) equation with Dirichlet boundary conditions, a system of two 3D Laplace equations with weak inter-variable coupling, and a simple 3D electromagnetic diffusion problem. The main results demonstrated the usefulness of considering assumed partition instead of global partitions, and both a distributed memory implementation using MPI and a hybrid distributed/shared memory MPI/OpenMP implementation were able to achieve accurate scalability values when solving large problem instances.

The truncation criterion proposed in this article is aimed to improve load balancing, by truncating each V cycle at the deepest possible level l that also guarantees that each processing unit processes a given constant number of vertices N. This criterion was suggested by Linden et al. [11], in their analysis of scalability aspects of parallel multigrid, but instead of using V cycles, they implemented a full multigrid method with W cycles in the intermediate grids. Experimental results indicated that the (truncated) coarsest grid calculations are determinant for scalability performance.

In a more recent article, Dexuan and Ridgway analyzed the convergence and efficiency of multigrid methods with truncated V-cycle, which they call U-cycle [17]. However, authors did not propose a specific criteria for selecting the coarsest grid, which was generally defined as "fine enough so that all processors are productively busy in doing the coarse-grid solver".

4 The Proposed Parallel Geometric Multigrid Solver

This section describes the proposed parallel multigrid method over distributed memory systems.

4.1 Truncated V Cycle: Criteria and Properties

The proposed method is based on choosing a truncation level which is deep enough to have few vertices per processing unit, but not as few as to have idle processing units. The formal definition is presented next.

Definition 1 (Truncation criteria)**.** *Consider an initial grid with $n = q \times 2^r$ vertices per dimension, where q is odd. In this case, each V cycle has a maximum of r levels. Given $N \geq 1$ and p processing units, truncate the V cycle at the deepest possible level $l \leq r$, which also satisfies: $n_l^2/p \geq N$, where $n_l = n/2^l$ is the number of vertices per dimension in the (truncated) coarse grid.*

Algorithm 2 presents a pseudocode for the proposed truncation criteria. Two conditions (line 3) control the iterative descending procedure. When using this criteria with an ideal load balancing, no processing unit is idle at the coarsest (truncated) grid. To compensate for some unbalance, the value of N should be chosen greater than one.

The level reached by the proposed criteria depends on the initial size n, the number of processing units p, and the value chosen for N. Theorem 1 characterizes this dependence.

Algorithm 2 Proposed criteria for truncating a V cycle of geometric multigrid

```
Require: $n = q \times 2^r$, $q$ odd, $r \geq 0$, $r \in \mathbb{N}$, $N \geq 1$, $N \in \mathbb{N}$
1: $l = 0$                                  ▷ Initial level of $V$ cycle
2: $n_l = n$                                ▷ Vertices per node at level $l$
3: while $l < r$ and $(n_l/2)^2 \geq Np$ do
4:     $n_{l+1} = n_l/2$
5:     $l = l + 1$
6: end while
7: return $l$, $n_l$
```

Theorem 1. *Let assume that the initial number of vertices per dimension is $n = q \times 2^r$, q odd, $r \in \mathbb{N}$, $r \geq 0$. Then, the level l of the truncated V cycle reached with the proposed criteria, is (for $p \in \mathbb{N}$, $p \geq 1$):*

$$l = \begin{cases} 0, & \text{if } \frac{n}{\sqrt{Np}} < 2 \\ \lfloor \log_2 \left(\frac{n}{\sqrt{Np}} \right) \rfloor \in [1, r-1], & \text{if } 2 \leq \frac{n}{\sqrt{Np}} < 2^r \\ r, & \text{if } \frac{n}{\sqrt{Np}} \geq 2^r. \end{cases}$$

Proof.

1. If $n/\sqrt{Np} < 2$, the second iteration condition (line 3 in Algorithm 2) is not satisfied at level $l = 0$, and the method does not descend any level.

2. To reach a given level $1 \leq l \leq r$, both iteration conditions must be satisfied at level $l - 1$. The first condition holds, as $l - 1 < r$. The second condition is:

$$n_{l-1}^2 = \left(\frac{n}{2^{l-1}} \right)^2 \geq 4Np \Leftrightarrow \frac{n}{2\sqrt{Np}} \geq 2^{l-1}$$

$$\Leftrightarrow \frac{n}{\sqrt{Np}} \geq 2^l.$$

   The reached level l is the last level, whenever the first or second stopping condition is satisfied at this new level. That is: if $l = r$, or:

$$n_l^2 = \left(n/2^l\right)^2 < 4Np \Leftrightarrow \frac{n}{2\sqrt{Np}} < 2^l.$$

3. Thus, level r is reached, if and only if:

$$\frac{n}{\sqrt{Np}} \geq 2^r.$$

4. In the rest of the cases, level l is reached, with $1 \leq l < r$. For those cases:

$$\frac{n}{2\sqrt{Np}} < 2^l \leq \frac{n}{\sqrt{Np}} \Leftrightarrow \log_2 \left(\frac{n}{2\sqrt{Np}} \right) < l$$

$$\leq \log_2 \left(\frac{n}{\sqrt{Np}} \right) = 1 + \log_2 \left(\frac{n}{2\sqrt{Np}} \right)$$

$$\Leftrightarrow l = \lfloor 1 + \log_2 \left(\frac{n}{2\sqrt{Np}} \right) \rfloor = \lfloor \log_2 \left(\frac{n}{\sqrt{Np}} \right) \rfloor.$$

□

Example 1. *Consider $n = 10240 = 5 \times 2^{11} = q \times 2^r$ and $p = 64 = 2^6$. Without truncating the V cycle, the deepest possible level is $l = 11$; resulting in a coarse grid with $n_c = 5$ vertices per dimension, and $n_c^2/p = 25/64 < 1$ vertices per processing unit (for an ideal load balance). Now consider truncating each cycle with the proposed criteria, using $N = 4$. In this case: $n/\sqrt{Np} = 5 \times 2^7 < 2^{11}$. Therefore, by Theorem 1, the value of l is given by Eq. 4:*

$$l = \lfloor \log_2 \left(\frac{n}{\sqrt{Np}} \right) \rfloor = \lfloor \log_2 \left(5 \times 2^7 \right) \rfloor = \lfloor \log_2 (640) \rfloor = \lfloor 9.3219 \rfloor = 9. \quad (4)$$

The number of vertices per dimension at this level is $n_l = n/2^9 = 20$. Assuming an ideal load balance, the number of vertices per processing unit is $n_l^2/p = 6.25$. Fig. 3 shows the reached level l, coarse grid vertices per dimension n_l, and per processing unit n_l^2/p, for different values of p and N, as given by Theorem 1. Corollary 1 bounds the number of vertices in the coarse grid, independent from n and p.

Corollary 1. *For a truncated coarse grid reached at level $l \in \{1, ..., r-1\}$, the total number of vertices is bounded by $Np \leq n_c^2 < 4Np$, and the number of vertices per processing unit is bounded by $N \leq n_c^2/p < 4N$ (for ideal load balance).*

Proof. From the proof of Theorem 1: $n/2\sqrt{Np} < 2^l \leq n/\sqrt{Np} \Leftrightarrow \sqrt{Np} \leq n_c = n/2^l < 2\sqrt{Np} \Leftrightarrow Np \leq n_c^2 < 4Np$. □

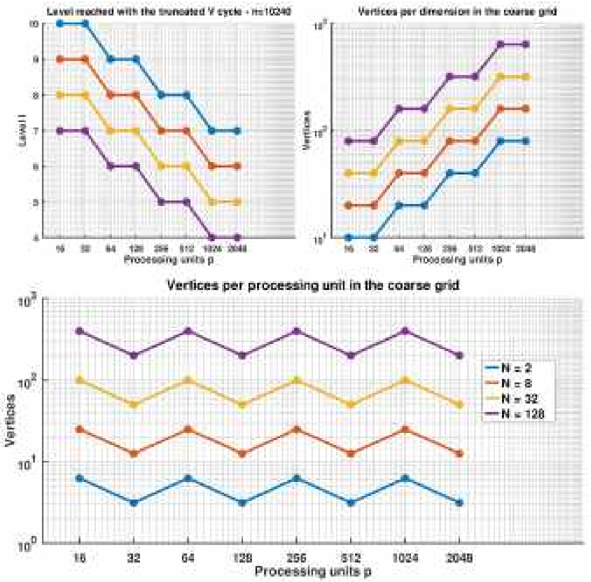


Fig. 3. Truncated V cycle for $n = 5 \times 2^{11}$, when using the proposed criteria: level reached (l), coarse grid vertices per dimension (n_l), and per processing unit n_l^2/p

4.2 Design and Implementation Details

This subsection presents the design considerations and implementation details of the proposed method.

4.2.1 Design Considerations

The discretization of the Poisson problem results in a square linear system with $(n-1)^2$ unknowns, $n = 1/h$. The solution of the system approximates the exact solution $u(x, y)$ in the selected grid vertices. Thus, to obtain a good spatial resolution, the value of n must be taken sufficiently large. This decision imposes two practical limitations: (1) the calculation time required to process an increasing number of vertices, and (2) the required memory to store the partial and final estimations of each vertex. A distributed memory parallel implementation is applied to overcome these practical limitations.

4.2.2 Domain Decomposition and Processing

The decomposition approach is based on dividing the domain grid into rectangular sub-domains, to be assigned to each of the available processing units. A Cartesian grid is created and the optimal number of processing units per dimension (N_x and N_y) is determined using the `MPI_Dims_create` function of MPI. The output information is then used to create a new MPI communicator with Cartesian topology, by using the `MPI_Cart_create` function.

The grid vertices are then distributed along the processing units. Let (q_x, q_y) the coordinates of a generic processing unit q in the Cartesian communicator, processing unit q receives the inner grid vertices with coordinates in the two intervals I_x^q and I_y^q, as defined in Eqs. 5 and 6, where the integer division operator is used:

$$I_x^q = \left[\frac{q_x(n-1)}{N_x} + 1, \; \frac{(q_x+1)(n-1)}{N_x} \right], \tag{5}$$

$$I_y^q = \left[\frac{q_y(n-1)}{N_y} + 1, \; \frac{(q_y+1)(n-1)}{N_y} \right]. \tag{6}$$

Once the data is partitioned, each processing unit applies the same MPI multigrid code to its own assigned vertices. The numerical resolution is performed in a coordinate way with other processing units, by exchanging *halo layers* of *ghost vertices*, as explained in the next subsection. Halo layers act as boundary values for each sub-domain, and contain values estimated by adjacent processing units. Fig. 4 illustrates an example of the domain decomposition.

4.2.3 Communications

Halo layers are exchanged by message passing between neighboring processing units, i.e., those adjacent to each other in the defined Cartesian communicator. The `MPI_Cart_shift` function is used to determine neighbors. Each processing unit updates its halo layers after receiving data from its neighbors, and then sends its updated vertices to act as halo layers for its neighbors. Message passing is done using the combined exchange mechanism implemented in the `MPI_Sendrecv`

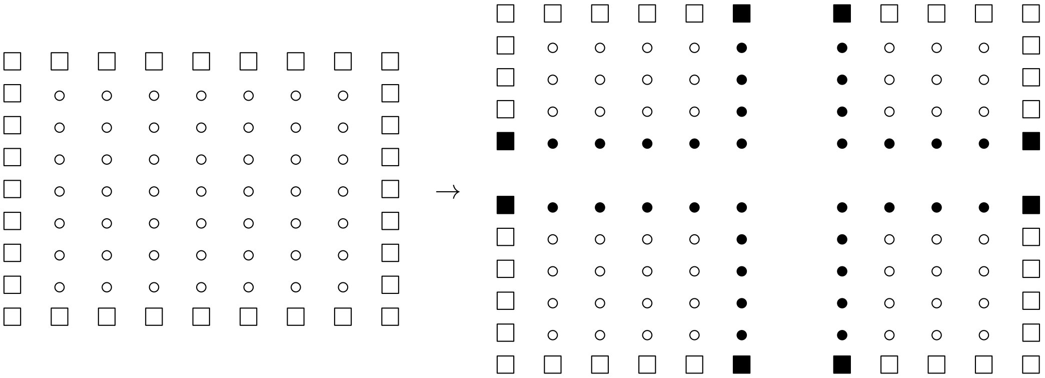


Fig. 4. Example of domain decomposition into four sub-domains, with an overlap region (halo layers) of width one. Inner vertices (white circles), boundary vertices (white squares), and ghost vertices (black circles/squares) are distinguished

function, which has the specific advantage of avoiding deadlocks between processes.

In order to improve the efficiency of the halo layers exchange, all the information is packed into MPI datatypes. Halo layers corresponding to a grid row are grouped into a contiguous datatype using the `MPI_Type_contiguous` function. In turn, for packing column halo layers, which are not contiguous, a vector datatype is created by using the `MPI_Type_vector` function. The exchange of halo layers is performed in the following stages: at every iteration of GS-RB and SOR-RB, after updating black or red vertices; after calculating the final residual vector of each GS-RB call; at the end of each restriction or interpolation operation.

Besides exchanging halo layers, a reduction operation (using `MPI_Reduce`) is applied: (1) to compute the residual, needed for the stopping criteria, after each multigrid cycle or SOR-RB iteration; (2) to compute the norm of the independent term at the beginning of the multigrid main iteration; and (3) to determine the overall execution time, as the maximum execution time of all processes.

4.2.4 MPI-IO Storage

During the multigrid execution, each processing unit stores in local memory the estimated values for its assigned vertices. In the developed implementation, the final estimation is stored concurrently into a unique binary file, by using MPI-IO. Each processing unit writes the solution of its sub-domain to the unique binary file using function `MPI_File_write_all`. The time spent in I/O operations is not included in the overall execution time when analyzing the performance of the proposed multigrid implementation, since it heavily depends on the capabilities (technology, transfer speed) of the hard disk. In turn, omitting the I/O times allows providing a baseline for comparison with other proposals.

4.2.5 Time Measurement

Execution time is measured as the difference between two calls of functions `time` (sequential) or `MPI_Wtime` (parallel). The parallel time is the maximum of all times of processing units, obtained with an `MPI_Reduce`() directive. The following initial steps are included in the overall execution time: determining the optimal number of processing units per grid dimension, creating a Cartesian communicator, assigning vertices to the processing unit, and determining neighboring processing units.

5 Experimental Evaluation

This section describes the experimental evaluation of the proposed truncation level criteria and discusses the main results.

5.1 Development and Execution Platform

The proposed multigrid method was implemented in the C programming language, using the MPICH implementation of the MPI standard (`www.mpich.org`). All floating point calculations use double precision.

The experimental evaluation was performed on HP ProLiant DL380 G9 servers with two Xeon Gold 6138 processors with 20 cores at 2.00 GHz each, and 128 GB of RAM, connected by Ethernet 10 Gbps, and running the Linux CentOS 7 OS, from National Supercomputing Center (Cluster-UY), Uruguay [13].

5.2 Test Problem, Convergence Criterion and Parameters

Tests were performed considering the Poisson problem defined in Eq. 1, whose exact solution given by Eq. 7:

$$u(x,y) = e^{-\left((x-\frac{1}{2})^2+(y-\frac{1}{2})^2\right)} + \frac{5}{100}\left(\sin(10\pi x) + \sin(4\pi y)\right). \quad (7)$$

The source function is taken as $v(x,y) = -\Delta u(x,y)$, and boundary values $g(x,y)$ are obtained by restricting the exact solution to the domain boundary.

Regarding the convergence criterion, SOR-RB stops execution when the residual of the corresponding linear system at step k satisfies $\|r_k\|_2/\|r_0\|_2 \leq 10^{-6}$. The same criteria is used for the multigrid main iteration, but with b instead of r_0. For multigrid the maximum number of iterations is set to 15 V cycles. For SOR-RB this is set to $5n_c$ iterations; where n_c is the number of vertices per dimension in the coarse grid. The truncation criteria uses $N = 5$.

The first multigrid iteration (first V cycle) uses $u^0 = \vec{0}$. For subsequent cycles, the initial estimation is taken as the final estimation of the previous cycle ("warm restart"). The GS-RB smoother does $\theta = 2$ iterations before descending to a coarser grid (pre-smooth), and after ascending to a finer grid (post-smooth).

5.3 Execution Time, Speedup and Efficiency

The *algorithmic speedup* metric is applied to analyzed the performance of the proposed method. Five independent executions were performed to reduce bias due to the utilization of a non-dedicated hardware platform. The chosen values for n are of the form $q \times 2^r$, with $q \in \{1,3,5\}$.

Table 1 reports the average execution times of the proposed method ($\overline{T_p}$), with their corresponding coefficient of variation ($\hat{v}$(%)), using an increasing number of computing units (p, up to 70) and cluster nodes (c), for different problem sizes (n from 8192 to 20480).

The execution time results reported in Table 1 Results demonstrate that the proposed implementation was able to steadily reduce the execution times of the multigrid method, as p increases. In particular, for $n = 20480$, the execution time reduced from 781 seconds for the sequential implementation to 14.0 seconds with $p = 70$. Despite using a non dedicated cluster, a robust behavior was observed in the execution times, with coefficients of variation less than 5% for most cases.

Table 2 reports the corresponding algorithmic speedup values (S_p) for the execution times reported in Table 1. A graphical analysis of speedup values is presented in Fig. 5.

The efficiency results in Table 2 indicate that the best speedup ($S_{70} = 55.6$) was obtained for the two largest problems. Constant speedup intervals are observed at different values of p; particularly for $[48, 50]$ and $[56, 60]$, possibly explained by the transition from 2 to 3 computing nodes, and the corresponding increase in communication latencies. In general, speedup is always increasing for the selected range of processing units, which may indicate that communications do not prevail over computations, which is the main objective of the proposed truncation level criteria.

Table 3 reports the computational efficiency of the proposed method, defined by the normalized value of the speedup when using p computing resources (Equation 8):

$$E_p = \frac{S_p}{p}. \quad (8)$$

Table 1. Average execution time ($\bar{T}_p$ seconds) and coefficient of variation ($\hat{v}$(%)) for different problem size (n) and number of computing resources (p, c)

$p\,(c)$	**problem size** (n)													
	8192		**10240**		**12288**		**14336**		**16384**		**18432**		**20480**	
	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)	$\overline{\mathbf{T_p}}$	$\hat{\mathbf{v}}$(%)
1 (1)	91.6	0.53	142	0.35	206	0.84	276	0.53	380	7.70	549	0.59	781	0.48
2 (1)	50.8	0.13	79.0	0.35	115	0.87	154	0.09	204	0.93	256	0.64	360	1.05
4 (1)	25.8	0.45	40.5	1.08	58.0	1.41	78.3	0.27	103	0.46	130	0.16	182	0.39
8 (1)	15.8	0.90	24.4	2.35	34.9	1.97	47.6	2.36	65.2	3.51	77.2	1.85	109	0.67
12 (1)	8.74	1.00	13.8	3.60	20.2	5.57	26.5	0.20	35.2	1.22	44.0	0.54	61.9	0.42
16 (1)	8.14	1.20	12.8	0.65	18.4	1.02	24.9	0.99	32.4	0.46	41.0	0.51	57.4	0.53
20 (1)	6.33	0.57	10.1	1.34	14.5	0.17	19.9	1.73	26.6	4.60	33.2	0.31	46.1	0.75
24 (2)	5.27	0.35	8.23	0.37	12.0	0.31	16.0	0.64	20.9	0.48	26.7	0.89	36.9	3.05
28 (2)	5.10	11.9	8.45	14.1	11.8	17.1	16.0	9.16	19.3	6.43	24.0	4.08	32.6	0.52
32 (1)	4.21	1.38	6.64	2.76	9.47	0.70	12.9	3.10	16.8	0.56	21.2	0.60	30.1	0.42
36 (1)	3.64	0.86	5.71	0.46	8.43	0.38	11.3	0.41	14.7	0.57	19.3	1.57	26.4	0.77
40 (2)	3.41	0.30	5.23	0.53	7.59	0.82	10.1	0.31	13.3	0.69	17.2	4.95	27.0	25.6
44 (2)	3.14	1.32	4.84	0.83	6.91	0.27	9.34	0.40	12.0	0.42	15.2	0.43	24.0	22.8
48 (2)	2.86	5.03	4.30	0.66	6.20	0.54	8.37	0.88	10.9	0.66	13.6	0.50	19.2	0.88
50 (3)	2.79	0.39	4.33	0.94	6.49	12.3	8.24	0.82	10.6	0.36	13.6	1.13	19.5	5.52
56 (2)	2.48	1.09	3.73	1.21	5.33	0.63	7.17	0.59	9.44	0.60	11.8	0.96	16.5	0.76
60 (3)	2.40	1.05	3.63	0.41	5.12	0.67	7.15	6.03	8.97	0.44	13.2	11.8	15.9	0.68
64 (2)	2.22	0.88	3.32	0.69	4.89	1.38	6.47	0.73	8.63	1.15	10.7	0.47	15.5	3.78
70 (2)	2.14	4.09	3.40	13.5	4.53	2.09	6.00	0.36	7.82	0.42	9.89	0.53	14.0	0.46

Table 2. Algorithmic speedup values for the execution times reported in Table 1

n	**computing resources** (p)																	
	2	**4**	**8**	**12**	**16**	**20**	**24**	**28**	**32**	**36**	**40**	**44**	**48**	**50**	**56**	**60**	**64**	**70**
8192	1.8	3.6	5.8	10.5	11.2	14.5	17.4	18.0	21.8	25.2	26.9	29.2	32.0	32.8	37.0	38.1	41.3	42.7
10240	1.8	3.5	5.8	10.2	11.1	14.0	17.2	16.8	21.3	24.8	27.1	29.3	33.0	32.7	38.0	39.0	42.7	41.6
12288	1.8	3.6	5.9	10.2	11.2	14.2	17.2	17.4	21.8	24.5	27.2	29.9	33.3	31.8	38.7	40.3	42.2	45.6
14336	1.8	3.5	5.8	10.4	11.1	13.9	17.2	17.3	21.4	24.6	27.4	29.6	33.0	33.5	38.5	38.6	42.7	46.0
16384	1.9	3.7	5.8	10.8	11.7	14.3	18.2	19.6	22.6	25.8	28.5	31.6	34.9	35.7	40.2	42.3	44.0	48.6
18432	2.1	4.2	7.1	12.5	13.4	16.6	20.6	22.9	26.0	28.5	31.9	36.1	40.3	40.4	46.8	41.7	51.2	55.6
20480	2.2	4.3	7.2	12.6	13.6	16.9	21.2	23.9	25.9	29.5	28.9	32.5	40.6	40.0	47.4	49.1	50.4	55.6

The computational efficiency values ranged from 0.9 to 0.6. From $p = 16$ to $p = 70$, efficiency remained almost constant, in the order of 0.8 for the two largest problems, and 0.7 for the rest.

The best efficiency was 0.88, obtained for $n = 20480$, when using $p = 24$ in a two nodes environment.

A graphical analysis of efficiency values is presented in Fig. 6.

5.4 Scalability Analysis

A scalability analysis was performed to determine the capability of solving problems in rather similar execution times.

The setup for the analysis consisted in increasing the number of processing units p, together with the problem size n, whereas keeping constant the initial number of vertices per processing unit.

Table 3. Algorithmic efficiency for speedup values of Table 2 ($E_p = 100 \times S_p/p$)

n	**computing resources** (p)																	
	2	**4**	**8**	**12**	**16**	**20**	**24**	**28**	**32**	**36**	**40**	**44**	**48**	**50**	**56**	**60**	**64**	**70**
8192	90.0	88.8	72.6	87.3	70.3	72.3	72.4	64.1	68.0	69.9	67.2	66.4	66.7	65.6	66.0	63.6	64.6	61.0
10240	89.5	87.5	72.5	85.3	69.3	69.9	71.7	59.8	66.7	68.8	67.7	66.5	68.6	65.4	67.9	65.0	66.7	59.4
12288	90.0	89.0	73.9	85.2	70.3	71.2	71.9	62.3	68.1	68.1	68.0	67.9	69.4	63.6	69.1	67.2	65.9	65.1
14336	89.5	88.2	72.5	86.8	69.4	69.6	71.8	61.7	67.0	68.2	68.6	67.2	68.8	67.0	68.8	64.3	66.7	65.7
16384	93.0	92.0	72.9	89.9	73.3	71.5	75.8	70.2	70.5	71.6	71.2	71.7	72.7	71.4	71.8	70.6	68.7	69.4
18432	107	106	88.9	104	83.8	82.8	85.6	81.7	81.1	79.1	79.8	82.0	83.9	80.8	83.5	69.5	80.0	79.4
20480	109	107	89.4	105	85.0	84.6	88.2	85.4	81.0	82.0	72.3	73.8	84.5	79.9	84.6	81.8	78.7	79.5

Table 4. Average execution time ($\bar{T_p}$ in seconds) and its coefficient of variation ($\hat{v}$(%)), for different number of computing resources (p, c), and problem size (n)

	problem size (n)							
	4096	**8192**	**10240**	**12288**	**16384**	**20480**	**24576**	**32768**
$p\,(c)$	1 (1)	4 (1)	6 (1)	9 (1)	16 (2)	25 (2)	36 (3)	64 (4)
$\overline{T_p}$	24.9	26.4	28.8	24.7	28.0	37.7	37.7	38.1
$\hat{v}$(%)	0.00	0.01	0.01	0.01	0.00	0.01	0.01	0.00

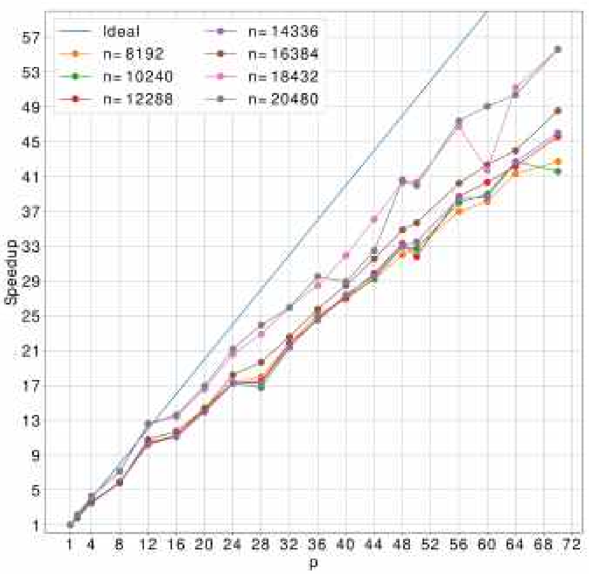


Fig. 5. Algorithmic speedup of the multigrid method with truncated V cycle ($N = 5$)

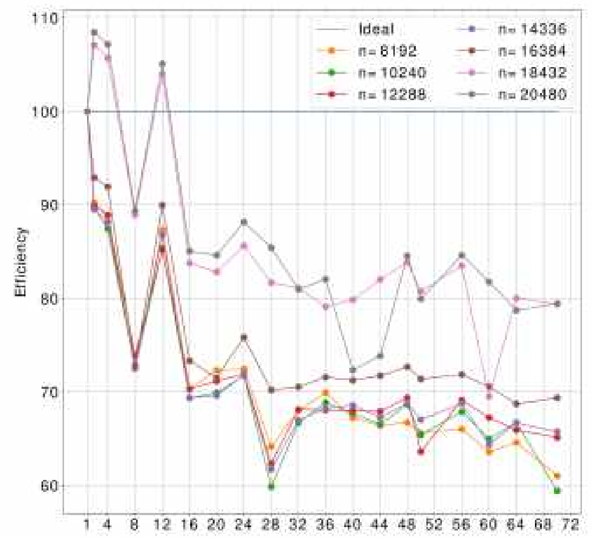


Fig. 6. Efficiency of the proposed multigrid algorithm with truncated V cycle ($N = 5$)

The value $n^2/p = 4096^2$ was chosen to have $p = 1$ for the smallest problem size, to fit in 1GB of RAM. Values of n are of the form $q \times 2^r$, $q \in \{1, 3\}$.

Table 4 reports the average execution times and coefficients of variation of the proposed multigrid implementation. Results demonstrate a good scalability behavior of the proposed implementation.

The execution time only increased 53% when the problem size n increased in a factor of 8 (from 4096 to 32768). Also, the execution time remained constant for n in 4096 to 12288 and 20480 to 32768.

6 Conclusions and Future Work

This article presented a truncating strategy for the V cycle of a parallel geometric multgird method. A theoretical analysis computed an explicit expression of the coarse grid level reached, and a bound for the number of vertices in the coarse grid, both in terms of the problem size and number of processing units.

The multigrid method with the proposed truncation criteria was implemented and evaluated in a distributed memory, non-dedicated cluster, for a Poisson problem. Accurate speedup and efficiency values were obtained for different problem sizes. Also, a weak scalability analysis revealed that problems of very different size may be solved in similar times by increasing the processing units.

The main lines for future work are related to analyze the performance with other coarse grid solvers, e.g., Conjugate Gradient or Chebyshev preconditioned with SSOR, which has fewer operations than SOR and CG. However, it uses two SOR steps per iteration, implying more communications. Thus, it may be interesting to analyze the computation-communication trade-off.

References

1. **Baker, A. H., Falgout, R. D., Kolev, T. V., Yang, U. M. (2012).** Scaling hypre's multigrid solvers to 100,000 cores. High-Performance Scientific Computing, Springer, pp. 261–279. DOI: 10.1007/978-1-4471-2437-5_13.

2. **Briggs, W. L., Henson, V. E., McCormick, S. F. (2000).** A multigrid tutorial. Society for Industrial and Applied Mathematics.

3. **Daley, C., Vanella, M., Dubey, A., Weide, K., Balaras, E. (2012).** Optimization of multigrid based elliptic solver for large scale simulations in the FLASH code. Concurrency and Computation: Practice and Experience, Vol. 24, No. 18, pp. 2346–2361. DOI: 10.1002/cpe.2821.

4. **Demmel, J. W. (1997).** Applied numerical linear algebra. Society for Industrial and Applied Mathematics.

5. **Dongarra, J., Heroux, M. A., Luszczek, P. (2015).** HPCG benchmark: a new metric for ranking high performance computing systems. Knoxville, Tennessee, pp. 42.

6. **Gradl, T., Rüde, U. (2008).** High performance multigrid on current large scale parallel computer. 9^{th} Workshop on Parallel Systems and Algorithm–workshop of the GI/ITG special interest groups PARS and PARVA, pp. 37–45.

7. **Guermond, J. L., Minev, P., Shen, J. (2006).** An overview of projection methods for incompressible flows. Computer Methods in Applied Mechanics and Engineering, Vol. 195, No. 44–47, pp. 6011–6045. DOI: 10.1016/j.cma.2005.10.010.

8. **Hackbusch, W., Trottenberg, U. (1982).** Multigrid methods: Proceedings of the conference held at köln-porz. Lecture Notes in Mathematics, Springer, Vol. 960 , pp. 23–27.

9. **Henshaw, W. D. (1994).** A fourth-order accurate method for the incompressible navier-stokes equations on overlapping grids. Journal of Computational Physics, Vol. 113, No. 1, pp. 13–25. DOI: 10.1006/jcph.1994.1114.

10. **Hülsemann, F., Kowarschik, M., Mohr, M., Rüde, U. (2006).** Parallel geometric multigrid. Numerical Solution of Partial Differential Equations on Parallel Computers, Springer, Vol. 51, pp. 165–208. DOI: 10.1007/3-540-31619-1_5.

11. **Linden, J., Lonsdale, G., Ritzdorf, H., Schüller, A. (1994).** Scalability aspects of parallel multigrid. Future Generation Computer Systems, Vol. 10, No. 4, pp. 429–439. DOI: 10.1016/0167-739X(94)90007-8.

12. **Müller, E. H., Scheichl, R. (2014).** Massively parallel solvers for elliptic partial differential equations in numerical weather and climate

prediction. Quarterly Journal of the Royal Meteorological Society, Vol. 140, No. 685, pp. 2608–2624. DOI: 10.1002/qj.2327.

13. **Nesmachnow, S., Iturriaga, S. (2019).** Cluster-UY: Collaborative scientific high performance computing in Uruguay. International Conference on Supercomputing in Mexico, Springer, Vol. 1151, pp. 188–202. DOI: 10.1007/978-3-030-38043-4_16.

14. **Sterk, M., Trobec, R. (2003).** Parallel performances of a multigrid Poisson solver. Parallel and Distributed Computing, International Symposium on, pp. 238–238. DOI: 10.1109/ISPDC.2003.1267669.

15. **Strang, G. (2007).** Computational science and engineering. Wellesley-Cambridge Press.

16. **Trottenberg, U., Oosterlee, C., Schüller, A. (2001).** Multigrid. Academic Press, an Elsevier Science Imprint.

17. **Xie, D., Scott, L. (2009).** An analysis of parallel U-cycle multigrid method.

Article received on 02/05/2022; accepted on 03/10/2022.
Corresponding author is Matías Valdés.

ISSN 2007-9737

Parallel Performance and I/O Profiling of HPC RNA-Seq Applications

Lucas Cruz[1,2], Micaella Coelho[1], Marcelo Galheigo[1], Andre Carneiro[1], Diego Carvalho[2],
Luiz Gadelha[1], Francieli Boito[3], Philippe Navaux[4], Carla Osthoff[1], Kary Ocaña[1]

[1] National Laboratory of Scientific Computing,
Brazil

[2] Federal Center for Technological Education Celso Suckow da Fonseca,
Brazil

[3] University of Bordeaux,
CNRS, INP, INRIA, LaBRI, UMR,
France

[4] Federal University of Rio Grande do Sul,
Informatics Institute,
Brazil

{lucruz, micaella, galheigo, andrerc, lgadelha, osthoff, karyann}@lncc.br, d.carvalho@ieee.org,
francieli.zanon-boito@u-bordeaux.fr, navaux@inf.ufrgs.br

Abstract. Transcriptomics experiments are often expressed as scientific workflows and benefit from high-performance computing environments. In these environments, workflow management systems can allow handling independent or communicating tasks across nodes, which may be heterogeneous. Specifically, transcriptomics workflows may treat large volumes of data. ParslRNA-Seq is a workflow for analyzing RNA-Seq experiments, which efficiently manages the estimation of differential gene expression levels from raw sequencing reads and can be executed in varied computational environments, ranging from personal computers to high-performance computing environments with parallel scripting library Parsl. In this work, we aim to investigate CPU and I/O metrics critical for improving the efficiency and resilience of current and upcoming RNA-Seq workflows. Based on the resulting profiling of CPU and I/O data collection, we demonstrate that we can correctly identify anomalies of transcriptomics workflow performance that is an essential resource to optimize its use of high-performance computing systems.

Keywords. Supercomputing, sorkflow, RNA-seq.

1 Introduction

In recent years, a deluge of large-scale transcriptomics data from high-throughput sequencing is increasingly raising the demand in computing power and storage. Processing this enormous amount of data requires the use of specialized techniques such as scientific workflow management systems (SWfMSs) and high-performance computing (HPC) resources to extract knowledge from data-intensive RNA-seq experiments [1, 9].

Scientific workflows deal with automating the execution of computational tasks and are needed for improving reproducibility and productivity. They have been used by scientists in a wide variety of domains, including astronomy, bioinformatics, physics, biology, biodiversity, among many others. Scaling workflows on large HPC systems is not an easy task due to the size and nature of data to be processed, the inherent complexity of workflows,

doi: 10.13053/CyS-26-4-4437

the number of workflow instances to be executed, and the complexity of large HPC systems [9].

We identified the following capabilities of SWfMSs relevant to transcriptomics analysis: workflow modularity and automated elasticity to enable checkpointing; scalability concerning the use of the number of workflow tasks versus the number of nodes per run; robustness and fault tolerance due to data issues, resource unavailability, or aborted executions; reproducibility via data provenance recording; portability across computing environments from desktops to parallel and distributed clusters; interoperability of metadata and the use in the same workflow into several programming languages; and ease of development by users with different skill levels in informatics.

In this paper, we present how a collection of performance metrics of well-established transcriptomics software can be orchestrated to optimize the performance behavior of current scientific RNA-Seq workflows. Due to the massive amount of scientific transcriptomics data, the complexity of scientific applications, and the features of distributed computing, the performance analyses require data metrics, information of the workflow execution, and to understand the environmental system as a whole. This also can include capturing the input and experimental provenance data, optimizing the workflow structure, and gathering and storing performance information such as CPU and memory usage and I/O operations at the system level.

Using the Parsl-RNASeq workflow, we are able to execute data-intensive transcriptomics software in HPC infrastructure to enable performance optimization of the workflow execution in a useful way. We have validated our approach by executing a massive amount of cardiomyocyte sequencing data of the evolutionary conserved Wnt pathway, using normal and anomalous conditions. Zelarayan et al. [7] used an *in vivo* mouse model in which β-catenin is acutely stabilized in adult cardiomyocytes, leading to increased ventricular TCF7L2 expression and activation of target genes. The aim is to understand the consequences of increased Wnt signaling pathway activity, comparing transcriptome profiles of normal (Cre recombinase “positive” control with a WT β-catenin locus) and β-catenin stabilized murine adult cardiac ventricles.

Our main goal is to study the viability of efficiently executing transcriptomics workflows on large HPC systems. The main contributions of this paper include 1. The collection and analysis of performance data from transcriptomics workflows; 2. The case study use of a real-world RNA-Seq workflow; and 3. The analysis of HPC performance metrics as CPU and memory usage, I/O operations, job dependencies, among others.

This paper is organized as follows. Section 2 brings the background on RNA-Seq differential gene expression. Section 3 describes the specification and implementation of the ParslRNA-Seq workflow. Section 4 presents materials and methods. Section 5 shows experimental results. Finally, Section 6 summarizes our findings and discusses future work.

2 Background on RNA-Seq

RNA sequencing (RNA-Seq) uses the capabilities of high-throughput sequencing methods to provide insight into the transcriptome of a cell in a given physiological or developmental condition as diseases derived by genetic variation e.g., cancer. Beyond quantifying gene expression, the data generated by RNA-Seq facilitate the discovery of novel transcripts, identification of alternatively spliced genes, and detection of allele-specific expression. RNA-Seq investigates different populations of RNA including the study of alternative splicing, Single Nucleotide Polymorphisms (SNPs), post-transcriptional modifications, and changes in gene expression over time or between treatment groups or disease progression.

Differential Gene Expression analysis (DGE) allows for elucidating the expression level between different experimental conditions and establishing whether there is a significant difference between them. DGE of RNA-seq data generally consists of three components: normalization of counts, parameter estimation of the statistical model, and tests for differential expression [4].

doi: 10.13053/CyS-26-4-4437

The data count of samples is tabulated containing the number of sequence fragments assigned to each gene. The quantification and statistical inference are established between systematic changes and variability of different conditions. Testing differential expression, provided in the DESeq2 package, uses negative binomial generalized linear models to estimate dispersion and logarithmic changes and to incorporate prior distributions based on data.

3 ParslRNA-Seq Scientific Workflow

This section presents the conceptual specification of the ParslRNA-Seq workflow for differential gene expression analysis (DGEs), available from https://github.com/lucruzz/rna-seq. ParslRNA-Seq receives four data information: the reference genome of *Mus musculus*; the gene transfer format (GTF) file used to hold information about gene structure; the FastQ files designed to handle sequence and quality scores represented as single ASCII characters; and a CSV text file containing the list of FastQs and metadata from experimental conditions.

ParslRNA-Seq is composed of six activities (illustrated in Figure 1). Activity 1 executes the Bowtie2 package that maps and compares genome readings, character by character. Activity 2 executes the Samtools program that orders readings and generates a compressed binary format. Activity 3 executes the Picard program to handle and split the readings files into $n = 24$ subfiles, such as 24 is the number of available threads.

Activity 4 runs the htseq-count program (HTSeq package), which counts the overlap of reads with genes in DGE. HTSeq sends the mapped reading files to each of the n available cores in a multicore execution. It generates a single output file with $n + 1$ columns containing the genes in the first column and the counts of each file in the remaining columns. Activity 5 executes the HTSeq-Merge Python script that joins the data information from the previous HTSeq execution and generates a file with a column containing the counts' results. Activity 6 executes the DESeq2 package to apply DGE statistics on the counts of the control and Wnt Wingless pathway conditions.

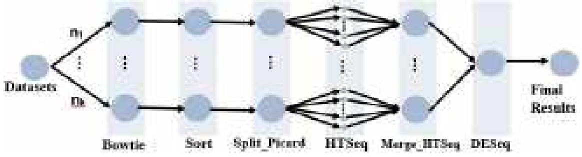


Fig. 1. ParslRNA-Seq conceptual view

Parsl [2], a parallel scripting library, provides an easy-to-use model composed of Parsl-Python functions and supports the management and execution of transcriptomics software, assuring reproducibility. Parsl manages the parallel execution of ParslRNA-Seq by applying parameter sweep mechanisms in HPC clusters. Each processing unit operates on the data independently via separate instruction streams. ParslRNA-Seq provenance data is automatically captured by Parsl. Parsl has already been successfully experienced in other complex computing-intensive bioinformatics experiments in HPC environments [5, 6]. Experimental results reinforce the importance of ParslRNA-Seq to help scientists in detecting DEGs from raw sequencing data, with Parsl supporting the management of tasks and provenance data in HPC environments.

4 Materials and Methods

4.1 RNA-Seq Data

Activation of the evolutionarily conserved Wnt pathway has been reported during maladaptive cardiac remodeling. However, the function of Wnt-transcriptional activation in the adult heart is mainly unknown. Zelarayan et al. [7] performed the transcriptome and genome analysis at the University Medical Center, Goettingen. RNA was isolated from mice cardiac tissue and RNA libraries were prepared for sequencing using standard Illumina protocols. Sequence reads were aligned to the mouse reference assembly (UCSC version mm9) using Bowtie 2.0. For each gene, the number of mapped reads was counted using htseq-count and DESeq2 was used to analyze the differential expression. *Mus musculus* GEO.ID is GSE97763.

This study belongs to a real RNA-Seq[1] experiment. It uses six FastQ sequencing data files of cardiac ventricles deposited in Gene Expression Omnibus[2] (GEO) public repository under accession GSE97763 and GSE97762 for RNA-seq datasets. FastQs accession numbers of the control condition group are: SRR5445794, SRR5445795, SRR5445796 and for the Wnt pathway: SRR5445797, SRR5445798, SRR5445799, including datasets of varying sizes (1.8 to 3 GB).

4.2 Experiment Setup

We follow the same protocol adopted by Zelarayan et al. 2018 [7] to validate and analyze transcriptomics results obtained by the executions performed with the ParslRNA-Seq scientific workflow. The transcriptomics software used in experiments are Bowtie2[3] program, Samtools[4] program version 1.10, Picard[5] program version 2.25.0, HTSeq[6] framework version 0.13.5 with the htseq-count script, HTSeq-Merge Python homemade-script, and DESeq2[7] package. All software, libraries and dependencies, Parsl and Python components, Intel VTune Profiler[8], and Darshan[9] tool were deployed at the top of the Santos Dumont environment.

4.3 Santos Dumont Supercomputer

The Santos Dumont (SDumont) supercomputer, one of the largest in Latin America, is located at the National Laboratory for Scientific Computing (LNCC/MCTI, Brazil). SDumont has an installed processing capacity of 5.1 Petaflops with a total of 36,472 CPU cores distributed across 1,134 compute nodes. SDumont has a Lustre parallel file system, integrated into the Infiniband network, with a raw storage capacity of around 1.7 PBytes and a secondary file system with a raw capacity of 640 TBytes. For our experiments, six nodes of SDumont were utilized, each node uses two Ivy Bridge Intel Xeon E5-2695v2 CPUs (12c @2.4GHz) and 64 GB of RAM. For more information, visit http://sdumont.lncc.br.

4.4 Parsing RNA-seq Files Strategy

The NGS's big challenge lies in the enormous data size for every single sample analyzed. A strategy to optimize parallelization in an HPC environment is to split input data into small and equally-sized portions. For parallelized read mapping, the alignment is carried out in parallel either by making use of array jobs or by distributing the data across multiple threads using OpenMP or across multiple compute nodes using the message passing interface (MPI).

We propose strategies managed by Parsl that scale to hundreds of threads better than single processed workflows or pipelined approaches. We explore how the FastQ file format, its unpredictable record boundaries, in particular, can impede thread scaling. We suggest a way to modify FastQ files while dividing the file size into blocks and how including these activities in workflow enables further improvements in thread scaling.

In RNA-Seq, to improve thread scaling we should restructure inputs and outputs, converting standard FastQ (or SAM) files to blocked FastQ files [8], where the number of input reads per block (N) are 24, to make best use of the 24 threads per node used in SDumont. The number of FastQ reads per thread must be chosen for each ParslRNA-Seq configuration and system.

ParslRNA-Seq starts with Bowtie (first activity); then the second activity Sort sorts the SAM or BAM files. The third activity Split_Picard retains the sort order of reads matching to the original BAM, splits files, separates files into Nth reads, and finally creates an output directory for storing split BAM files. In the fourth activity, HTSeq processes split files by calling the *"–nprocesses = 24"* argument.

[1] https://sfb1002.med.uni-goettingen.de/production/literature/publications/201

[2] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE97763

[3] http://bowtie-bio.sourceforge.net/bowtie2/index.shtml

[4] http://www.htslib.org/doc/samtools.html

[5] http://broadinstitute.github.io/picard/

[6] https://htseq.readthedocs.io/

[7] https://bioconductor.org/packages/DESeq2/

[8] http://intel.ly/vtune-amplifier-xe

[9] https://www.mcs.anl.gov/research/projects/darshan/

4.5 Multithreading & Multiprocessing Strategy

We consider in this work that a Multiprocessor (MP) is a system with more than one processor that assigns separate memory and resources for each process. Conversely, Multithreading (MT) is a programming model in which multiple threads run collaboratively in a single processor. Creating multiple threads inside a single process often help increasing performance. Moreover, we notice that modifications in some ParslRNA-Seq activities (mainly Bowtie2 and HTseq) are potential points to explore MT or MP thread scaling, as they can increase the computing speed of the system.

The ParslRNA-Seq workflow code (https://github.com/lucruzz/RNA-seq/blob/master/RNA-seq.py) shows the software command lines. While Bowtie2 creates MT processes, HTSeq "–nprocesses" (MP argument) only works to process different BAM files in parallel, i.e., htseq-count on one file is not parallelized. For instance, let us consider the following context in our ParslRNA-Seq processes. While we focus on making the best use of threads in a single process, an alternative is to run multiple simultaneous processes, possibly with many threads each. ParslRNA-Seq consumes six input FastQs, each deployed in parallel in an independent node.

For each node, Bowtie2 sets the performance option "-p/–threads NTHREADS" to launch the number of parallel search threads (default: 1) to process each FastQ. The threads will run on separate processors/cores and synchronize when parsing reads the output alignments, increasing alignment throughput by approximately a multiple of the number of the threads (linearly).

Split_Picard "SplitSamByNumberOfReads" option splits an input query-grouped SAM or BAM file into multiple (e.g., 24) BAM files while maintaining the sort order to parallelize alignment. The HTSeq *"–nprocesses"* processes those BAM files.

MP can suffer from load imbalance as some batches take longer to execute than others, and the job's duration is determined by the longest-running batch. Merge_HTSeq suffers this impact whereby some lock-holding threads are slow to finish their works (and release the lock) due waiting threads are using its resources. Finally, DESeq2 should wait for Merge_HTSeq finishes to be executed.

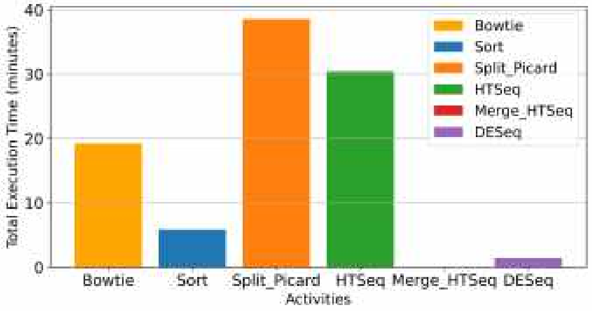


Fig. 2. Execution time in minutes of the ParslRNA-Seq' activities on a single node

4.6 Profiling Strategy

A global overview of the CPU and I/O systems is the first step to understanding the computational demands on the machine and detecting opportunities to optimize application performance, system performance, and system configuration for HPC.

We obtained and studied CPU and I/O performance reports obtained on the Santos Dumont supercomputer (SDumont). Those were obtained with the Intel VTune and the Darshan[10] I/O profiling tools, for CPU and I/O, respectively. This choice was motivated by a previous study of SDumont's performance conducted by Bez et al., 2020 [3], which provides details about how to characterize the application I/O phases from coarse-grained aggregated traces using Darshan.

Intel VTune provides insight into CPU and threads performance, scalability, bandwidth, caching, and more. VTune collects the tallies from all the cores' counters at frequent intervals; when the run is over, it analyzes the collection and presents the results via a GUI. The Darshan tool version 3.1.4 profiles executions on I/O metrics. Bowtie2 and HTseq are the most representative CPU and time-consuming software of ParslRNA-Seq executed in SDumont in 2021, and they were the main focus in our case studies.

[10] https://www.mcs.anl.gov/research/projects/darshan/

5 Results and Discussions

In this section, we present the performance and scalability of the parallel executions of transcriptomics scientific workflows. We have evaluated the CPU and I/O behavior of the ParslRNA-Seq executions in the Santos Dumont supercomputer. The CPU analyses are extensions of [5, 3].

5.1 CPU Performance Results Using VTune

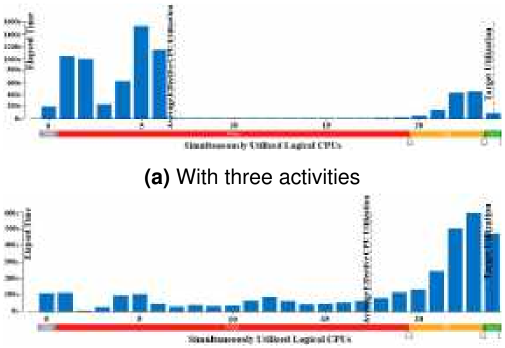
ParslRNA-Seq CPU Performance.

ParslRNA-Seq allocates, executes, and manages each of the six FastQs in one node of 24 threads. CPU and I/O influence the HPC scalability of the workflow. The original workflow modeling -of three activities- ([5, 6]) has been modified to optimize parallel and distributed executions and improve the total execution time (TET) (Figure 2, Figure 3, and Figure 4). The actual ParslRNA-Seq workflow — of six activities — was described in Figure 1.

Figure 2 presents the TET achieved by each of the six activities of ParslRNA-Seq. Bowtie2, Split_Picard, and HTSeq are the activities with the longest execution times (Figure 2), while Bowtie2 and Sort are the most I/O-intensive activities, i.e., they spend more time computing I/O relative to their total execution time (Figure 5). Then, we focused on parsing files, multithreading, and multiprocessing to improve the workflow behavior, mainly for the Bowtie2 and HTSeq activities.

ParslRNA-Seq Multithreading Performance. Figure 3 presents the workflow efficiency for the multithreading performance. The figure plots the node count on the horizontal axis and maximum per-thread wall-clock time, i.e., the time required to align all reads, on the vertical axis. The figure shows both versions of ParslRNA-Seq with (a) the previous version of three activities and (b) the ParslRNA-Seq modified model of six activities. We can see that the new model, while adding activities, improved performance and efficiency.

In Figure 3(a) Bowtie2 multithreading executes a task for each (not parsed) FastQ file in a node (24 threads). In addition, HTSeq executes each (not parsed) file in an entire node, i.e., no MT or MP strategy was applied. In Figure 3(b), Bowtie2 multithreading also executes a task for each (not parsed) FastQ in a node. Nevertheless, HTSeq executes a task of each SAM block in a thread (a SAM file was parsed in 24 blocks), i.e., each block is assigned to a thread.



(a) With three activities

(b) With six activities

Fig. 3. ParslRNA-Seq multithreading performance (seconds) on a single node. The y axis is different for each plot

A better distribution in the use of CPU cores in parallel was observed in Figure 3(b) due to the use of MP and MT approaches. However, that also required the insertion of extra activities in the workflow modeling. So despite the parallel execution of tasks performed by Parsl, there is still a considerable number of idle CPUs in most of the workflow execution. Figure 3(b) shows the use of up to 20 CPUs has an effective average CPU utilization considered as “poor” but over 20 CPUs the simultaneous use of processors presented an “ideal” utilization.

ParslRNA-Seq Multiprocessing Performance. Figure 4 shows two multiprocessing scenarios executed on SDumont: previous ParslRNA-Seq version of three activities in red and the optimized ParslRNA-Seq version of six activities in blue. With a single node, the new ParslRNA-Seq version (in blue) improves performance over the previous version from 72 to 65 minutes.

We further increased the number of nodes from 1 to 6 (with 24 cores per node).

This experiment consumed six FastQs by execution. The previous workflow version does not scale with the number of nodes (the execution

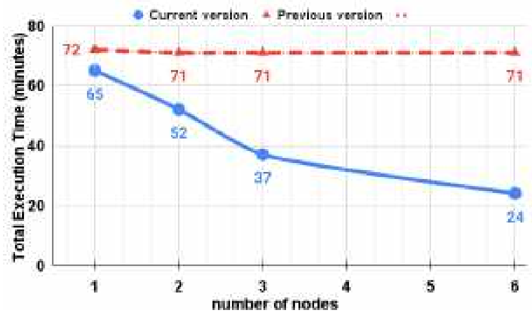


Fig. 4. Execution time of ParslRNA-Seq (in minutes) varying the number of nodes

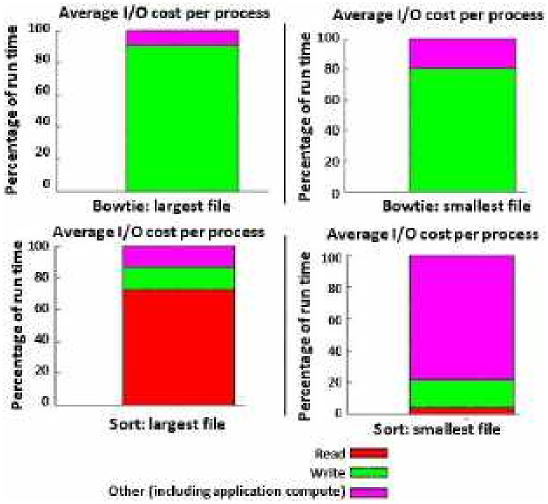


Fig. 5. Execution time of Bowtie2 and Sort I/O separated by activity, as reported by Darshan

time remains the same). On the other hand, the optimized workflow's performance is improved with more available nodes and cores. Time was decreased from 65 minutes (using 24 cores) to 24 minutes (using 144 cores), which means a speed-up (an acceleration factor) of 2.7.

5.2 I/O Performance Results Using Darshan

The workflow was executed with the Darshan profiler to investigate its I/O behavior. For discussion, we analyzed two input sizes, 1.8 and 3 GB. Figure 5 presents the distribution of execution time, as reported by Darshan, in POSIX read (in red, the first part of the bars on the bottom) and write (in green, the middle part of the bars) operations and other operations including processing (in pink, the top portion of the bars). The plots on the top show results for Bowtie2, and on the bottom Sort due both activities present the highest average I/O cost. On the left are the results with the largest tested input size, and on the right the smallest. HTSeq, Split, DESeq2, and Merge_HTSeq were omitted here because they spent less than 10% of time in I/O.

For both applications, increasing the input size increases the proportion of the execution time spent on I/O. That indicates that the I/O limits the scalability of these codes: as more data is treated, most time is spent on I/O and thus the CPU-focused optimizations presented earlier may have less impact on performance.

I/O Analysis for Bowtie2. Changing the input from 1.8 to 3 GB increases the run time from 152 seconds (80% on write operations) to 263 seconds (90% on writes). This increase was only due to I/O, with the write time increasing practically linearly with the input size. The output size was of 6 and 11 GB.

I/O Analysis for Sort. Time increased from 41 seconds (5% on read and 15% on write operations) to 91 seconds (70% on reads and 10% on writes). While the writing time remained relatively constant (output size was 657 MB and 1.1 GB), the reading time of Sort increased over 30 times by doubling the input size, which indicates the reading portion of this code is an important limitation factor for its performance.

5.3 Biological Analysis

Our biological performance results were validated to the reported by Zelarayan et al. [7] at the Gottingen University, in a collaboration between our research groups, reporting almost identical biological results. In the present article, we proposed the ParslRNA-Seq workflow and the computational executions performed in SDumont.

6 Conclusions and Future Work

In this work, we have presented a real-world workflow analysis for data-intensive transcriptomics applications to enable performance optimization of HPC systems. To this end, we have evaluated various transcriptomics applications from ParslRNA-Seq to analyze massive amounts of RNA-seq data in a controlled environment. We have used Parsl as a workflow management system, VTune and Darshan as profiling tools, and the SDumont as our machine. Moreover, we have developed a new ParslRNA-Seq version tailored to the needs of tracking a workflow execution and identifying potential issues to improve performance.

Our experiments demonstrate that this optimized workflow can accurately orchestrate computation resources, helping to pinpoint relevant metrics to help identify performance problems. Our results show performance improvements of up to 63.08% of ParslRNA-Seq executions, from 65 minutes (24 cores) to 24 minutes (144 cores). Additionally, we characterized the I/O behavior of the workflow components, identifying I/O problems in two of them, which will be the focus of future optimization efforts.

Indeed, we plan on continuously improving the ParslRNA-Seq modeling and performance. We want to explore the possibility to stage intermediate data on computing nodes to minimize parallel file system activity. Furthermore, we want to include system-level monitoring data in our analysis, which may explain the observed behaviors, particularly regarding I/O. Finally, we plan to provide a mechanism to track data and metadata to enable offline analysis. We aim to introduce a new database automatically populated by Parsl or another SWfMSs so that users can retrieve workflow performance data. The data collected is a valuable training resource for automated machine learning analysis.

Acknowledgments

The work was partially funded by the Brazilian agencies FAPERJ and CNPq. Authors acknowledge LNCC/MCTI and SDumont supercomputer for the contribution to results reported within this paper. We thank Harald Kusch and Laura Zelarayan-Behrend from the Gottingen University for scientific discussions about RNA-Seq.

References

1. **Ahmed, A. E., Allen, J. M., Bhat, T., Burra, P., Fliege, C. E., Hart, S. N., Heldenbrand, J. R., Hudson, M. E., Istanto, D. D., Kalmbach, M. T., et al. (2021).** Design considerations for workflow management systems use in production genomics research and the clinic. Scientific Reports, Vol. 11, pp. 1–18. DOI: 10.1038/s41598-021-99288-8.

2. **Babuji, Y., Woodard, A., Li, Z., Katz, D. S., Clifford, B., Kumar, R., Lacinski, L., Chard, R., Wozniak, J. M., Foster, I., et al. (2019).** Parsl: Pervasive parallel programming in python. Proceedings of the 28th International Symposium on High-Performance Parallel and Distributed Computing, pp. 25–36. DOI: 10.1145/3307681.3325400.

3. **Bez, J. L., Carneiro, A. R., Pavan, P. J., Girelli, V. S., Boito, F. Z., Fagundes, B. A., Osthoff, C., da Silva-Dias, P. L., Méhaut, J. F., Navaux, P. O. (2020).** I/O performance of the santos dumont supercomputer. The International Journal of High Performance Computing Applications, Vol. 34, No. 2, pp. 227–245. DOI: 10.1177/1094342019868526.

4. **Costa-Silva, J., Domingues, D., Lopes, F. M. (2017).** RNA-seq differential expression analysis: An extended review and a software tool. PLOS ONE, Vol. 12, No. 12, pp. e0190152. DOI: 10.1371/journal.pone.0190152.

5. **Cruz, L., Coelho, M., Gadelha, L., Ocaña, K., Osthoff, C. (2020).** Avaliação de desempenho de um workflow científico para experimentos de rna-seq no supercomputador santos dumont. Anais Estendidos do XXI Simpósio em Sistemas Computacionais de Alto Desempenho, SBC, pp. 86–93. DOI: 10.5753/wscad_estendido.2020.14093.

6. **Cruz, L., Coelho, M., Terra, R., Carvalho, D., Gadelha, L., Osthoff, C., Ocaña, K. (2021).** Workflows científicos de rna-seq em ambientes distribuídos de alto desempenho: Otimização de desempenho e análises de dados de expressão diferencial de genes. Anais do XV Brazilian e-Science Workshop, SBC, pp. 57–64. DOI: 10.5753/bresci.2021.15789.

7. **Iyer, L. M., Nagarajan, S., Woelfer, M., Schoger, E., Khadjeh, S., Zafiriou, M. P., Kari, V., Herting, J., Pang, S. T., Weber, T., et al. (2018).** A context-specific cardiac β-catenin and gata4 interaction influences tcf7l2 occupancy and remodels chromatin driving disease progression in the adult heart. Nucleic Acids Research, Vol. 46, No. 6, pp. 2850–2867. DOI: 10.1093/nar/gky049.

8. **Langmead, B., Wilks, C., Antonescu, V., Charles, R. (2018).** Scaling read aligners to hundreds of threads on general-purpose processors. Bioinformatics, Vol. 35, No. 3, pp. 421–432. DOI: 10.1093/bioinformatics/bty648.

9. **Papadimitriou, G., Wang, C., Vahi, K., da Silva, R. F., Mandal, A., Liu, Z., Mayani, R., Rynge, M., Kiran, M., Lynch, V. E., et al. (2021).** End-to-end online performance data capture and analysis for scientific workflows. Future Generation Computer Systems, Vol. 117, pp. 387–400. DOI: 10.1016/j.future.2020.11.024.

Article received on 11/05/2022; accepted on 18/09/2022.
Corresponding author is Kary Ocaña.

ISSN 2007-9737

Towards an Active Foveated Approach to Computer Vision

Dario Dematties[1], Silvio Rizzi[2], George K. Thiruvathukal[3],
Alejandro Wainselboim[4]

[1] Northwestern University,
Northwestern Argonne Institute of Science and Engineering,
United States

[2] Argonne Leadership Computer Facility,
Argonne National Laboratory,
United States

[3] Loyola University Chicago,
Computer Science Department,
United States

[4] Instituto de Ciencias Humanas, Sociales y Ambientales,
CONICET Mendoza Technological Scientific,
Argentina

dario.dematties@northwestern.edu, srizzi@alcf.anl.gov

Abstract. In this paper, a series of experimental methods are presented explaining a new approach towards active foveated Computer Vision (CV). This is a collaborative effort between researchers at CONICET Mendoza Technological Scientific Center from Argentina, Argonne National Laboratory (ANL), and Loyola University Chicago from the US. The aim is to advance new CV approaches more in line with those found in biological agents in order to bring novel solutions to the main problems faced by current CV applications. Basically this work enhance Self-supervised (SS) learning, incorporating foveated vision plus saccadic behavior in order to improve training and computational efficiency without reducing performance significantly. This paper includes a compendium of methods' explanations, and since this is a work that is currently in progress, only preliminary results are provided. We also make our code fully available.[1]

Keywords. Foveated computer vision, saccadic behavior, reinforcement learning, self-supervised learning, General-Purpose Graphics Processing Units (GPGPUs).

1 Introduction

1.1 About the Collaboration

We begin by highlighting some aspects of international collaboration. This work is an extended version of a talk presented in the Americas HPC Collaboration Workshop, part of the CARLA 2021 Latin America High Performance Computing Conference.

Our scope aligns particularly well with the aims of the workshop, especially "partnerships formed between researchers and entities across the Americas, from Patagonia to Alaska" [2].

[1] https://github.com/dariodematties/Multimodal-Active-AI

[2] See http://carla2021.org/callforworkshops

doi: 10.13053/CyS-26-4-4436

We are a geographically-distributed team of investigators hailing from research and educational institutions in Argentina and the United States. We collaborate by leveraging leadership supercomputing resources in our research, and state-of-the-art tools for remote collaboration, which are discussed below.

Throughout years of successful collaboration, we have advised and graduated a doctoral student (Co-author, Dematties) and published in prestigious journals [7, 6, 8]. As part of his graduate education, Dr. Dematties attended the Argonne Training Program for Extreme-Scale Computing (ATPESC), where he acquired invaluable experience with common tools used in High Performance Computing. Readers interested in knowing more about the program are invited to visit https://extremecomputingtraining.anl.gov/

This experience allowed Dr. Dematties to port his software infrastructure to supercomputers, leveraging hybrid OpenMP+MPI parallelism. A Director's Discretionary allocation was granted at the Argonne Leadership Computing Facility, providing the foundation to perform large-scale computational experiments.

This collaboration makes extensive use of tools such as Zoom videoconferencing, GitHub for collaborative development, and Zenodo for publishing datasets and results.

Our work continues well past Dr. Dematties earning his Ph.D. We would especially like to mention our participation in the CyberColombia 2020 conference, where we presented a tutorial at the HPC Summer School, which covered the science behind bio-inspired models, working with supercomputers, and software engineering. See https://figshare.com/articles/presentation/Towards_High-End_Scalability_on_Bio-Inspired_Computational_Models/12762260 for the tutorial materials.

1.2 About the Research

The difficulty linked to CV comes from its hardware limitations as well as its data set shortages for training. In some cases, CV applications could depend on near real-time video processing, demanding Artificial Intelligence (AI) solutions on edge computing devices [3] which appear as the only way to overcome the latency limitations of centralized computing.

Fitting CV models on edge devices is not an easy task, given the complexity of such models. In other applications–such as in 3D medical imaging–the computational demands could be prohibitively expensive and the data sets collection could require extremely skillful staff resulting in prevalent scarcity.

Facing such challenges requires new ideas in this area. For instance, the exorbitant demands on labeled data sets could be alleviated using new SS strategies while the excessive computational demands imposed by these algorithms could be reduced utilizing inspiration from visual systems found in biological agents.

We humans as well as other higher mammals do not sense visual information as we perceive images. The retina, a specific organ located in the posterior hemisphere of the eye ball has the function of transforming rays of light entering the eye into electric signals which are later processed by the brain [25].

Yet, the perception of an image is not only a matter of the information coming from outside. We also affect our visual field perception with the architecture of our visual system and our behavior.

From an architectural point of view, our retina samples the visual field with a very high resolution in a tiny portion called fovea and with very low resolution in the periphery of such a structure (Fig. 1). From a behavioral point of view, our saccadic behavior determines where, when and how long we fixates. This significantly affects the way in which we sense and perceive the world around us.

Evidently, foveated vision reduces computational (metabolic) cost, since it is not necessary for the brain to process all the scene at high resolution. Yet, this strategy brings an undeniable cost in a lost of information. What remains is only an appropriate saccadic behavior in order to make information

[3] Edge computing is a distributed computing paradigm that brings computation and data storage closer to the sources of data. This is expected to improve response times and save bandwidth. Source: https://en.wikipedia.org/wiki/Edge_computing

processing more efficient for reproduction and survival in certain niche.

In this manuscript we report our current endeavor towards solving some of the major challenges faced by CV by means of active foveated strategies. Basically, the complete system depicted in Fig. 2 aims to palliate data sets scarcity and the prohibitive computational demands found in CV models.

We first developed a foveated system which pre-processes a batch of images. This foveated system is based on the physiology of the visual system and not on psychological aspects of vision as is the case in other works [1, 11]. We addressed foveated vision utilizing its properties as a natural augmentation approach for self-supervised learning.

Fig. 2 A shows how we advanced a strategy utilized in SimCLR [4], where a new approach to contrastive self-supervised learning algorithm is proposed.

A network is taught to discriminate images disregarding several augmentations–*i.e* crop-resize, Gaussian blur, Gaussian Noise, Color distortions, flipping, rotations, cutouts, etc. In our work we propose that such augmentations could be obtained in a more biologically inspired strategy by means of our foveated system.

A similar rationale is conducted in [11]. Basically we implemented a SimCLR like algorithm in which we teach a Residual Neural Network (ResNet) architecture to distinguish foveated fixations that come from the same image from those coming from different images.

We also incorporated additional augmentations such as color distortion, crop and resize, Gaussian noise and flipping. We tested the learned representations by means of a linear classifier–as is the standard protocol used in SimCLR.

Fig. 2 B shows how we also incorporated a transformer architecture utilizing our pre-trained ResNet network as a backbone. To that end we adapted an architecture developed by Facebook AI Research Group called DEtection TRansformer (DETR) [3]. In DETR a new method is developed that conceives object detection as a direct–end-to-end–prediction problem.

Transformers are usually employed in Seq2Seq modelling approaches especially in language models. In DETR, such an architecture is used encoding an image pixel by pixel–instead of word by word as in Natural Language Processing (NLP). In our case we adapted the original architecture eliminating several losses concerning detection, keeping only losses concerned with classification.

We also adapted the positional encoding mechanisms of the network and instead of encoding the position of every component from the ResNet backbone output we encoded the position of each fixation from our pre-trained ResNet backbone.

Self-attention has a quadratic complexity and in a patch by patch scenario–as is the case in image Transformers–this situation represents a great obstacle in the implementation of this kind of architectures when trying to process high resolution images. We address this by changing the strategy of giving each patch a position.

We instead give each fixation a position in the network. The number of fixations will be considerable smaller than the number of patches in an image. This is a huge advantage in computational load terms, especially when the use transformers brings to the scene a complexity of n^2 where n is the length of the sequence.

Finally, as shown in Fig. 2 C, we incorporated a RL mechanism in our model with the aim of learning an effective saccadic behavior. Basically we trained a DQN, which treated the dynamic of the Transformer classifier as the environment.

The observation of the state of the environment was the output from our foveated system, the actions taken by our network were the coordinates of the next fixation which gave rise to the next state from our foveator. Finally the classification performance from DETR was taken as the reward in this RL scenario.

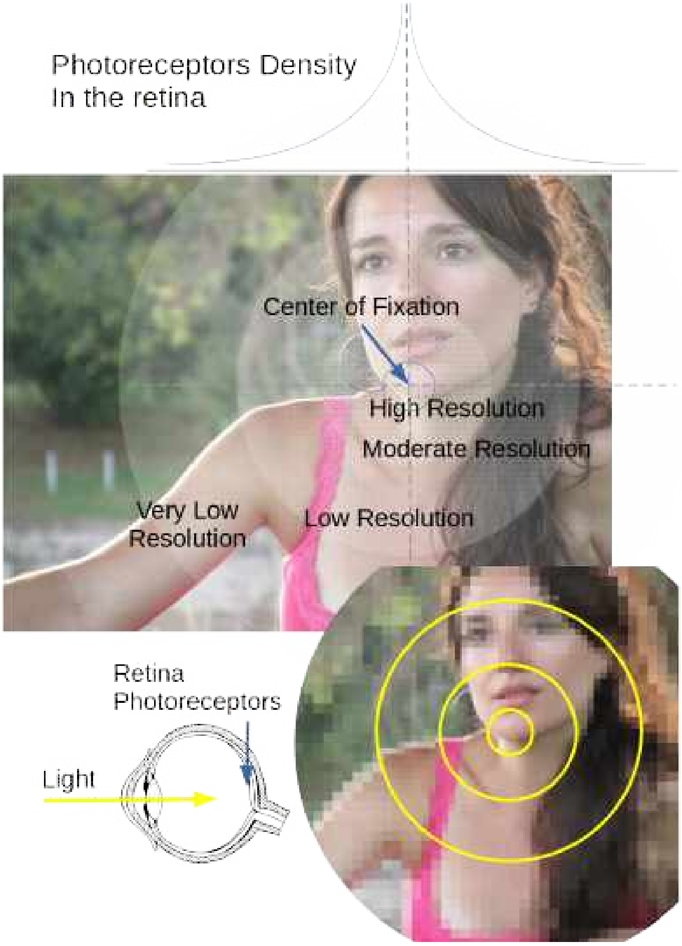


Fig. 1. Foveation in biological agents is a phenomenon in which the density of photoreceptors located on the eye's retina varies in a way that there is acutely more density near the fovea, while such a density decreases drastically in the fovea's vicinity. The fovea, is a small fraction of the retina which corresponds to the center of fixation in the sight. The consequence is that the perceptual detail regarded by the agent varies across the image according to the current fixation point, which confers the highest resolution region of the image to the center of the eye's retina, (*i.e.* the fovea)

2 Related Work

In a recent work [5], without using foveation, but in lines with saving computational effort and retaining fine details in CV tasks, a method based on a differentiable Top-K operator to select the most relevant parts of high resolution images was introduced.

In regards to foveation, in [13], a foveated vision system was introduced for face reconstruction algorithms.

In [9] a foveated model to provide *clutter* measures was introduced[4]. In regards to object detection, in [1] a foveated object detector was introduced.

Later, in [10] NeuroFovea was proposed as a model to generate visual metamerism in images [5]. In [18], it was investigated quantitatively how

[4] Clutter perception is the typically negative visual perception effect that emerges from the disordered organization of an excessive number of objects in a visual scene

[5] Metamers are stimuli that are physically distinct but that are perceived to be the same by a human observer

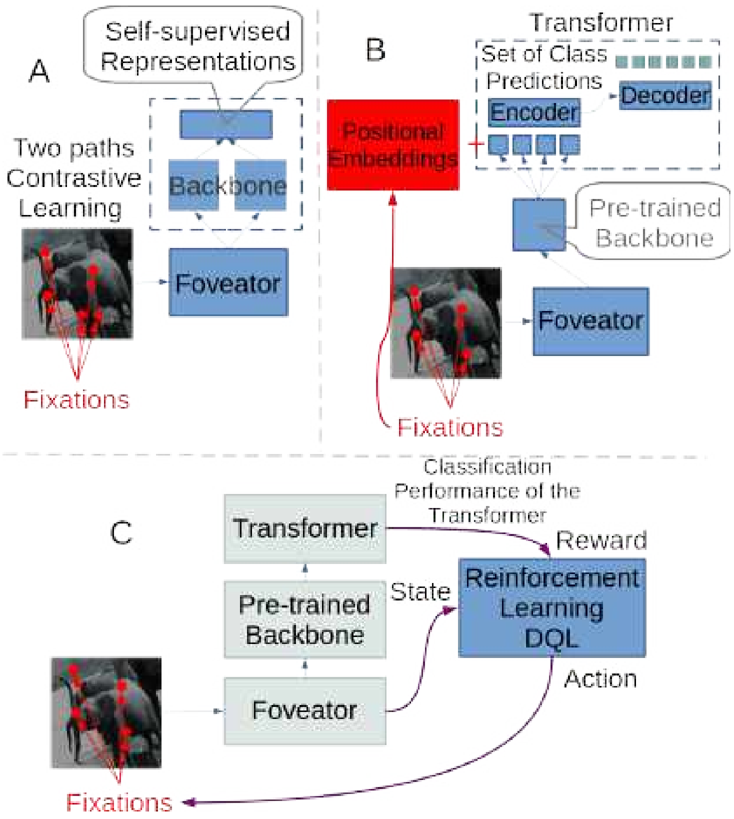


Fig. 2. Global strategic scheme to solve CV problems such as the labeled data sets scarcity and the high computational complexity demanded by the models implementation. (A) Self-supervised contrastive learning approach using foveated vision as an additional biologically-inspired augmentation strategy. With this strategy we aim to mitigate labeled data sets scarcity. (B) A transformer architecture processing a sequence of outputs from a pre-trained backbone which process foveated fixations. Positional embeddings are determined by individual fixations and not by image patches which saves great computing power demands from the transformer architecture perspective. (C) A Reinforcement Learning (RL) architecture–a Deep Q Network (DQN)–is added to the architecture to learn the saccadic behavior which is supposed to generate more effective fixation coordinates in order to increment the classification performance from the transformer

detection, recognition and processing speed in a Convolutional Neural Network (CNN) were affected by reducing image size using a foveated transformation.

In [20] images were compressed in videos applying foveation, gradually reducing the resolution in the periphery.

Afterwards images were reconstructed utilizing generative adversarial approaches. In [11] it was found that a CNN trained on foveated inputs with texture-like encoding on the peripheral information has similar scene classification performance to a matched resource CNN without foveated inputs.

Finally in [19] a foveated Transformer model was proposed.

None of the previous research analysed foveation utilizing computational hypotheses based on a developmental approach.

As is the case for this study, foveation has been applied following a developmental appeal from a SS learning strategy, passing through a Seq2Seq scheme–with random fixations–to finally end up employing a RL policy, learning the saccadic behavior of the agent.

3 Computational Hypotheses

3.1 Image Foveator

In Fig. 3 we have sketched the foveation process. First of all, we resize all the images in the batch to the same specific size (640 x 640). Then we apply a series of successive crop and resize operations. All crop operations are resized to a resolution of 30 x 30 pixels. In the first row in Fig. 3, there is not crop operation, we only resize the complete 640 x 640 span to a 30 x 30 pixels resolution. Successive crop operations are illustrated with corresponding yellow squares. Such operations reduce the complete 640 x 640 span to 400 x 400, 240 x 240, 100 x 100 and finally to 30 x 30 pixels.

We use NVIDIA DALI library for the process of foveation[6]. This library is utilized for data loading and pre-processing accelerating Deep Learning (DL) workflows.

Even though the foveation operation is–in itself–an augmentation operation, we also apply additional augmentations to the batch. The pipelines also apply operations of random resize plus crop before foveation. The random area in the cropping operation ranges from 10% to 100% of the span. The location of the cropping operation is also random.

The final size of the image after the random crop plus resize is 640 x 640 pixels. Flipping, Gaussian noise, color distortion and grid mask operations are also applied randomly. Both, flipping as well as color augmentations are applied with a 50% chance. When applied, color augmentation has several components such as brightness, contrast, hue and saturation which are also generated randomly. Grid mask is applied with a 10% chance. When applied, this has two components which are generated randomly too (*i.e.* ratio, tile). The ratio indicates the quotient between occluded and free space in the image, whilst the tile sets the size of the grid inside the image.

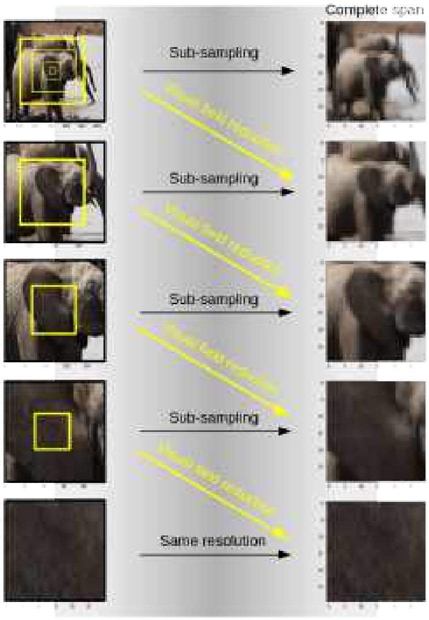


Fig. 3. Foveator. To simulate the foveation process found in some biological agents such as some mammals, we generate 5 spans from the same image around a fixation point. All the spans have the same number of pixels (30 x 30) but some of them sub-sample the complete original image while others sub-sample smaller regions of it. The smaller the region spanned the better the resolution captured by the (30 x 30) span. On the left hand side column we show the spans extracted from the original portions of the image. On the right hand side column we show the 30 x 30 spans returned by the foveator. From top to bottom rows we have bigger spans with less resolution to smaller spans with higher resolution. Yellow squares show the following span proportion respecting the previous span

3.2 Self-Supervised Approach

SS methods do not rely on human created labels but on intrinsic characteristics immersed in the statistical structure of data sets. Yet, SS learning does not only confers advantages from saving large and expensive labeled data sets. It delivers much richer representations which are not constrained to loss functions supported only by human provided labels, but by diverse features hidden in the statistical structure of the data sets. Human provided labels are instead subjective and limited. The pre-training phases in SS learning

[6] https://docs.nvidia.com/deeplearning/dali/user-guide/docs/

make the networks to acquire relevant information through loss functions based on pretext tasks.

In vision, pretext tasks are very diverse. In this work we will concentrate our attention in Contrastive Learning (CL), specifically, in the research conducted by T. Chen et al. [4]. In CL the pretext task consists on teaching a model to classify different augmented versions of an image as coming from the same image and to discriminate such augmentations when coming from different images. In this way the model learns image features with invariant properties to the different augmentations applied. The hypothesis is that the more augmentations one adds to the inputs, the more robust are the features acquired.

In CL, a batch of images is augmented applying diverse augmentations such as crop and resize, flipping, color distortions, rotations, cutouts, Gaussian noise, Gaussian blur and filtering among others. Generally a batch of images is augmented twice, providing two augmented versions of the batch. Afterwards a Neural Network (NN) is trained to maximize agreement between representations produced by augmentations coming from the same image and minimize such an agreement between augmentations coming from different images.

In [4] the NN is composed by two stages–*i.e.* $f(\cdot)$ and $g(\cdot)$. $f(\cdot)$ is called the base encoder network, while $g(\cdot)$ is called the projection head. After training, the projection head is removed using only the output from $f(\cdot)$ for any downstream task.

We approached this same strategy utilizing augmentations coming from our foveated system. We humans perceive visual information as static and well defined scenes even when we do several saccades per second. This means that at each second our retina is receiving information from very different versions of the same perceived scene. In some way, our visual system is considering such different representations provided by different fixations as coming from the same source of information. As a result we see a static scene.

We hypothesize that our propioceptive system, in tandem with our saccades, inform our visual system that we are watching at the same image. Maybe the best way to survive and reproduce is that–under such circumstances–our visual system learned [7] to produce a representation that we perceive as a static and well defined scene. Moving our gaze to another location is reported by our propioceptive system and the information aproaching our retina could be considered as coming from a different source.

Inspired by this biological rationale we applied the method developed in [4] but producing the augmented images by means of different fixations coming from our foveator. We only used 4 of the 5 spans showed in Fig. 3. We discarded the first complete span and used instead only the spans from 2 to 5. We also implemented $f(\cdot)$ utilizing a ResNet 50 and $g(\cdot)$ by means of a MultiLayer perceptron (MLP) as in the original implementation.

Additionally we incorporated further augmentations in advance to the foveation process, for instance, we added crop and resize, color distortion, Gaussian noise and grid mask. Such additional augmentation improved representations considerably. Some aspects related to the quality of the representations will be addressed in the following section.

3.3 Supervised Approaches

3.3.1 Linear Evaluation

With the aim of evaluating the learned representations in $f(\cdot)$, we followed the linear evaluation protocol utilized in [4]. To that end we trained a linear classifier on top of the frozen base network $f(\cdot)$, and then tested the accuracy of the linear classifier using it as a proxy for representation quality.

In our case we generated n fixations from our foveator and passed them through our frozen base network $f(\cdot)$. We then collected the n outputs from $f(\cdot)$ and merged then in a unique vector which we used as input for the linear classifier. The n fixations were produced randomly, *i.e.* no pattern was followed to cover the image in any conceivable way with the fixation locations.

[7] When we talk about *learn* here we mean phylogenetic and ontogenetic processes

3.3.2 Processing Sequences of Fixations with a Transformer Architecture

Attention mechanisms–predominantly self–attention–came to the scene playing a more important role in deep feature representation in CV. This strategy captures long-range dependencies within a single sample [29].

Nevertheless, self-attention has a quadratic complexity which could make its implementation difficult, especially in high resolution images with maybe millions of pixels. With the aim of circumventing this problem, alternative architectures are implemented for substituting self-attention [14, 23, 28].

In our approach we propose a different strategy. Instead of encoding positions for each pixel–or patch–in an image, we encode positions for each fixation in the visual field.

The number of fixations executed by our foveator will–logically–tend to be considerably smaller than the number of pixels–or maybe patches–found in an image. For instance, for humans only two fixations suffice to recognize faces [16].

We used the learned representations in our base network $f(\cdot)$ fine-tuning it by means of the architecture showed in Fig. 2 B. We fed a Transformer with a series of learned representations from a sequence of fixations. We used the main organization and strategy introduced by N. Carion et al. [3].

We used a random number of fixations which ranged from 2 to 9. We also used 10 prediction queries which ended up being 10 image class predictions which in a way *voted* for the different classes in imagenet.

3.4 Reinforcement Learning for the Acquisition of Saccadic Behavior

In the system shown in Fig. 2 B, not only the number but also the locations of the successive fixations in an image were chosen at random.

Yet the behavioral patterns found in saccades of biological systems are far from random [26, 22, 27, 2]. Which are the optimization mechanisms behind the oculomotor behavior emergence in biological systems? Compelling research shows that there are links between the dopaminergic reward system and the saccadic behavior of some mammals [21, 15, 17]. Hence, RL in saccadic behavior is amply supported by these data.

Thus, in Fig. 2 C we show the application of RL to our model. As can be seen in the figure, we use a Deep Q Learning strategy to optimize the saccadic behavior of the system to achieve better performance [24].

We use a ResNet-50 architecture which takes the model in Fig. 2 B as the part of the environment that produces rewards in response to changes in its states.

4 Implementation

Regarding the High Performance Computing (HPC) system We used ThetaGPU, which is an extension of Theta supercomputer at the Argonne Leadership Computing Facility (ALCF). ThetaGPU is composed of 24 NVIDIA DGX A100 nodes.

Each DGX A100 node comprises eight NVIDIA A100 Tensor Core Graphical Processing Units (GPUs) that provide 320 gigabytes of GPU memory for running Machine Learning (ML) workflows.

Regarding ML framework and model parallelization All the implementations have been done utilizing the Pytorch ML framework. DistributedDataParallel from Pytorch and mpi4py were used to manage dataset parallel processing. Basically we distributed 8 Message Passing Interface (MPI) processes in 8 GPUs.

MPI manages the communication among processes. DistributedDataParallel strategy–on the other hand–is to replicate the whole model in each MPI process.

Each model replica processes a different part of the dataset. Gradients computed during the backward pass in each model replica are communicated, averaged and used to conduct weight adjustments in each model.

In regards to dataset and pre-processing The dataset used in this work is imagenet ILSVRC 2012. The foveator is implemented using NVIDIA DALI library. With NVIDIA DALI we load, decode, foveate and augment the images from the dataset. DALI takes care of splitting the dataset in different shards in each epoch. Therefore, each network replica in each MPI rank takes care of a part of the dataset which corresponds to such a process in a given epoch. From one epoch to another, the assignment of shards to specific processes rotates in order to provide variation to the training process.

Regarding software compatibility To run our ML workflow, we used Singularity containerization. Singularity is a container system specifically designed for HPC that allowed us to define our own environment making our work portable and reproducible on any HPC that supports it. Therefore we proceeded to install all the necessary software in a singularity container and afterward we could use such a container to run our models in ThetaGPU.

5 Preliminary Results

The preliminary results of our experiments are shown in Table 1. Here we show results of contrastive accuracy while training the base network $f(\cdot)$, the linear evaluation of the frozen base network, the classification performance of the transformer architecture and finally the performance of the same transformer when successive fixations are guided by a DQN.

To train the base network we used a mini batch size of 128 images in 8 GPUs (*i.e.* batch size of 1024 images). This pre-training phase took 300 epochs, with *adam* optimizer, with a linear decaying *learning rate* schedule and with 5 *warm-up* epochs. The base network utilized was a ResNet 50. With a global batch size of 1024 images we end up with 2048 augmented fixations, each fixation has one positive example and 2046 negative examples in the CL approach.

For the linear evaluation we used the base network (*i.e.* our pre-trained ResNet 50) with its weights frozen and added a linear classifier at the top. We used 5 fixations for each image, without any augmentation. We applied a mini batch size of 512 images in 8 GPUs (*i.e.* batch size of 4096 images). The total number of epochs was 500 with 5 warm-up epochs. We used the same learning rate schedule used for the CL task.

For the Transformer training process we used the base network as a pre-trained backbone fine-tuning its weights with reduced learning-rate. We used a random number of fixations for each image which ranged from 2 to 9. We applied a mini batch of 64 images in 8 GPUs (*i.e.* batch size of 512 images). The total number of epochs was 68 without learning-rate scaling schedule.

6 Conclusion and Future Work

In this paper we compiled a series of methods aimed to find solutions to CV challenges by bio-inspired strategies. Our focus is in the application of foveation and saccadic behavior–in a developmental fashion–to achieve data set and computational savings in CV tasks without diminishing performance significantly. Although foveation provides notorious computational savings in the information processing flow, during experimentation we noticed that it also compromised performance in downstream CV tasks considerably (see Tab. 1).

One important aspect that could be causing this decline is the fact of considering only one positional location per fixation in section 3.3.2. Inside each fixation, there exist much more information corresponding to the complete foveation. From a tiny fraction of the visual field to almost its complete range, one foveation spans almost the entire image from low resolution wide spans to higher resolution acute spans at the center of fixation (Fig. 3).

Incorporating such information to the processing flow of the Transformer in the system could drastically improve the model's performance. Hence, a suitable strategy to follow is the one implemented by Jonnalagadda et al. [19]. In such a model, 11 Transformer blocks (0 to 10) process single fixations. Each of these blocks uses positional embeddings inside each foveation. That is, each foveation comprise all the positional information.

Table 1. Performance in contrastive learning when training the base network $f(\cdot)$, in the linear evaluation of the frozen base network and finally in the classification using a Transformer architecture

Top-1 Acc.	Contrastive acc.	Linear evaluation	Transformer class.	DQL class.
All augmentations	0.7696	0.2093	0.0937	0.0561
Without grid mask	0.8601	0.2502	0.1289	——

Then, the last Transformer block in this set (Transformer block number 10) provides its attention weights to chose the coordinates of the best next fixation location. A final Transformer block (Transformer 11) collects the successive outputs, corresponding to each fixation (as a sequence of fixations). The output from Transformer block 11 is used to classify the image. As we can see, in this model, positional information is managed inside each fixation (foveation).

In our model instead, we collapse all the information corresponding to one fixation in one position and use the positional information corresponding to the centers of the fixations. Our strategy provides an enormous computational saving regarding the quadratic complexity concerning Transformers but it could also be the source in the lost of information that is producing a sharp performance decline in our system. Future applications will take into account this issue, incorporating in some way positional information inside each fixation.

In regards to the acquisition of the saccadic behavior proposed in section 3.4, the RL mechanism ”sees” the classifier introduced in section 3.3.2 as its environment. The RL algorithm receives the successive outputs from the foveator as the states of the environment and the reward is the classification performance of the Seq2Seq classifier. As expected, the actions produced by the system control the next fixation coordinates.

The problem that instantaneously arise in the approach is that the classifier–which is considered as the environment by the RL system–is highly dynamic. The classifier's behavior changes continuously as a result of its training. This circumstance makes extremely difficult for the RL algorithm to learn the environment behavior and can in this way ”catch” a good policy at time of generating future fixations.

Several possible solution strategies arise in such regard, one is to alternatively freeze the classifier (the environment) and the RL mechanism as training proceeds. We could freeze the RL algorithm during one epoch and the classifier during the next one or maybe use several frozen epochs alternatively to promote stabilization in each algorithm.

In our case, we train the two networks together. In the first epoch we give a preliminary training to the Seq2Seq classifier using random fixation coordinates. In this first epoch, data is accumulated in a memory which collects information regarding states, actions, next states and rewards. This memory is used in the meantime to train the RL algorithm. At each batch the RL algorithm is trained at random, consuming the memory.

Next, in the subsequent epochs the training process of the classifier continues but the sequence of fixations is chosen by following actions accordingly to an epsilon greedy policy from the DQN. Briefly, sometimes we use our DQN to choose the action, and sometimes we just sample one randomly. The probability of choosing a random action starts high at the beginning and decays exponentially towards as training proceeds epoch by epoch.

This strategy is not returning good results either (Tab. 1). The dynamic character of the classifier makes us think in the application of more sophisticated RL strategies. Unexpected perturbations or unseen situations in RL scenarios cause proficient but specialized policies to fail at test time. Here our main problem is that the learning process in RL requires a huge number of trials every time the environment is modified. Animals instead learn new tasks in only a few trials, exploiting their prior knowledge about the world. We need a RL system that could adapt quickly to the changes produced in the classifier (our environment in this case). Several groups have tackled such a challenge [12].

In the next steps of our research we will inspire our model in the work produced in [19], incorporating positional information inside foveation in some way. There are many ways to do that, but we have to try to find the best way given the semantic behind images and foveation. In a second stage we will inspire our RL strategy trying to incorporate meta-RL to adapt rapidly the changes of the environment (our Seq2Seq classifier) [12].

7 Conclusions

CV as a sub-field of ML is a specific discipline in which humans prepare machines, making them able to autonomously do some of the visual task we do routinely. While humans and animals can naturally solve some of the more challenging CV tasks for machines, the possibility that machines provide in terms of scalability is peerless by biological agents in general. Therefore, it is paramount to provide machines with such a natural ability to solve routine CV problems at scale.

Yet, one of the far-reaching challenges in CV is our inability to understand the human visual system, which we think is cardinal for this endeavour. In this paper we report a series of steps devoted to solve some of the major challenges faced by CV–*i.e.* data set scarcity and algorithmic high computational demands–precisely, proposing a compendium of methodologies inspired in the visual system found in biological agents.

We fused SS learning with active foveated vision with the aim of palliating excessive data set and computational demands. We also noticed that the implementation of such methods seriously degrade performance in simple CV tasks. In this paper we propose alternative solutions to be implemented in future editions of this research.

Acknowledgments

This work used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy, Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

1. **Akbas, E., Eckstein, M. P. (2017).** Object detection through search with a foveated visual system. PLOS Computational Biology, Vol. 13, No. 10, pp. e1005743. DOI: 10.1371/journal.pcbi.1005743.

2. **Alahyane, N., Lemoine-Lardennois, C., Tailhefer, C., Collins, T., Fagard, J., Doré-Mazars, K. (2016).** Development and learning of saccadic eye movements in 7- to 42-month-old children. Journal of Vision, Vol. 16, No. 1, pp. 6–6. DOI: 10.1167/16.1.6.

3. **Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. (2020).** End-to-end object detection with transformers. European Conference on Computer Vision, Springer, Vol. 12346, pp. 213–229. DOI: 10.1007/978-3-030-58452-8_13.

4. **Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020).** A simple framework for contrastive learning of visual representations. International conference on machine learning, PMLR, pp. 1597–1607.

5. **Cordonnier, J. B., Mahendran, A., Dosovitskiy, A., Weissenborn, D., Uszkoreit, J., Unterthiner, T. (2021).** Differentiable patch selection for image recognition. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2351–2360.

6. **Dematties, D., Rizzi, S., Thiruvathukal, G. K., Pérez, M. D., Wainselboim, A., Zanutto, B. S. (2020).** A computational theory for the emergence of grammatical categories in cortical dynamics. Frontiers in Neural Circuits, Vol. 14, pp. 12. DOI: 10.3389/fncir.2020.00012.

7. **Dematties, D., Rizzi, S., Thiruvathukal, G. K., Wainselboim, A., Zanutto, B. S. (2019).** Phonetic acquisition in cortical dynamics, a computational approach. PLoS ONE, Vol. 14, No. 6, pp. e0217966. DOI: 10.1371/journal.pone.0217966.

8. **Dematties, D., Thiruvathukal, G. K., Rizzi, S., Wainselboim, A., Zanutto, B. S. (2020).**

Towards high-end scalability on biologically-inspired computational models. Parallel Computing: Technology Trends, IOS Press, Vol. 36, pp. 497 – 506. DOI: 10.3233/APC200077.

9. **Deza, A., Eckstein, M. P. (2016).** Can peripheral representations improve clutter metrics on complex scenes?. Advances in Neural Information Processing Systems, Vol. 29.

10. **Deza, A., Jonnalagadda, A., Eckstein, M. (2018).** Towards metamerism via foveated style transfer. arXiv preprint arXiv:1705.10041. DOI: 10.48550/arXiv.1705.10041.

11. **Deza, A., Konkle, T. (2021).** Emergent properties of foveated perceptual systems. arXiv preprint arXiv:2006.07991. DOI: 10.48550/arXiv.2006.07991.

12. **Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., Abbeel, P. (2016).** RLˆ2: Fast reinforcement learning via slow reinforcement learning. arXiv preprint arXiv:1611.02779.

13. **Fang, F., Ma, Z., Qing, L., Miao, J., Chen, X., Gao, W. (2008).** Face reconstruction using fixation positions and foveated imaging. 8th IEEE International Conference on Automatic Face & Gesture Recognition, pp. 1–6. DOI: 10.1109/AFGR.2008.4813393.

14. **Guo, M. H., Liu, Z. N., Mu, T. J., Hu, S. M. (2021).** Beyond self-attention: External attention using two linear layers for visual tasks. arXiv preprint arXiv:2105.02358.

15. **Hikosaka, O., Nakamura, K., Nakahara, H. (2006).** Basal ganglia orient eyes to reward. Journal of Neurophysiology, Vol. 95, No. 2, pp. 567–584. DOI: 10.1152/jn.00458.2005.

16. **Hsiao, J. H. W., Cottrell, G. (2008).** Two fixations suffice in face recognition. Psychological Science, Vol. 19, No. 10, pp. 998–1006. DOI: 10.1111/j.1467-9280.2008.02191.x.

17. **Ikeda, T., Hikosaka, O. (2003).** Reward-dependent gain and bias of visual responses in primate superior colliculus. Neuron, Vol. 39, No. 4, pp. 693–700. DOI: 10.1016/S0896-6273(03)00464-1.

18. **Jaramillo-Avila, U., Anderson, S. R. (2019).** Foveated image processing for faster object detection and recognition in embedded systems using deep convolutional neural networks. Conference on Biomimetic and Biohybrid Systems, Springer, Vol. 11556, pp. 193–204. DOI: 10.1007/978-3-030-24741-6_17.

19. **Jonnalagadda, A., Wang, W., Eckstein, M. P. (2021).** Foveater: Foveated transformer for image classification. arXiv preprint arXiv:2105.14173. DOI: 10.48550/arXiv.2105.14173.

20. **Kaplanyan, A. S., Sochenov, A., Leimkühler, T., Okunev, M., Goodall, T., Rufo, G. (2019).** Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos. ACM Transactions on Graphics (TOG), Vol. 38, No. 6, pp. 1–13. DOI: 10.1145/3355089.3356557.

21. **Kato, M., Miyashita, N., Hikosaka, O., Matsumura, M., Usui, S., Kori, A. (1995).** Eye movements in monkeys with local dopamine depletion in the caudate nucleus. I. Deficits in spontaneous saccades. Journal of Neuroscience, Vol. 15, No. 1, pp. 912–927. DOI: 10.1523/JNEUROSCI.15-01-00912.1995.

22. **Kowler, E. (2011).** Eye movements: The past 25 years. Vision Research, Vol. 51, No. 13, pp. 1457–1483. DOI: 10.1016/j.visres.2010.12.014.

23. **Melas-Kyriazi, L. (2021).** Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. arXiv preprint arXiv:2105.02723. DOI: 10.48550/arXiv.2105.02723.

24. **Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M. (2013).** Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.

25. **Purves, D., Augustine, G. J., Fitzpatrick, D., Hall, W. C., Matia, A.-S. L., Mcnamara, J. O., Williams, S. M. (2019).** Neurosciences,

chapter 11. De Boeck Supérieur, 6 edition, pp. 259–282.

26. **Ross-Sheehy, S., Reynolds, E., Eschman, B. (2020).** Evidence for Attentional Phenotypes in Infancy and Their Role in Visual Cognitive Performance. Brain Sciences, Vol. 10, No. 9, pp. E605. DOI: 10.3390/brainsci10090605.

27. **Spotorno, S., Malcolm, G. L., Tatler, B. W. (2014).** How context information and target information guide the eyes from the first epoch of search in real-world scenes. Journal of Vision, Vol. 14, No. 2, pp. 7–7. DOI: 10.1167/14.2.7.

28. **Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., et al. (2021).** Mlp-mixer: An all-mlp architecture for vision. Advances in Neural Information Processing Systems, Vol. 34, pp. 24261–24272.

29. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017).** Attention is all you need. Advances in Neural Information Processing Systems, Vol. 30.

Article received on 07/05/2022; accepted on 18/09/2022. Corresponding author is Dario Dematties.

ISSN 2007-9737

An Evaluation of a Ray-Tracing Based Model for Photorealistic Image Rendering of Confined Plasma in Stellarators

Luis Campos[1,2], Diego Jiménez[1,2], Silvio H. Rizzi[3], Esteban Meneses[1,2]

[1] National High Technology Center,
Costa Rica

[2] Costa Rica Institute of Technology,
Costa Rica

[3] Argonne National Laboratory,
United States

{lcampos,djimenez,emeneses}@cenat.ac.cr, srizzi@anl.gov

Abstract. As the world moves away from traditional energy sources based on fossil fuels, several alternatives have been explored. One promising clean energy source is nuclear fusion. The fusion of hydrogen isotopes may provide generous consumable energy gains. However, nuclear fusion reactors are not ready to become a productive mechanism yet. To get a better understanding of plasma, numerical simulations and scientific visualizations over high-performance computing systems are mandatory. The results from the simulations and a proper display of the data are key to design and tune up nuclear fusion reactors. It is also thanks to the international collaboration effort such as the advisory contribution and tools of researchers from the Argonne National Laboratory in the United States in conjunction with the National Center for High Technology of Costa Rica that this work was successfully carried out. In a previous work, we explored a new approach of the scientific visualization of plasma confinement, presenting one model to generate realistic plasma representations. This work presents an evaluation of the expected quality of the images rendered with the created model. We propose a concept called visual plausibility as an evaluation attribute to rate each rendered image by physicists that already know about the plasma appearance.

Keywords. Plasma fusion, simulation, stellarator, photorealism, visual plausibility, scientific visualization, ray tracing.

1 Introduction

Fusion energy is a promising source of clean energy for a future beyond fossil fuels. To realize the full potential of fusion energy, it is necessary to continue developing an understanding of both theory and engineering of particle fusion devices. In particular, computer-based tools are fundamental in studying how physical variables behave and what device design provides the highest energy production.

Stellarators are one particular plasma physics device that heats up a gas and uses coils to magnetically confine plasma and generate particle fusion. The Plasma Laboratory for Fusion Energy and Applications of the Costa Rica Institute of Technology recently developed a stellarator called SCR-1 [9]. Such reactor provides a research platform to extend our understanding of plasma physics and how future reactors should be designed and built.

Our team at the Costa Rica National High Technology Center has a long-standing collaboration with the plasma physicists to generate computer tools for the SCR-1 stellarator. In a joint effort, we developed a parallel computer simulator called BS-SOLCTRA [5] to collect several

doi: 10.13053/CyS-26-4-4435

data on different physics variables from plasma discharges. Additionally, we have also created a visualization model [2] to generate photorealistic images of plasma reactions. Those images are meant for scientific communication of fusion energy processes.

This paper provides an evaluation of the computer graphics model that generates photorealistic images of plasma phenomena. A selected group of people, deeply involved in plasma physics research, participated as evaluators of the images. We show the results of the evaluation and demonstrate how our model fulfills its goal in creating powerful images for science communication.

2 Background

2.1 Computer Graphics Model

This paper presents a methodology to evaluate images aimed at achieving photorealism, that is, simulating the appearance of a real photograph. In previous studies, we have developed image representations of plasma particles to check for simulation correctness, for example the Poincaré plots and magnetic fields maps [5]. In this work however, our interest lies not in evaluating images used to check for simulation correctness but in using and evaluating images whose main function is broader science communication. Meaning, communicating the results of an investigation to the interested community or gaining the attention of a non-technical/scientific audience.

After defining the images to be evaluated, the need arose to design a ray tracing model capable of generating high-quality photorealistic images that would give the observer a more accurate idea of how the stellarator-confined plasma looks. A previous study [2] had presented a model that generates images from the BS-SOLCTRA results.

This model uses the simulation results to convert the raw data into a mesh that represents the shape of the plasma and once that mesh is reconstructed it can be inserted as an object within the scene of any renderer. The results of the simulation consist of a set of files where each one represents all the steps that a particle had during the whole simulation. Each file uses comma-separated values to represent the 3 dimensional particle position at each iteration step. What is important is the selection of data necessary to successfully reconstruct the surface mesh.

It is thanks to Poincaré plots that we know that plasma surfaces are formed by the trajectory of a single particle. To build the desired surface it will be enough to find a suitable file. BS-SOLCTRA in its original version generates the position data of the particles that will form a surface, so the decision was to choose the largest file since it has the largest amount of data. Once the largest file is found it is converted to an $\langle x, y, z\rangle$ file format to be used as input by the surface reconstruction algorithm.

2.1.1 Screened Poisson Surface Reconstruction

The Poisson Surface Reconstruction is a well known technique for creating surface-objects from oriented point samples or particle data. This technique is resilient to noisy data and it fits very well for our purposes given that the input data is the same as our simulation results. The output of the algorithm is exactly what we need for the ray tracer, an object that represents the plasma last surface. Reconstructing 3D surfaces from point data is a well known problem in computer graphics. It allows fitting of extracted data from simulations, filling of surface holes or irregularities, and remeshing of existing models. The Poisson approach expresses surface reconstruction as the solution to a Poisson equation.

The Poisson algorithm takes the input data S being a set of samples $s \in S$, each s consisting of a point sp and an inward-facing normals $s.\vec{N}$, assumed to lie on or near the surface of an unknown model. The goal here is to reconstruct a triangular approximation to the surface by approximating the indicator function of the model and finally getting the isosurface [6].

The original algorithm adjusts the implicit function using a single global offset such that its average value at all points is zero. However, the presence of errors can cause the implicit function to drift so that no global offset is satisfactory.

The screened version instead seeks to explicitly interpolate the points [7].

The screened approach tries to modify the original Poisson to incorporate positional constraints. The associated Poisson equation is “screened” by a data fidelity term. In the algorithm context, the screening term means a soft constraint that encourages the reconstructed isosurface to pass through the input points. The difference with the first approach is that the position and gradient constraints are defined over different domain types.

The gradients are constrained over the full 3D space, positional constraints are introduced only over the input points, which lie near a 2D manifold. These two types of constraints, gradient and positional can be efficiently integrated, so that we can leverage the original multigrid structure to solve the linear system saving the significant overhead in space or time in the original way.

A requirement to run this algorithm is that each point had to have its own normal values in each $\langle x, y, z. \rangle$ axis. This normal calculation is a well-known algorithm and was achieved through Meshlab [3], a software that allows the manipulation of particles files or mesh files and the automation of these processes. Meshlab includes the implementation of the algorithm for calculating normals and a *screened Poisson* surface [7].

2.1.2 Lighting and Density Model

Once the extracted surface is added into the scene, the next step for the model is to add visual attributes to that object so that it looks similar to plasma. To get a better idea of what plasma should look like, real photos of confined plasma from different confinement chambers such as stellarators were analyzed. These photographs give an idea of the physical characteristics of the plasma, being like a gas suspended in the vacuum of the chamber with bright flashes of a color given by the gas used in the discharge.

The plasma seen in a real picture is similar to a gas and with a not much density, so what is behind from it can be seen, similar to what occurs to a translucent object. Another interesting attribute that can be seen in that last surface, is the light scattering happening in the surface, simulating its own light emission in blue and purple tones.

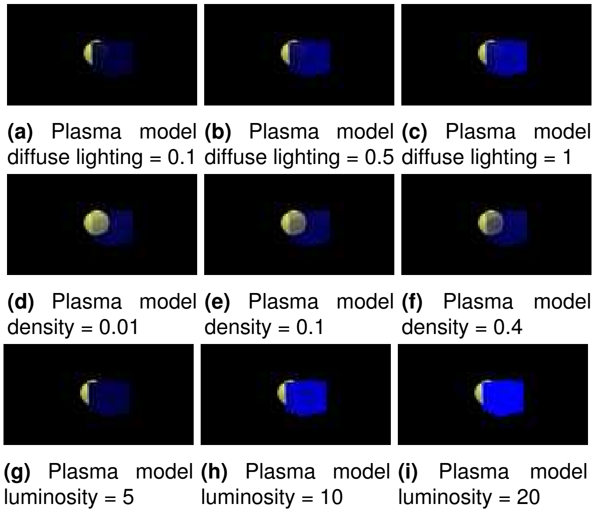
Figure 1 will be a guide to a better understanding of the effects of parameter changes in the final result. The cube will represent the plasma material and the red sphere will represent whatever is behind the plasma in the scene, since our goal is to see what is behind the plasma due to its translucency.

For the model creation, the work is based on modifying different parameters for each object to be described into the scene, these parameters will help to calculate both the surface and its interaction with light. Three of these parameters are called ambient, diffuse and specular lighting, referencing the Phong lighting model [8] used in ParaView. The three parameters are represented with numerical values between 0 and 1 to describe the contribution of each to illumination on the object surface. Those attributes will let us describe the light emission effect of the plasma by scattering the incoming light on the surface.

Ambient lighting brings a dim light to objects in the scene, simulating interaction with light from a very distant source, adding a bit of color to the object even without nearby lighting. The diffuse lighting simulates the directional impact that light has on an object, being the most significant visual component of the lighting model. The higher its value, the greater object visibility and brighter colors. Specular lighting is the light that reflects bright objects coming from the light source so the color is more associated with the light source than with the color of the object.

In our case, the specular reflection will be 0 because we don’t want any light source reflection in our plasma. We only want to reflect its own bright. Also the ambient light will not be needed because we only need the effect of the diffuse component combined with the opacity value and color. As it is shown in figures 1a, 1b, and 1c, we can notice more object light emission as we set a higher diffuse lighting value.

There are two attributes related to the object that will give the final appearance that we are looking for in the plasma, the opacity and luminosity of the object. The opacity of the object is a very important attribute for our representation since it is in charge



(a) Plasma model diffuse lighting = 0.1 **(b)** Plasma model diffuse lighting = 0.5 **(c)** Plasma model diffuse lighting = 1

(d) Plasma model density = 0.01 **(e)** Plasma model density = 0.1 **(f)** Plasma model density = 0.4

(g) Plasma model luminosity = 5 **(h)** Plasma model luminosity = 10 **(i)** Plasma model luminosity = 20

Fig. 1. Plasma model differences in diffuse illumination, density and luminosity attributes

of giving the effect of a translucent object and a little dense gaseous texture described above. The lower opacity value within our model, the lower the density of the object, so we define a value between 0.01 and 0.4 for a suitable translucent representation of plasma. In figures 1d, 1e, and 1f, the appearance is changed as we increment the opacity of an object.

On the other hand, any value greater than 0 in the luminosity of the object turns the object into a light emitter black body. The issue with this attribute is that this kind of objects lose their translucency and the result would lose an important element of the realistic appearance. This effect can be used to obtain an unrealistic but a different light-emitting representation of the plasma that can be used to catch the attention of the audience or similar cases. Figures 1g, 1h, and 1i show how the luminosity affects the material translucency and how a value higher than 0 in the luminosity affects how we see through the object.

2.1.3 Visual Plausibility

For the next step, in which the images resulting from the model are evaluated, it is necessary to create a concept to evaluate each image.

This concept tries to standardize a method of evaluating a set of images according to the attribute of realness. The mentioned concept is what we previously defined as “visual plausibility”, which is the quality of appearing visually reasonable or a probable representation of reality.

With this definition, what is wanted is that the evaluators only rate the appearance of the plasma and how plausible does it looks, since they are the ones who know how real it looks compared to a real photo. These images play an important role in communicating research results because this is how scientists explain to others what they are doing and why it is important to support research funding sources.

3 Evaluation Methodology

3.1 Qualitative Questionnaire

Evaluating these images should be done cautiously, given the amount of subjectivity involved in this process. Each evaluator could provide varying opinions about an image, making the process of coming to conclusions difficult.

The way of comparing them at a qualitative stage is a lot simpler and handy to our possibilities and it grants a scale based on the concept called Visual Plausibility on which the evaluators can rate each image. In this way, we can diminish the subjectivity of the evaluation.

For this matter, the evaluation tool needed must be capable of collecting and documenting information regarding our evaluators, like their knowledge, experiences and their backgrounds. On the other hand, it must provide a way so that any person could rate any presented image and also be capable of summarizing the results of the image evaluation.

As [4] mentioned, the answers to qualitative questionnaires consist of memories, opinions and experiences. This kind of questionnaires generate a rich material, useful for researchers from many disciplines.

This provided material is highly informative about various aspects of everyday life, and depending of how the question formulation is, the researchers are able to subtract any kind of information from the evaluators.

The method used is a qualitative questionnaire, which consists of collecting memories, opinions and experiences for a specific situation. Respondents answer the questions in the questionnaire based on these memories and experiences to give an opinion on what was asked. So our task is to correctly design the questions to collect their knowledge about the appearance of the plasma and the evaluations of the plasma model without interfering with their opinion.

According to [4] the use of qualitative questionnaires to produce research material has been criticized, not least because of the mentioned lack of representation from all social strata amongst the respondents. But, it is important to keep in mind that the material the tool generates can form the basis for generalization.

As with other qualitative methods, its strength lies in the deep insights that may be gained from the respondents. Such is the case in this study, where we are interested in the expert criteria to evaluate with their opinions our results, not to get a representation from all kind of people in the society.

3.2 Question Design

To develop the survey questions, different factors associated with it must be taken into account. Among the factors is the number of people, who to direct it to, number of questions, information you want to collect about the respondents, in addition to defining which images will be evaluated.

The goal about the amount of respondents was a number between 15 and 20 by recommendation of expert colleagues in the field of information of visualization. Regarding the target audience, the intention is to direct it to experts on plasma physics and even better if the person has had experience seeing plasma either physically or in a photograph.

These people are the right ones to evaluate a photograph that aims to render photorealistic images due to their previous knowledge and experiences added to the concept of visual plausibility created for the evaluation of the images.

The questionnaire was designed in such a way that with 5 questions before evaluating the images, it can be avoided that the respondents do not understand the concept of visual plausibility designed to evaluate and, on the other hand, it can be ensured that they are the appropriate people to make the evaluation. The first question aims to classify our evaluators in their relation with the plasma laboratory. This is why the first question was *What is your relation with plasma physics?*

The second and third question try to shed light on the respondents' context and experiences with plasma. The second question asks if they have worked with high temperature plasma discharges either directly or indirectly. The third question asks if they had ever seen a real picture of plasma confinement in devices like stellarators or tokamaks.

There is a fourth question that we asked to our evaluators. It tries to answer how they think the plasma should look like before starting to evaluate our images. In this way, we can get an idea of their expectations before we show a real image.

The fifth question was designed to ensure that the evaluators understand the concept of visual plausibility. We asked if they already know the concept of "plausibility" and remind them that if they are not familiarized with it, they can go back and read the survey introduction where it is defined.

Visual plausibility aims to recognize that although something isn't real, it looks very similar to how it looks in real life. In order to evaluate our images, we propose a scale from 1 to 10 to evaluate the visual plausibility of the picture, 1 being a picture with total lack of visual plausibility and 10 being a picture that looks very similar to a real one.

The first picture (Figure 2) to evaluate was rendered using PBRT in order to model plasma appearance using a similar approach to the result of this work. We extracted the surface from the particles and the object was the input for the ray tracer, this time to give that surface a plasma appearance we had to create a mixed material using glass material and scattering surface material. The result was not so satisfactory but an interesting approach that although not very realistic is worth evaluating in order to confirm that respondents can differentiate a representation

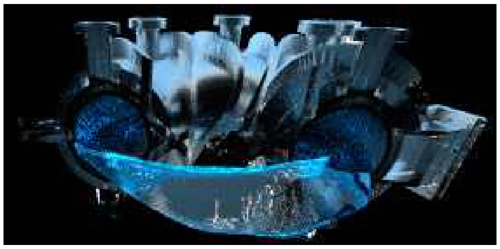


Fig. 2. First picture in evaluation

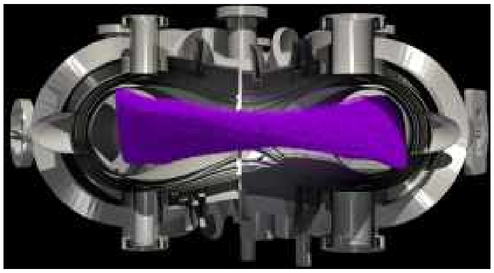


Fig. 3. Second picture in evaluation

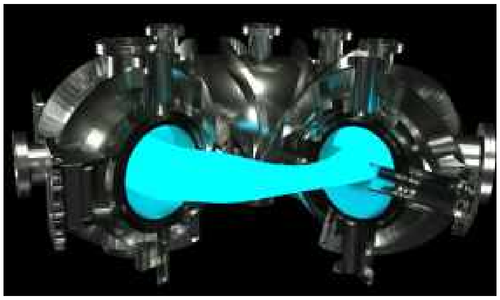


Fig. 4. Third picture in evaluation

with low visual plausibility from one with high visual plausibility.

The second picture (Figure 3) to be evaluated aims to do the same as the previous one, to show the respondent a plasma representation that is not intended to be realistic, so low ratings are expected for both this and the first representation. For the rendering of the image, Paraview[1] was used as software and the approach was to render the data of a simulation that calculated the trajectory of a million particles.

In Figure 4 we can see a variant of our solution. The idea was to make this model as flexible as possible and we found out a way to represent a black-body radiant object, so the plasma can irradiate light without the external light source.

This definitely was an attribute we wanted to have in our solution but the problem is that the ray tracer shows converts the object into a black body one and this means that it doesn't show any transparency at all, actually we cannot see through the volume, and this transparency is an important attribute that we definitely want to have into our final model.

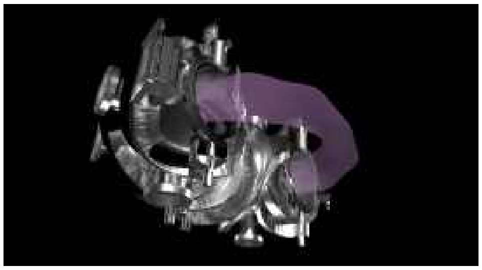
Just as we mentioned with the last two approaches that we didn't intend to be realistic or get a good score with these representation, this image doesn't pretended either. The intention is to try to evaluate different approaches that we face throughout the process of this work and that may be useful on other occasions.

To render this image, the luminosity model described in previous sections was used, which converts the object into a blackbody object that emits light. This is a way of representing plasma from a different perspective that is intended to impress the viewer but is unrealistic so a high score is not expected.

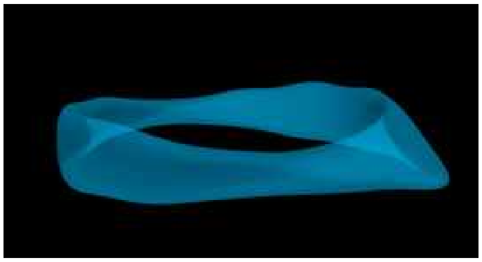
The last two images (Figure 5) are equivalent to the ones used in the questionnaire. These images differ slightly from the ones actually used, in the first one what changes from the one used in the survey is the angle from which the camera is positioned to obtain the image. On the second image, what changes is the color used, initially blue tones were used for testing and the final rendering was done in pink-purple tones according to the tones emitted by the gases used in reality.

These two images were obtained using our model, so that the evaluation of these images represents the evaluation of the proposed model. Figure 5a considers details such as the actual material of the stellarator device or at least a similar one, the density of the plasma, the translucency and the exact color according to the gas used in the plasma discharges.

And for the fifth image we thought that the chamber may confuse or distract some people from evaluating only the plasma appearance, so we added one more image from the pipeline without the chamber to completely appreciate the plasma shape, the emptiness of the volume and the color of the low density gas. Figure 5b shows a very similar representation of the image displayed in



(a) Fourth picture in evaluation (Reference)



(b) Fifth picture in evaluation (Reference)

Fig. 5. Final photorealistic plasma model (a), and final plasma surface in vacuum (b), both rendered with OSPRay

the last image evaluation question differing only in its color.

4 Evaluation Results

4.1 Respondents Evaluation

The intention with the first question was to give us an idea about our image evaluators, so that we could make sure that a vast majority are involved with plasma physics work and that they at least have knowledge on the subject. That is why the first question was *What is your relationship to plasma physics?* The result shows that half of the respondents are researchers from the plasma physics laboratory. The other half of them are divided between students and research assistants. Table 1 shows the results obtained for the first question.

The second and third questions help in understanding the context and experience of the respondents. The second question is whether they have worked with high-temperature plasma discharges, either directly or indirectly. The answer to this questions reveals how much our respondents know, since 17 out of 18 people answered that they have worked with plasma discharges, at high temperature directly or indirectly as we can see on Table 2.

The third question is a continuation of the previous question as it asks if they have ever seen a real picture of plasma confinement on devices like stellarators. Table 3 shows how 88.9% (16) have seen a real image before. That shows the experience of our testers, they have worked with plasma and they already know what a real image looks like, so they are the right evaluators to rate the images generated by the model by giving you a grade on their visual plausibility.

There is a fourth question we asked our testers, it tries to answer how they think the plasma should look before starting to test, this way we can get an idea of how they see the plasma before we show them a real image. The results confirm assumptions made from the beginning when we analyze the photographs of plasma discharges from different devices.

The responses have many matches between them, with words such as gas, bright, bright, region, shape, fuchsia, light, gradient, gaseous, flow, glow, light, luminous, intensity being among the majority of responses written by respondents. This shows that the initial idea of what the plasma would look like is not far from the experts conception of the same idea.

Finally with the fifth question we wanted to make sure that our respondents knew the concept designed to evaluate the images. So we asked them if they already knew the concept of ”plausibility” and reminded them that if they are not familiar with the concept, go back and read the introduction to the survey where the definition of visual plausibility was found. The table 4 shows the results and illustrates that half of the respondents, that is, 9 people, did not know the concept previously and that 2 more were not sure if they knew it or not.

Table 1. Results for first question “Select your relation with plasma physics”

Relation	Percentage	Count
Researcher	50%	9
Assistant	33,3%	6
Student	16,7%	3

Table 2. Results of second question: “Have you ever worked directly or indirectly with high temperature plasma discharges?”

Answer	Percentage	Count
Yes	94,4%	17
No	5,6%	1

Table 3. Results for third question: “Have you seen a real picture of plasma in confinement in Stellarator or Tokamak reactors?”

Answer	Percentage	Count
Yes	88,9%	16
No	11,1%	2

Table 4. Results of fifth question: “Did you know the concept of plausibility before taking this survey?”

Answer	Percentage	Count
Yes	38,9%	7
No	50%	9
Maybe	11,1%	2

Table 5. First Image Evaluation

Score	Percentage	Count
2	5.6%	1
3	5.6%	1
4	22.2%	4
5	16.7%	3
6	16.7%	3
7	11.1%	2
8	16.7%	3
9	5.6%	1

4.2 Image Evaluation

This part of the evaluation is the one that finally gave us the opinion of the expert criteria on the work we are presenting. Each expert evaluator rated from 1 to 10 the visual plausibility attribute of 5 different images shown to them. Our idea in this section was to compare the images generated by our work with images generated with other tools which did not present much realism since that was not the intention when they were generated. In this way, by comparing unrealistic representations with others that are much more realistic, we were sure that the concept of visual plausibility was understood and that although the images in the model are not hyper-realistic pictures, they do simulate a simple photograph with high-quality results.

For the first image of the evaluation we use an image rendered with a ray tracer called PBRT. This image is the result of a proof of concept to make the photorealistic model, which in our project was a dead end since the results did not meet the needs of the study. Still in this way the data obtained for the first image were shown in Table 5. The result of the evaluation of these approach was a low visual plausibility. With a mean of 5.65 and a standard deviation of 1.84 we got a non realistic result but still interesting because a lot of evaluators appreciate the translucency that the glassy material achieved and that glitter that plasma irradiates.

As we can see in Table 5 the 77,7% of the answers do not consider this approach with high visual plausibility. These results show how bad was the results using PBRT since these image was the most accurate representation we achieve using that ray tracer (mean score $x = 5.65$, standard deviation $s = 1.84$).

As we expected this representation was the lowest scored with a 4.2 mean for visual plausibility rate and a standard deviation of 2.33. It is important to remember that it lacks of reality because it does not tries to be a realistic representation. It shows a very accurate shape but without the translucency we are looking for, it does not looks like a gas or with low density volume, nor shows luminosity either. However, interestingly in Table 6 we can see how we have a result of 10 which is clearly an outlier which affects significantly to the calculation of the mean and the standard deviation (mean score $x = 3.8$, standard deviation $s = 1.99$).

Table 6. Second Image Evaluation

Score	Percentage	Count
1	16.7%	3
2	11.1%	2
3	11.1%	2
4	16.7%	3
5	16.7%	3
6	16.7%	3
8	5.6%	1
10	5.6%	1

Table 7. Third Image Evaluation

Score	Percentage	Count
2	5.6%	1
3	16.7%	3
5	22.2%	4
6	22.2%	4
7	22.2%	4
9	11.1%	2

Table 8. Fourth Image Evaluation

Score	Percentage	Count
1	5.6%	1
2	5.6%	1
6	11.1%	2
7	5.6%	1
8	44.4%	8
9	27.8%	5

Table 9. Fifth Image Evaluation

Score	Percentage	Count
1	5.6%	1
5	5.6%	1
7	16.7%	3
8	44.4%	8
9	27.8%	5

Table 7 shows the results of the evaluation of the third figure. As the table shows, with a mean of 5.55 and a standard deviation of 1.87 we had a 88.9% of the evaluators that considers the image has low visual plausibility. Although the image is not a realistic one, they see the value in this representation for showing results and cause a good impression in the audience (mean score $x = 5.55$, standard deviation $s = 1.88$).

As showed in Table 8 with a mean of 8 and a standard deviation of 0.94 we had an impressive result of the 72.2% of the answers considers it with high visual plausibility. This is the second highest score we got from our evaluators and the first evaluation for our final resulting model. These results were obtained by filtering the data, eliminating those outliers that, as in the second question, significantly modified the calculated data.

We can see how the mean value is around 8, however a vote of 1 and another one of 2 negatively affect the rating, which could be due an evaluation misunderstanding. The value calculations counting the outliers was lower (mean score $x = 7.28$, standard deviation $s = 2.30$) than without those outliers answers (mean score $x = 8$, standard deviation $s = 0.94$).

The satisfactory results shows the highest score in the evaluation. With a mean of 7.94 and a standard deviation of 0.99, we got the same 72.2% of the evaluators who consider it has high visual plausiblity. Using the same logic of the fourth image evaluation, we can see in Table 9 how a vote of 1 affects the rating mean decreasing it and increases the standard deviation, which could be due a evaluation misunderstanding.

The value calculations counting the outlier were lower (mean score $x = 7.58$, standard deviation $s = 1.87$) than without the outlier data (mean score $x = 7.94$, standard deviation $s = 0.99$).

5 Final Remarks

5.1 Conclusions

To make particle fusion a productive energy source, it is imperative to continue developing simulation and visualization tools that help scientists and engineers build efficient fusion devices. Scientific visualization has a twofold contribution.

First, it provides researchers with a powerful tool to deeply study complex phenomena. In the case of particle fusion, visualizations help in understanding the behavior of variables of interest. Second, scientific visualizations provide a communication tool for a broader audience.

Having people well informed about scientific endeavors is key in sustaining the public investment on groundbreaking research.

The result of this work presents a contribution in the way of evaluating works where the result is a series of images that must be qualified qualitatively. This evaluation method is very similar to the one used in information visualization contexts, where visualizations are made and opportunities for improvement are discussed at a qualitative level.

The results of this work were evaluated under a design methodology that exposes how much experience the respondents had working with plasma and if they had had contact with plasma images, which assured us that they were the appropriate experts to evaluate our images. The results obtained confirm that they comply with the proposed hypothesis, so the images generated by our model obtained a high visual plausibility score according to the expert criteria.

5.2 Future Work

Using this work as a starting point for future work, we consider that different efforts can be made to make the visualization tool much more realistic or useful for error checking. New visualization tools could be adopted, such as in-situ visualization to obtain high-quality images while the simulation is running. The possibility of generating frames of these visualizations and producing animations could also be explored.

We also consider a qualitative improvement could happen by adding a functionality in the ray tracer used, OSPRay in our case, so that the luminosity attribute can be used mixed with the opacity attribute of the object.

Hence, it would be possible to represent the object as a light emitter object but that it is not a black-body object. In other words, it would emit light without losing the characteristic of being translucent. By achieving this form of representation effectively, the physical aspect of the plasma can be represented in a much more realistic way.

Acknowledgments

This work used resources of the Argonne Leadership Computing Facility, which is a U.S. Department of Energy, Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

1. **Ahrens, J., Geveci, B., Law, C. (2005).** Paraview: An end-user tool for large data visualization. The Visualization Handbook, Vol. 717, No. 8.

2. **Campos-Duarte, L., Jiménez, D., Meneses, E., Solano-Piedra, R., Pérez, E., Vargas, V., Rivera-Alvarado, E. (2021).** Towards photorealistic visualizations for plasma confinement simulations. Practice and Experience in Advanced Research Computing, PEARC'21, Association for Computing Machinery, , No. 23, pp. 1–4. DOI: 10.1145/3437359.3465608.

3. **Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., Ranzuglia, G. (2008).** Meshlab: an open-source mesh processing tool. Eurographics Italian Chapter Conference, Salerno, Vol. 2008, pp. 129–136.

4. **Eckerdal, J. R., Hagström, C. (2017).** Qualitative questionnaires as a method for information studies research. Information Research, Vol. 22, No. 1.

5. **Jiménez, D., Campos-Duarte, L., Solano-Piedra, R., Araya-Solano, L. A., Meneses, E., Vargas, I. (2019).** Bs-solctra: Towards a parallel magnetic plasma confinement simulation framework for modular stellarator devices. Latins American High Performance Computing Conference, Springer, Cham, Vol. 1087, pp. 33–48. DOI: 10.1007/978-3-030-41005-6_3.

6. **Kazhdan, M., Bolitho, M., Hoppe, H. (2006).** Poisson surface reconstruction. Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Vol. 7, pp. 61–70.

7. **Kazhdan, M., Hoppe, H. (2013).** Screened poisson surface reconstruction. ACM Transactions on Graphics (ToG), Vol. 32, No. 3, pp. 1–13. DOI: 10.1145/2487228.2487237.

8. **Phong, B. T. (1975).** Illumination for computer generated pictures. Communications of the ACM, Vol. 18, No. 6, pp. 311–317. DOI: 10.1145/360825.360839.

9. **Solano-Piedra, R., Vargas, V. I., Köhn, A., Coto-Vílchez, F., Sanchez-Castro, J., López-Rodríguez, D., Rojas-Quesada, M., Mora, J., Asenjo, J. (2017).** Overview of the SCR-1 stellarator. 23rd IAEA Technical Meeting on the Research Using Small Fusion Devices.

Article received on 06/05/2022; accepted on 16/09/2022.
Corresponding author is Luis Campos.

ISSN 2007-9737

Cardiovascular Disease Detection Using Machine Learning

Rodrigo Ibarra, Jaime León, Iván Ávila, Hiram Ponce

Universidad Panamericana,
Facultad de Ingeniería,
Mexico

{0252692,0251991,0252002,hponce}@up.edu.mx

Abstract. The detection of Cardiovascular Diseases (CVDs) prematurely is of great interest for the Healthcare Industry. According to the World Health Organization, heart diseases represent 32% of global deaths by 2019. In this work, we propose building an interpretable machine learning model to detect CVDs. For this, we use a public dataset consisting of over 320 thousand records and 279 features. We explore the performance of three well-known classifiers and we build them using hyper-parameter techniques. For interpretability, feature relevance is tested. After the experimental results, we found Random Forest to performed the best with 94% of accuracy and 81% of area under the ROC curve. We also implement an easy web application as a tool for detecting CVDs using relevant features information.

Keywords. Machine learning, classification, heart disease.

1 Introduction

Detecting Cardiovascular Diseases (CVDs) prematurely is of great interest for the Healthcare Industry. According to the World Health Organization (WHO), almost 18M deaths, representing 32% of global deaths in 2019 were caused by heart diseases, as part of CVDs [6]. Furthermore, 75% of these cases took place in low or middle income countries, not to mention that 38% of premature deaths (under the age of 70) were caused by CVDs [6].

On top of that, Centers for Disease Control and Prevention (CDC) claim that about half of USA's population has at least one of three risk factors for CVD [9]: high blood pressure, high cholesterol and smoking. With that in mind, CDC [3] surveyed American citizens all over the country which led to a data set containing almost 400k observations, whether the person in question has ever been diagnosed either with Coronary Heard Disease or Myocardial Infarction.

Even though several approaches on how to detect these kind of disease using Machine Learning (ML) have been developed through out recent times, in this work we aim to find a robust yet interpretable way of finding out which risk factors are more strongly correlated with CVDs, particularly Coronary Heart Disease and Myocardial Infarction.

To pursue our goal, we use a public dataset from the CDC to test three different ML models. We defined the importance of features in order to select the most relevant ones. Through a benchmark among these models, we determined the best model. Lastly, we implemented the best model in an easy application that can be used in preventing the detection of CVDs.

Here we aim not only to find an accurate predictor but to examine which risk factors are more closely related to CVD detection, leading to prevention or better and early management.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the methodology implemented in this work, from the data collection, data preparation, training models, and evaluation of models.

Section 4 shows the experimental results and the implementation in an application. Lastly, Section 5 concludes the paper.

doi: 10.13053/CyS-26-4-4422

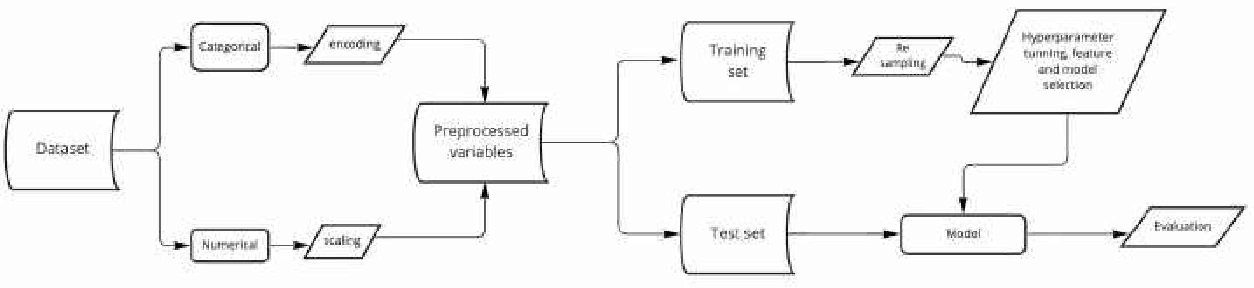


Fig. 1. An overview of the ML workflow implemented in this work

2 Related Work

Various efforts have been made towards finding out which is the best, most accurate model for CVD detection. In 2018, the work in [8] showed how Decision Trees might not be a great candidate by themselves as it's quite hard not to over-fit, while Support Vector Machines (SVM) can easily outperform Decision Trees and Random Forest or Ensemble Methods can perform well too.

Complementing these results, the authors in [5] have shown how different models performed better while coupled with different over sampling techniques. Particularly, it was found that Synthetic Minority Oversampling was the best match for Random Forest, SVM and Random Over Sampling work great together, too. Also, Adaptive Synthetic Sampling helped, once again, Random Forest gets the best results.

The latter is very relevant as it turns out some other works have been done around finding out which tree-based models work better on CVD detection. Using a data set from UCI Machine learning repository, the authors in [7] found that J48 is the best technique for CVD detection amongst some other well-known tree-based methods.

3 Methodology

In this work, we implement the overall workflow for achieving the CVD estimation using ML models, depicted in Fig. 1. This workflow consists of four main steps: data collection, data preparation, training models, and evaluation of models.

Table 1. Categorical variables in the dataset

Feature	Categories	Top Category	Frequency
HeartDisease	2	No	292422
Smoking	2	No	187887
AlcoholDrinking	2	No	298018
Stroke	2	No	307726
DiffWalking	2	No	275385
Sex	2	Female	167805
AgeCategory	13	65-69	34151
Race	6	White	245212
Diabetic	2	No	269653
PhysicalActivity	2	Yes	247957
GenHealth	5	Very Good	113858
Asthma	2	No	276923
KidneyDisease	2	No	308016
SkinCancer	88	788	289976

3.1 Dataset Description

The dataset used for this study, was downloaded from Kaggle repository. Originally, it comes from the CDC's (Centers for Disease Control and Prevention) Behavioral Risk Factor Surveillance System [2] which conducts telephone surveys about the health status of USA's citizens and contained around 300 features. It has been pre-processed and cleaned. So that in this work, we use the smaller data set with 18 features.

The dataset is divided into 13 categorical variables, 4 numerical variables and one categorical target ('HeartDisease'), consisting of 319,795 records without null values; but it is heavily unbalanced. Table 1 represents the frequency of the most common category on each categorical feature, while Fig. 2 represents the distribution of the four numerical variables.

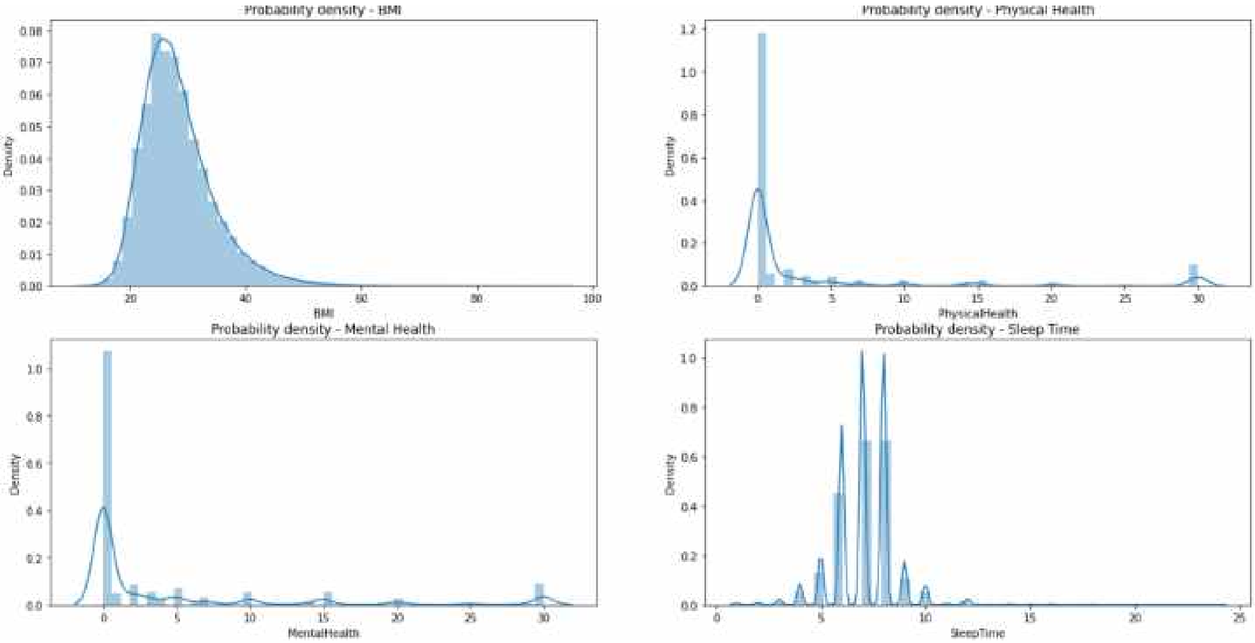


Fig. 2. Probability density for the numerical variables in the dataset

Figure 3 shows an example of the relationship between the variables and the target. We know that the number of observations are 319,795, of those 292,422 belong to the negative class. Thus, it is important to balance the data.

3.2 Data Pre-Processing

As we are dealing with both categorical and numerical variables on top of an imbalanced dataset, we must have at least three pre-processing tasks to do:

— Perform One-Hot encoding on categorical variables.

— Perform standard scaling to numerical variables.

— We use over sampling and under sampling to balance classes.

For feature preparation, we decided to use the sklearn pre-processing module on Python, OneHotEncoder and StandardScaler, to perform the first part of the process. Then, for re-sampling, we have used RandomOverSampler and RandomUnderSampler for balancing the classes. And, finally, we split the data in training and holdout sets.

After the EDA we know the data types of the variables, we created a pipeline. The first step was to normalize all numeric features using an standard scaler, for categorical values we used One-Hot encoding, and last we used a label encoder for the target feature. Imputing of null values was not necessary because we did not have any empty values.

Next, we separated the dataset in 80% of training data and 20% of testing data. It is remarkable to say that we stratified the data to assure all the target values were distributed in both train/test sets.

The third step consisted on balancing the target feature because originally we had a ratio of 91 vs 9 of the negative and positive classes respectively. We did a random over-sampling to get a relationship of $70 - 30$ and a random under-sampling to get a $50 - 50$ ratio. Only the train set was balanced.

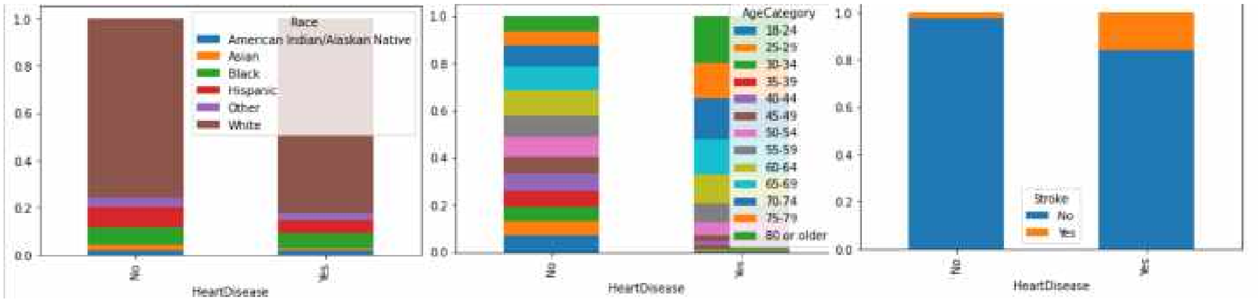


Fig. 3. Target relationship among other variables showing data unbalancing

3.3 Building Models

We decided to test three different model classifiers: Stochastic Gradient Descent Based (SGD) Classifier, Logistic Regression, and Random Forest.

We decided to use Optuna [1] for hyper-parameter tuning, which is an open source hyper-parameter optimization framework to automate hyper-parameter search more efficiently. One of the key reasons we decided to use this library is because it is able to do automated search for optimal hyper-parameters using Python conditionals, loops, and syntax. It also uses state-of-the-art algorithms which help to efficiently search large spaces and prune unpromising trials for faster results. In the basis, it uses Bayesian optimization that builds a probability model of the wrapped objective function and uses it to select hyper-parameters to evaluate in the true objective function.

For the SGD Classifier, we use four different hyper-parameters such as alpha, penalty, loss, and max_iter. For the Logistic Regression model the hyperparameters used were: penalty, c value, solver, and fit_intercept. Lastly for the Random Forest Classifier, it used four different hyper-parameters such as max_depth, n_estimators, criterion, and max_features.

For hyper-parameter tuning, we implement a 3-fold 20-repetition cross validation technique. Table 2 summarizes the best hyper-parameters per model.

The following graphs shows the objective value for each trial ran by the optuna objective function, in this case, it only shows the top 10 trials. We can see that it starts with an objective value around 0.76 and after the fourth trial, it shows the best objective value with a value around 0.915.

Figure 4 shows an example of the evolution of the model accuracy (objective function) based on the trials of the different combinations of the hyper-parameters. Another representation of the hyper-parameter evolution under the optimization process can be seen in Fig. 5. In addition, Fig. 6 shows an example of the influence of the hyper-parameters in the model accuracy (objective function). The latter would be interesting for better understanding on the role of the hyper-parameters in the building of the models.

3.4 Experimentation

In order to find the best algorithm, we first ran the hyper-parameter optimization process with Optuna. Then, the best classifier is selected in terms of the accuracy metric (1), where, TP, TN, FP, and FN represent true positive, true negative, false positive, and false negative, respectively:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \qquad (1)$$

In addition, we plot the ROC (Receiver Operator Characteristic) curve [10], which helps us understand the trade-off between sensitivity and specificity. Classifiers that give curves closer to the top-left corner indicate a better performance.

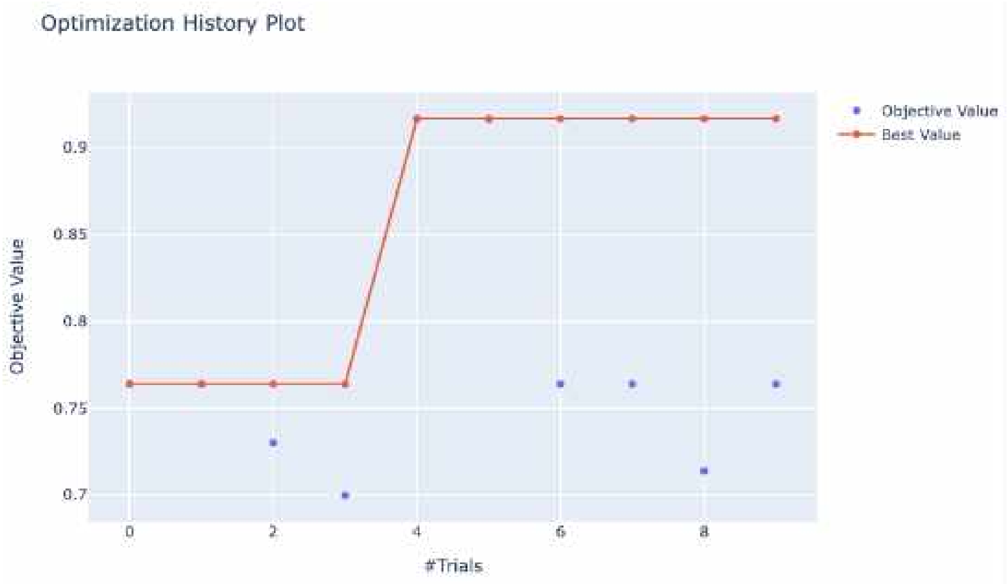


Fig. 4. Optimization history plot for the objective value in the Random Forest model

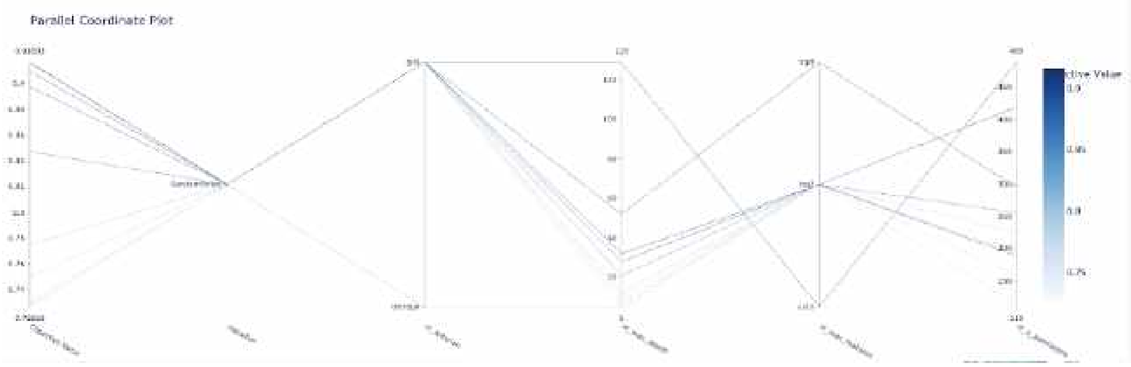


Fig. 5. Parallel coordinate plot in the Random Forest model

4 Experimental Results

We conducted the training of the three model classifiers, and results are summarized in Table 3. It can be observed that the Random Forest model gets the best accuracy metric (94%).

Figure 7 shows the ROC curve of the Random Forest model. We find that our model has an AUC (Area Unders Curve) of 0.81, which is equivalent to the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance. Note that the ROC does not depend on the class distribution. This makes it useful for evaluating classifiers predicting rare events such as ours.

We also did a feature importance graph to find which variables are the most important ones for the model. Out of 41 different features, Fig. 8 only shows the fourteen most important features, such as body mass index (BMI), sleep time, physical health, mental health, among others.

Furthermore, we created an application on Gradio library of Python in order to create a functional application to show our model. In

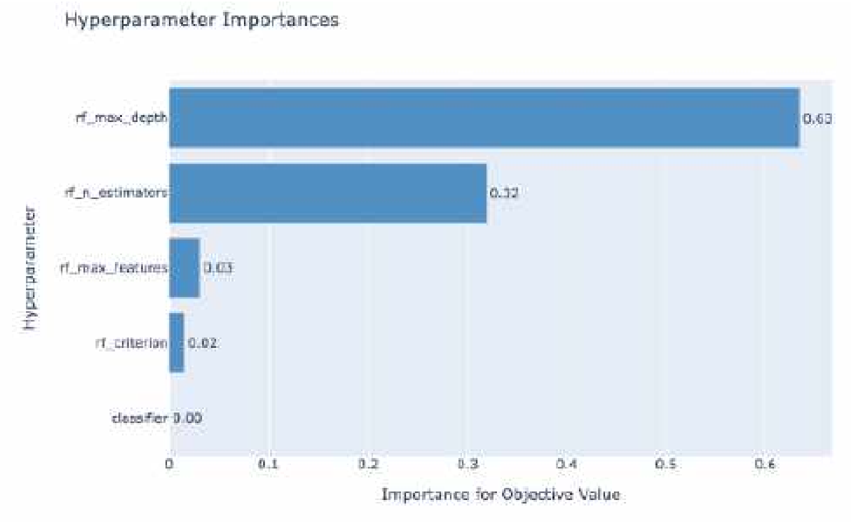


Fig. 6. Hyper-parameters importance in the Random Forest model

Table 2. Optimized hyper-parameters for the models

Model	Hyperparameters

SGD Classifier	alpha=3.81E-5,
	penalty='elasticnet',
	max_iter=60

Logistic Regression	regularization=L2,
	reg_coefficient=45.52

Random Forest	max_depth=38,
	n_estimators=299,
	criterion=gini,
	max_features='auto'

Table 3. Results of the training models

Model	Accuracy (%)
SGD Classifier	76
Logistic Regression	80
Random Forest	94

the application, the user inputs the required parameters using a friendly interface, and those inputs are processed in order to obtain a quick result of the heart disease prediction of the person. Figure 9 shows an excerpt of the web application running the Random Forest classifier.

The results validate that our proposal consists of an ML model, to say Random Forest classifier, that is highly accurate (94% of accuracy) and robust enough for a medical application (81% of AUC). Furthermore, we find that the automatic detection of feature relevance (see Fig. 8) is consistent with the literature, in which BMI, sleep time, and physical health are the most common risk factors in CVDs [4].

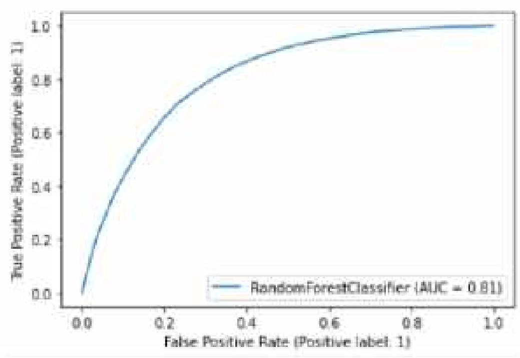


Fig. 7. ROC curve of the Random Forest model

In addition, we developed a web application tool based on the Random Forest classifier for heart disease estimation, leading to prevention or better and early management of the disease. This work is limited on the data used, since it looks biased in terms of gender and ethnicity, and the dataset is unbalanced too. So, it would be better to expand the investigation with more detailed and unbiased

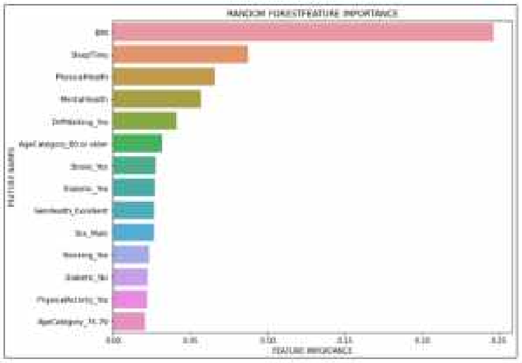


Fig. 8. Feature importance using the Random Forest model

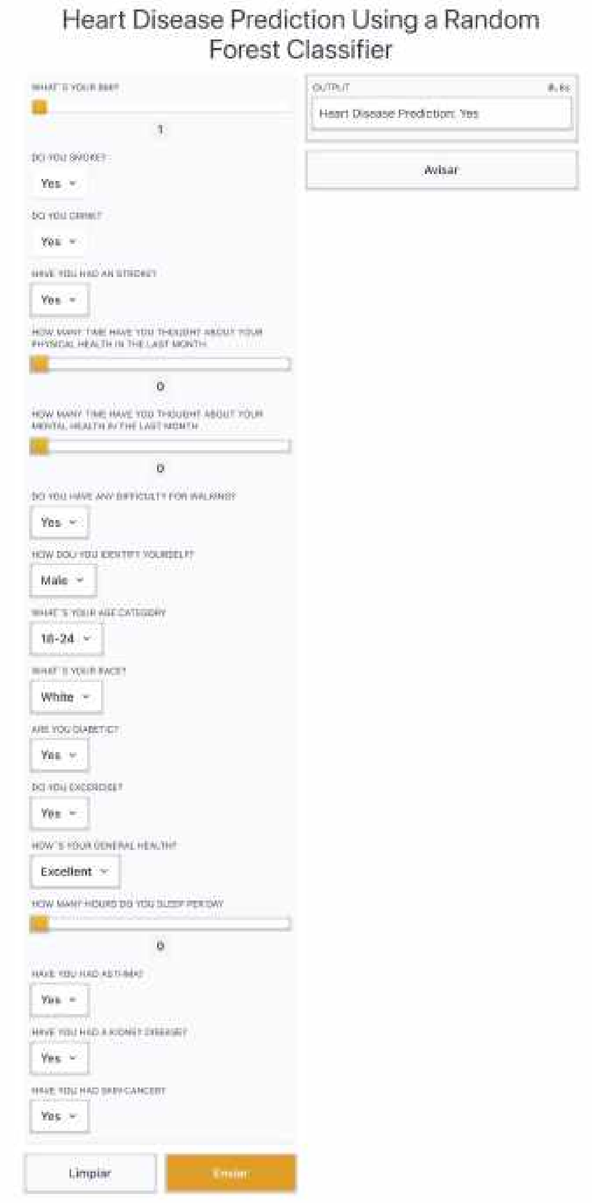
data. However, this work can confirm that this preliminary work might help facing CVD in patients.

5 Conclusions

This work proposed to build a robust yet interpretable way of finding out which factors are more strongly correlated with CVDs using machine learning models. To do so, we explored three ML classifiers, and we found that Random Forest model is the best for the dataset used.

It is evident that imbalance is a big problem for this data set on top on non-separability. Other techniques for imbalanced learning can be applied to this problem like class weights, using gradient boosting for a more robust sequential tree ensemble that would help better differentiate the binary classes. Using these techniques should result in better classification performance that could help in many cases. For instance, a Hospital could survey patients and find out whether they are candidates for CVD prevention treatment or feature importance could help pharmaceutical companies target specific drugs on their campaigns or simply.

Even though our model has a good performance, the model needs improvement. For future work, we can use other ML techniques and adding more features to our model for better complexity and understanding of the patient's information. Other data sets can also be explored.



Heart Disease Prediction Using a Random Forest Classifier

WHAT´S YOUR BMI?
1

DO YOU SMOKE?
Yes

DO YOU DRINK?
Yes

HAVE YOU HAD AN STROKE?
Yes

HOW MANY TIME HAVE YOU THOUGHT ABOUT YOUR PHYSICAL HEALTH IN THE LAST MONTH
0

HOW MANY TIME HAVE YOU THOUGHT ABOUT YOUR MENTAL HEALTH IN THE LAST MONTH
0

DO YOU HAVE ANY DIFFICULTY FOR WALKING?
Yes

HOW DOU YOU IDENTIFY YOURSELF?
Male

WHAT´S YOUR AGE CATEGORY
18-24

WHAT´S YOUR RACE?
White

ARE YOU DIABETIC?
Yes

DO YOU EXCERCISE?
Yes

HOW´S YOUR GENERAL HEALTH?
Excellent

HOW MANY HOURS DO YOU SLEEP PER DAY
0

HAVE YOU HAD ASTHMA?
Yes

HAVE YOU HAD A KIDNEY DISEASE?
Yes

HAVE YOU HAD SKIN CANCER?
Yes

OUTPUT
Heart Disease Prediction: Yes

Fig. 9. Example of the web application running the Random Forest classifier for CVD prediction

References

1. **Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M. (2019).** Optuna: A next-generation hyperparameter optimization framework. Proceedings of

the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

2. **CDC (2015).** Centers for disease control and prevention.

3. **Covid, C., Team, R., COVID, C., Team, R., Chow, N., Fleming-Dutra, K., Gierke, R., Hall, A., Hughes, M., Pilishvili, T., et al. (2020).** Preliminary estimates of the prevalence of selected underlying health conditions among patients with coronavirus disease 2019—United States, february 12–march 28, 2020. Morbidity and Mortality Weekly Report, Vol. 69, No. 13, pp. 382–386. DOI: 10.15585/mmwr.mm6913e2.

4. **Khan, S. S., Ning, H., Wilkins, J. T., Allen, N., Carnethon, M., Berry, J. D., Sweis, R. N., Lloyd-Jones, D. M. (2018).** Association of body mass index with lifetime risk of cardiovascular disease and compression of morbidity. JAMA Cardiology, Vol. 3, No. 4, pp. 280–287. DOI: 10.1001/jamacardio.2018.0022.

5. **Lakshmanarao, A., Swathi, Y., Sundareswar, P. S. S. (2019).** Machine learning techniques for heart disease prediction. International Journal of Scientific & Technology Research, Vol. 8, No. 11, pp. 374–377.

6. **Mensah, G. A., Roth, G. A., Fuster, V. (2019).** The global burden of cardiovascular diseases and risk factors: 2020 and beyond. Journal of the American College of Cardiology, Vol. 74, No. 20, pp. 2529–2532.

7. **Patel, J., Khaked, A. A., Patel, J., Patel, J. (2021).** Heart disease prediction using machine learning. Proceedings of Second International Conference on Computing, Communications, and Cyber-Security, Springer, Vol. 203, pp. 653–665. DOI: 10.1007/978-981-16-0733-2_46.

8. **Ramalingam, V. V., Dandapath, A., Raja, M. (2018).** Heart disease prediction using machine learning techniques: A survey. International Journal of Engineering & Technology, Vol. 7, pp. 684. DOI: 10.14419/ijet.v7i2.8.10557.

9. **Van Bussel, E. F., Hoevenaar-Blom, M. P., Poortvliet, R. K. E., Gussekloo, J., van Dalen, J. W., Van Gool, W. A., Richard, E., van Charante, E. P. M. (2020).** Predictive value of traditional risk factors for cardiovascular disease in older people: a systematic review. Preventive Medicine, Vol. 132, pp. 105986. DOI: 10.1016/j.ypmed.2020.105986.

10. **Zhou, Y., Zhang, J., Liu, R. H., Xie, Q., Li, X. L., Chen, J. G., Pan, X. L., Ye, B., Liu, L. L., Wang, W. W., et al. (2021).** Association between health-related physical fitness and risk of dyslipidemia in university staff: A cross-sectional study and a roc curve analysis. Nutrients, Vol. 14, No. 1, pp. 50. DOI: 10.3390/nu14010050.

Article received on 10/06/2022; accepted on 19/08/2022.
Corresponding author is Hiram Ponce.

ISSN 2007-9737

A Domain Specific Parallel Corpus and Enhanced English-Assamese Neural Machine Translation

Sahinur Rahman Laskar[1], Riyanka Manna[2], Partha Pakray[1], Sivaji Bandyopadhyay[1]

[1] National Institute of Technology Silchar,
Department of Computer Science and Engineering,
India

[2] Adamas University,
Department of Computer Science and Engineering,
India

{sahinurlaskar.nits, riyankamanna16, parthapakray, sivaji.cse.ju}@gmail.com

Abstract. Machine translation deals with automatic translation from one natural language to another. Neural machine translation is a widely accepted technique of the corpus-based machine translation approach. However, an adequate amount of training data is required, and there is a need for the domain-wise parallel corpus to improve translational performance that shows translational coverages in various domains. In this work, a domain-specific parallel corpus is prepared that includes different domain coverages, namely, Agriculture, Government Office, Judiciary, Social Media, Tourism, COVID-19, Sports, and Literature domains for low-resource English-Assamese pair translation. Moreover, we have tackled data scarcity and word-order divergence problems via data augmentation and prior alignment concept. Also, we have contributed Assamese pretrained LM, Assamese word-embeddings by utilizing Assamese monolingual data, and a bilingual dictionary-based post-processing step to enhance transformer-based neural machine translation. We have achieved state-of-the-art results for both forward (English-to-Assamese) and backward (Assamese-to-English) directions of translation.

Keywords. English-Assamese, low-resource, neural machine translation, parallel corpus, data augmentation, prior alignment, language model.

1 Introduction

Machine translation (MT) is a sub-field of natural language processing (NLP) that helps to bridge gaps in communication via automatic translation without human assistance. With the advancement of deep learning techniques, machine translation technique, namely, neural machine translation (NMT) shows remarkable translation accuracy [3, 26]. The NMT is a corpus-based approach of MT, which requires large amount of bilingual corpus for training a NMT model to achieve a good translation performance.

However, the adequate amount of training data is a challenging issue for low-resource settings [19]. Generally, low-resource pairs are considered if the training amount of parallel data is less than 1 million [16]. For instance, English–Mizo (En-Mz) [33, 21, 15], English–Assamese (En-As) [23, 24], English–Khasi (En-Kha) [22] are the examples of low-resource pairs.

The majority languages of the worldwide can be considered as “low-resource” based on the availability resources [29, 36]. Furthermore, the precise definition of “low-resource language pair” is a research question itself since the morphological rich low-resource languages in addition to the presence of varieties of inflected words, require more bitext data to achieve equivalent translation performance of languages that have less inflected words [7].

Moreover, NMT shows weakness in case of out-domain data [19], which demand to develop domain specific parallel corpus to improve low-resource pair translation.

doi: 10.13053/CyS-26-4-4423

In this paper, we have investigated a low-resource pair “En-As” to improve NMT for both directions, En-to-As and As-to-En translation. From the linguistic aspects, En and As are very different to each other, for instance, unlike En [23], as follows subject-object-verb (SOV), morphological rich language and adopts Assamese-Bengali script [28] originated from the Gupta script [8]. Our contributions are summarized as follows:

— We have created a domain specific En-As parallel corpus, which covers various domains, namely, social media, agriculture, Government office, judiciary, sports, tourism, COVID-19 and literature.

— We have addressed data scarcity and word-order divergence problems to enhance NMT for En-As language pair translation. By utilizing monolingual As data, synthetic En-As parallel sentences are prepared and extracted phrase pairs from the original parallel sentences (train set).

  To tackle the data scarcity issue, the extracted phrase pairs are augmented to the original parallel data and leveraging synthetic parallel data in the training model via two steps process: pretrain on the train data with synthetic parallel data and then fine-tuned on the train data without synthetic parallel data.

  Moreover, we have utilized pretrained multilingual contextual embeddings-based alignment technique to extract alignment information and that is used as prior alignment information during the training phase to tackle the word-order divergence issue.

— We have contributed an Assamese pretrained language model (AsLM) and word-embeddings vectors (AsGloVe) that shall be used in various downstream NLP tasks of Assamese language. The AsLM and AsGloVe are used for the improvement of En-As NMT.

— We have contributed a bilingual dictionary of En-As that is used in the post-processing step to tackle out-of-vocabulary issue and enhance En-to-As and As-to-En translations.

— We have achieved state-of-the-art results for low-resource En–As MT translational performance in terms of automatic and manual evaluation.

The rest of the paper is structured as follows: Section 2 discuss background concept and the related works. The domain specific parallel corpus and dataset description is presented in Section 3. Section 4 reported the baseline system results. Section 5 and 6 describe the proposed approach and reported results with analysis. Lastly, Section 7 conclude the paper with future scopes.

2 NMT Background and Related Work

Statistical machine translation (SMT) and NMT are two well-studied corpus-based MT techniques in the MT. To enhance low-resource pair translation, researchers have started experimenting with NMT recently. In this section, we have discussed the fundamentals of NMT and also emphasizes earlier research on English-Assamese MT.

2.1 NMT

The corpus-based (also known as data-driven) approach of NMT introduces RNN-based encoder-decoder architectures, where seq-2-seq learning is achievable by tackling variable length phrases of source-target sentences [3, 26].

To learn the long-term features of the source and target words for encoding and decoding, long short-term memory (LSTM) has demonstrated remarkable performance in this case. When encountering too lengthy sentences, it is unable to encode all the necessary information.

For that reason, the attention mechanism has been introduced in NMT [3, 26] that enables the decoder to take into account various segments of the source sequence during various decoding steps.

In the encoder-decoder based NMT, the encoder is responsible for the encoding of input sequences $sr_1, sr_2 \ldots sr_n$ and generates a vector U.

Whereas, the decoder decodes the output $tr_1, tr_2 \ldots tr_m$ using computation of condition probability, as given in Eq. (1):

$$P(tr \mid sr) = \sum_{i=1}^{m} P\left(tr_i \mid tr_{<1},\ U\right). \quad (1)$$

Using Eq. (2), the value of at_o correlates to the frequency of time steps in the input sentence. Therein, in the source side (h_i) and target side (h_o), the series of hidden states are computed, which are finally correlated to produce the attention vector at_o:

$$at_o = \frac{\text{exp}\left(\text{score}\left(h_o,\ h_i'\right)\right)}{\sum_{i'} \text{exp}\left(\text{score}\left(h_o,\ h_{i'}\right)\right)}. \quad (2)$$

The general estimate of score function defined in Eq. (3) is considered in this work for the preliminary experiments of the baseline system:

$$\text{score}\left(h_o,\ h_i'\right) = h_o\ W_a\ h_i'. \quad (3)$$

Then, the context vector c_l is computed by using the hidden states average input weights with the attention vector. The attentional hidden vector is computed using Eq. (4) by the concatenation of h_o and c_t:

$$h_o' = \text{tanh}\left(W_c\left[c_t,\ h_o\right]\right). \quad (4)$$

Finally, the softmax layer is included to the vector h_o' using Eq.(5) to obtain the predicted target sequence:

$$P\left(t_j \mid t_{<1}, U\right) = \text{softmax}\left(W_s\ h_o'\right). \quad (5)$$

The disadvantages of RNN-based NMT in terms of parallelization and long-term dependencies are tackled by introducing transformer-based NMT [42]. The primary idea behind the transformer model is to make use of the self-attention mechanism, an attention mechanism found inside the encoder.

Each token position is encoded by the transformer model, and self-attention is employed to connect two different tokens that aid in parallelization to quicken learning. The self-attention, also known as multi-head attention, computes attention several times.

The encoder-decoder architecture of transformer-based NMT contains six identical attention layers that are placed on stack of each other. The position of the input sequence is encoded and embedded to combine the sequence of tokens prior to feeding the sequence into the network.

The encoder consists of a point-wise connected feed-forward network layer and multiple headed attention layer. Whereas, the decoder comprises three layers and two of these layers are identical to the encoder.

The another multi-head attention layer is the third layer of the decoder that focuses to attend the output sequence a headed by the encoder. Here, the attention is calculated by considering the dot product of the input and utilizing a softmax function to get the weight of each token at a given position using Eq. (6):

$$\text{Attn}(Q,\ K,\ V) = \text{softmax}\left(\frac{QT^k}{\sqrt{d_k}}\right)V. \quad (6)$$

To compute the attention, input vectors such as query (Q), key (K) with dimension d_k, and value (V) are used. The advantage of using multi-head (MHD) attention in the transformer model over single-head attention is that it allows you to deal with different word representations through multiple positions. As shown in Eq. (7) and (8), the number of parallel attention heads accounts for $h = 8$:

$$\text{MHD}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)\ W^O, \quad (7)$$

$$\text{head}\ _i = \text{Attn}\left(QW_i^Q, KW_i^K, VW_i^v\right), \quad (8)$$

where the parameter matrices $W_i^Q \in R^{d_{\text{model}} \times d_k}$, $W_i^K \in R^{d_{\text{model}} \times d_k}$, and $W_i^V \in R^{d_{\text{model}} \times d_r}$.

2.2 Related Work on English–Assamese MT

In literature review of the MT for English-Assamese pair, it is noted that the researchers are working on the dataset preparation to overcome the dataset's scarcity for such a low-resource pair [4, 14, 23, 37]. The authors of [4] build a phrase-based SMT translation system via preparation of a small En-As parallel corpus of 14,371 sentences.

Table 1. Data statistics for domain wise parallel sentences

Domain	Parallel Sentences
Agriculture	2,150
COVID-19	5,500
Government Office	9,500
Judiciary	4,500
Social Media	3,220
Sports	8,600
Tourism	4,750
Literature	19,300
Total	**57,520**

Table 2. Train, validation and test data statistics

Type	Sent	Tokens	
		En	As
Train Set-1 [23]	203,315	2,414,172	1,986,270
Train Set-2 [37]	138,353	1,715,435	1,377,336
Train Set-3	46,016	560,972	446,500
Total	**387,684**	**4,690,579**	**3,810,106**
Validation Set-1 [23]	4,500	74,561	59,677
Validation Set-2 [37]	1,000	19,922	16,824
Validation Set-3	5,752	75,652	65,612
Total	**11,252**	**170,135**	**142,113**
Test Set-1 [23]	2,500	41,985	34,643
Test Set-2	5,752	75,348	65,576

Table 3. En/As Monolingual data statistics

Type	Sentences	Tokens
As	2,810,197	47,740,981
En	3,387,704	58,847,760

In our previous work [23], a parallel corpus, namely, EnAsCorp1.0 [23] is developed, and it contains 210,315 parallel sentences. And, the same has been used to build baseline models for En-As pair translation using the phrase-based SMT and RNN-based NMT.

Then in the previous work [24], we have explored different NMT models (RNN and transformer) with data augmentation approach and attains better results on the same test set [23] for En-As pair translation.

Moreover, a parallel corpora, namely, Samanantar [37] that contains 11 Indian languages with English, and it includes 141,353 English-Assamese parallel sentences. Also, they [37] implemented transformer-based NMT model for the En-to-Indic and Indic-to-En.

It is noted that all the prior works that have been conducted on this English-Assamese MT are not domain specific. In this work, we have prepared domain specific English-Assamese parallel corpus and utilized parallel corpus of EnAsCorp1.0 and Samanantar to enhance the translational performance for both forward and backward directions of translation. We have addressed data scarcity and word order divergence issues via data augmentation and guided alignment concept.

3 Domain Specific Parallel Corpus Preparation and Dataset Description

In this section, we briefly discuss dataset preparation. First, we have collected Assamese monolingual data from the available online sources. For agriculture and social media domains, we have collected from Assamese monolingual sentences[1] from [34].

The Assamese monolingual sentences of sports[2], literature[3] domains are extracted from the News and Xahityo online sources. For extraction, we used the technique of web scraping, which is an automatic method to obtain large amounts of data from websites.

We employed Scrapy[4] for this purpose. Scrapy is a free and open-source web-crawling framework written in Python. While scrapping, we faced several challenges, which were mainly because of different web page structure in different websites and dynamic web content. Then, Assamese monolingual sentences are translated into English sentences using Bing translator[5].

[1] https://github.com/anononymus/assamese-redup
[2] https://www.asomiyapratidin.in/
[3] https://xahitya.org/
[4] https://github.com/scrapy/scrapy
[5] https://www.bing.com/translator

Table 4. Baseline system results on test set-1 in terms of automatic evaluation scores

Translation	Model	BLEU	TER	RIBES	METEOR	F-measure
En-to-As	PBSMT (Baseline-1)	4.85	103.2	0.2598	0.0768	0.1745
	RNN (Baseline-2)	6.78	93.4	0.2847	0.0996	0.2074
	Transformer (Baseline-3)	6.92	93.1	0.2878	0.1043	0.2106
As-to-En	PBSMT (Baseline-1)	8.58	90.5	0.2938	0.1070	0.2095
	RNN (Baseline-2)	12.52	88.6	0.4262	0.1421	0.2871
	Transformer (Baseline-3)	12.84	88.1	0.4284	0.1477	0.2876

Table 5. Baseline system results on test set-2 in terms of automatic evaluation scores

Translation	Model	BLEU	TER	RIBES	METEOR	F-measure
En-to-As	PBSMT (Baseline-1)	3.62	105.6	0.1676	0.0472	0.1356
	RNN (Baseline-2)	4.26	98.3	0.1706	0.0647	0.1994
	Transformer (Baseline-3)	4.66	98.2	0.1732	0.0686	0.2006
As-to-En	PBSMT (Baseline-1)	4.02	100.8	0.1710	0.0526	0.1487
	RNN (Baseline-2)	6.28	96.6	0.2064	0.1062	0.2008
	Transformer (Baseline-3)	6.49	96.5	0.2098	0.1084	0.2096

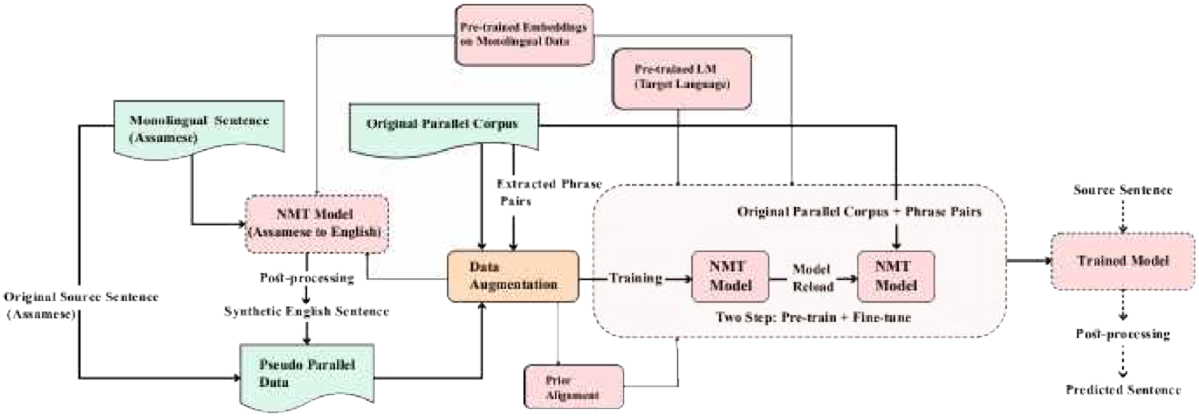

Pre-trained Embeddings on Monolingual Data
Pre-trained LM (Target Language)
Monolingual Sentence (Assamese)
Original Parallel Corpus
Extracted Phrase Pairs
NMT Model (Assamese to English)
Original Parallel Corpus + Phrase Pairs
Source Sentence
Data Augmentation
Training
NMT Model
Model Reload
NMT Model
Trained Model
Post-processing
Original Source Sentence (Assamese)
Synthetic English Sentence
Two Step: Pre-train + Fine-tune
Post-processing
Pseudo Parallel Data
Prior Alignment
Predicted Sentence

Fig. 1. Proposed approach for English-Assamese NMT

Similarly, English side sentences are extracted from News[6] via scraping for the domain of COVID-19 and tourism domains. And, we have collected English sentences of Government office and judiciary domains from IIT Bombay English-Hindi parallel corpus[7]. Then, utilize Bing translator to generate corresponding Assamese sentences. We have considered maximum sentence length 50 words.

Further, we have manually corrected and verified the parallel sentences. For manual verification, we have hired three linguistic experts who possess linguistic knowledge of both English and Assamese, and it took about 70 days.

The statistics of domain-wise parallel sentences are summarized in Table 1. The domain-wise parallel data is split into train, validation, and test data by considering 90%, 10%, 10% from each domain (Agriculture / COVID-19 / Government Office / Judiciary / Social media / Sports / Tourism / Literature) for train, validation and test set.

[6] https://theprint.in/

[7] https://www.cfilt.iitb.ac.in/iitb_parallel/

Table 6. Train data statistics before and after phrase pairs augmentation, OPC:“original parallel corpus”, PP: “phrase pairs”

Type	Sentences
OPC	387,684
PP	10,103,84
OPC + PP	13,980,68

Table 7. BLEU scores results of transformer-based NMT on test set-1, M1 (Without domain-specific parallel data (Train Set-3)): Train Set-1+Train Set-2; M2 (baseline-3): With domain-specific parallel data (Train Set-3) + Train Set-1+Train Set-2; M3: M1+PSP (Post-processing); M4: M2+PSP (Post-processing)

Translation	M1	M2	M3	M4
En-to-As	6.74	6.92	6.87	7.04
As-to-En	12.54	12.84	12.78	12.96

Table 8. BLEU scores results of transformer-based NMT on test set-2, M1 (Without domain-specific parallel data (Train Set-3)): Train Set-1+Train Set-2; M2 (baseline-3): With domain-specific parallel data (Train Set-3) + Train Set-1+Train Set-2; M3: M1+PSP (Post-processing); M4: M2+PSP (Post-processing)

Translation	M1	M2	M3	M4
En-to-As	1.16	4.66	1.21	4.84
As-to-En	2.24	6.49	2.32	6.68

We have named these sets: train set-3, validation-3 and test set-2. The data set statistics, that are used in this work, are summarized in Table 2.

In Table 2, we have merged parallel corpora, namely, EnAsCorp1.0 [23] and Samanantar [37].

Furthermore, we have used monolingual data of Assamese/English from [23] and Assamese/English side monolingual sentences from train set-3.

The data statistics of monolingual data are presented in Table 3. It is mainly used for the preparation of pretrained word embeddings and LM.

4 Baseline System

In our previous work, we have prepared EnAsCorp1.0 [23], wherein, parallel En-As corpus and monolingual sentences of As are collected. The same dataset was used to implement baseline systems by considering two models, namely, phrase-based SMT (baseline-1) and RNN-based NMT (baseline-2).

In this work, we have considered domain-wise En-As parallel corpus (as mentioned in Section 3) in addition to EnAsCorp1.0 [23] and Samanantar [37], data statistics are shown in Table 2. Moreover, custom pretrained word embeddings using GloVe [35] is utilized in NMT models.

For baseline systems, transformer-based NMT [42] (baseline-3) is also considered in addition to RNN-based NMT (baseline-2) and phrase-based SMT (baseline-1).

The reason behind choosing transformer-based NMT in baseline systems is that it outperforms RNN-based NMT and PBSMT (as reported in Table 4, 5) and performs fair comparisons with improved transformer-based NMT (as discussed in Section 5).

To evaluate quantitative results, standard evaluation metrics [32], namely, BLEU (bilingual evaluation under study), TER (translation error rate) [41], RIBES (rank-based intuitive bilingual evaluation score) [11], METEOR (metric for evaluation of translation with explicit ordering) [25], and F-measure scores are considered.

5 Enhanced English-Assamese NMT

In the previous section, we have reported baseline system results, and it is noticed that transformer-based NMT achieves best results for both directions of translation. Therefore, we have chosen transformer-based NMT for further investigation.

In this section, we have briefly described the improved transformer-based NMT for low-resource En-As pair by investigating different approaches like data augmentation, prior alignment, pretrained LM and post-processing step. Figure 1 depicts the proposed approach for En-As NMT.

Table 9. BLEU scores results of transformer-based NMT on test set-1, M5:M2+PP (Phrase pairs); M6: M5+PSP (Post-processing)

Translation	M2	M5	M6
En-to-As	6.92	8.46	9.12
As-to-En	12.84	14.34	15.06

Table 10. BLEU scores results of transformer-based NMT on test set-2, M5:M2+PP (Phrase pairs); M6: M5+PSP (Post-processing)

Translation	M2	M5	M6
En-to-As	4.66	7.86	8.17
As-to-En	6.49	10.34	10.84

Table 11. BLEU scores results of transformer-based NMT on test set-1, M7: M5+SP (synthetic parallel data (pretrain + fine-tune)); M8: M7+PSP (Post-processing)

Translation	M2	M5	M7	M8
En-to-As	6.92	8.46	9.52	10.12
As-to-En	12.84	14.34	15.66	16.04

Table 12. BLEU scores results of transformer-based NMT on test set-2, M7: M5+SP (synthetic parallel data (pretrain + fine-tune)); M8: M7+PSP (Post-processing)

Translation	M2	M5	M7	M8
En-to-As	4.66	7.86	8.64	9.06
As-to-En	6.49	10.34	11.74	12.10

5.1 Data Augmentation

We have tackled data scarcity problem via data augmentation in two-ways: augmenting phrase-pairs and utilizing synthetic parallel data without modifying the NMT model architecture.

Following the strategy [38], phrase-based SMT is trained on original parallel data using Moses[8] toolkit and extracted phrase pairs from the generated phrase table.

However, in our previous work [24], it is noticed that the extracted phrase pairs contain wrong alignment phrases [20].

Therefore, we have extracted phrase pairs by considering different translation probabilities (Set$_{p\geq0.5}$ / Set$_{p=1}$ / Set$_{\text{all}}$) of target phrases given source phrases following [38, 24] and observed that the translation accuracy with augmentation of extracted phrase pairs having translation probability Set $_{p\geq0.5}$ are higher.

Therefore, we have considered phrase pairs with translation probability Set$_{p\geq0.5}$ and the data statistics are reported in Table 6.

Further, to expand the parallel corpus, monolingual data is used to generate synthetic parallel data following BT strategy [39, 24]. However, it is observed that the translational accuracy with augmented data is lower than the without augmented one.

Therefore, following our previous work [24] a two-step solution is used [1]. First, pretrain the NMT model with synthetic data and “original parallel corpus + phrase pairs” and then fine-tune or reload it on the “original parallel corpus + phrase pairs”.

As a result of this , the final model initializes the parameters from the pretrained model that gains the training performance when the “original parallel corpus + phrase pairs” is utilized. We have used As-to-En transformer-based NMT model to generate synthetic parallel data using Assamese monolingual sentences since it gives higher translation accuracy, as shown in Table 4, 5.

To examine the effect of augmented synthetic parallel data, we have performed a series of experiments like our previous work [24] on the ratio of parallel and synthetic corpora. It is noticed that 1:3 + phrase pairs attain higher translation accuracy for As-to-En and similar observation is found in case of En-to-As with 1:4 + phrase pairs and therefore, we have reported these results in Section 6.

5.2 Prior Alignment and Pretrained LM

The word order or token position of English is different from Assamese [24] that leads to word-order divergence issue. In this work, we have attempted to extract token alignment information from the En-As bi-text data and feeded into NMT to enhance En-to-As and As-to-En directions of

[8] http://www.statmt.org/moses/

Table 13. BLEU scores results of transformer-based NMT on test set-1, M11:M7 with PA1 (BA); M12: M7 with PA1 (UA); M13: M7 with PA1 (RA); M14: M7 with PA2 (BA); M15: M7 with PA2 (UA); M16: M7 with PA2 (RA), where PA1:Prior Alignment (FastAlign), PA2:Prior Alignment (SimAlign), UA: Unidirectional Alignment, BA: Bidirectional Alignment (grow-diagonal heuristics), RA: Reverse Direction Alignment

Translation	Model	BLEU
En-to-As	M7	9.52
	M11	10.12
	M12	10.46
	M13	13.12
	M14	11.24
	M15	12.43
	M16	14.54
As-to-En	M7	15.66
	M11	16.42
	M12	17.32
	M14	17.44
	M15	18.32

Table 14. BLEU scores results of transformer-based NMT on test set-2, M11:M5 with PA1 (BA); M12: M5 with PA1 (UA); M13: M5 with PA1 (RA); M14: M5 with PA2 (BA); M15: M5 with PA2 (UA); M16: M5 with PA2 (RA), where PA1:Prior Alignment (FastAlign), PA2:Prior Alignment (SimAlign), UA: Unidirectional Alignment, BA: Bidirectional Alignment (grow-diagonal heuristics), RA: Reverse Direction Alignment

Translation	Model	BLEU
En-to-As	M7	8.64
	M11	8.86
	M12	8.94
	M13	9.08
	M14	8.98
	M15	9.04
	M16	9.16
As-to-En	M7	11.74
	M11	11.82
	M12	11.96
	M14	12.10
	M15	12.28

translation. In [30], FastAlign tool is used to extract the token alignment information from the parallel data and adopted the guided alignment concept in the transformer-based NMT [10].

In [30, 10], the optimization criteria for training the baseline transformer model [42] is presented in Eq. 10, where T denotes the number of target tokens, p represents the output probability distribution, and $r_{i,j}$ indicates $j-th$ the token in the dictionary is the true value at the $i-th$ position in the target sentence.

The modified optimization criteria is represented in Eq. 11, where a pair of source-target sentences of length K and T, respectively, and a prior alignment set:

$$A \subseteq (j-i)) : j = 1, \text{ ...}, \ k, \ \ i = 1, \text{ ...}, \ T. \tag{9}$$

It takes randomly the output of just a head from the fifth decoder layer and project it into T target token probability distribution over K corresponding source token.

It compares the probability distributions q_{ij} with the reference probability generated from prior alignments through cross-entropy. The symbol $a_{i,j}$ represents the $i-th$ target token is properly aligned with the $j-th$ source token.

Both $L1$ and $L2$ are combined in Eq. 12 [10, 30] which is the sum of cross-entropy for tokens and alignment weights of source-target sentences:

$$L_1 = -\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{m}(r_{i,j} \times \log(p_{i,j})), \tag{10}$$

$$L_2 = -\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{K}(a_{i,j} \times \log(q_{i,j})), \tag{11}$$

$$L = L_1 + \lambda L_2, \tag{12}$$

where, λ is a weighted cross-entropy for alignments (a hyperparameter), the authors [30] considered 0.05. For comparative analysis, we also considered FastAlign to extract alignment information and, however, we have considered weighted alignment $\lambda = 0.03$ since it yields the lowest training cost.

In this work, we have proposed to use SimAlign[9] [12] tool to extract the token alignment information. The SimAlign is a word alignment tool that uses static and pretrained multilingual language model (mBERT) based contextualized embeddings.

[9] https://github.com/cisnlp/simalign

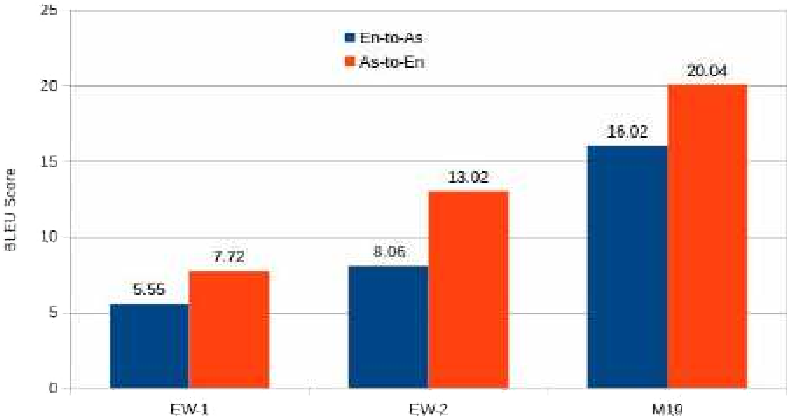


Fig. 2. BLEU score comparison among existing works, EW-1 [23], EW-2 [24] and M19:our best model for En-As pair translation on test set-1

It uses sub-word (BPE) level processing in three various methods: Argmax, Itermax and Match to obtain the alignment information. The basic difference among these three methods is Argmax finds a local optimum and Itermax uses greedy algorithm, whereas Match finds a global optimum via maximum-weight maximal matching technique [12].

Although, we have extracted alignment information using these three methods, reported only Match-based SimAlign, since it shows higher translational performance in the NMT. We have used the two-step process to construct the alignments. First, extract the alignment information from both the forward and backward direction, i.e., En-to-As and As-to-En.

Then, combine the bidirectional alignments using the grow-diagonal heuristics of [17]. For the comparative analysis, we also considered extracted unidirectional (En-to-As or As-to-En) alignment information (as reported in Section 6.2).

It is noticed that the backward direction, i.e., As-to-En translation attains a higher score than that of En-to-As translation. Therefore, we have proposed to use the backward/reverse direction (As-to-En) of alignment information in the forward direction (En-to-As) translation using a simple two-step solution.

First, we reverse the extracted alignment information of the backward direction and then sort them to obtain the alignment information of forward direction.

Table 15. BLEU scores results of transformer-based NMT on test set-1, M17:M16 (En-to-As) / M15 (As-to-En)+ PSP (Post-processing)

Translation	M7	M16/M15	M17
En-to-As	9.52	14.54	15.10
As-to-En	15.66	18.32	19.16

Table 16. BLEU scores results of transformer-based NMT on test set-2, M17:M16 (En-to-As) / M15 (As-to-En)+ PSP (Post-processing)

Translation	M7	M16/M15	M17
En-to-As	8.64	9.16	9.65
As-to-En	11.74	12.28	13.18

The Marian [13] toolkit is employed to uses the source-target prior alignment information in the training process of transformer-based NMT.

Moreover, the pretrained language model (LM) [5] could be used to improve low-resource NMT. We have used the Marian[10] toolkit that allows to use the pretrained language model (LM) in the training process of NMT.

We have utilized the monolingual data of the target language to train and generate an LM using the transformer model, and the weight matrices are loaded from the pretrained LM by initializing the decoder of an encoder-decoder architecture of transformer-based NMT. We have named AsLM for the custom pretrained Assamese LM.

[10] https://marian-nmt.github.io/docs/

Table 17. BLEU scores results of transformer-based NMT on test set-1, M18:M16 (En-to-As) / M15 (As-to-En)+PLM (Pretrained LM); M19: M18+ PSP (Post-processing)

Translation	M16/M15	M18	M19
En-to-As	14.54	15.46	16.02
As-to-En	18.32	19.62	20.04

Table 18. BLEU scores results of transformer-based NMT on test set-2, M18:M16 (En-to-As) / M15 (As-to-En)+PLM (Pretrained LM); M19: M18+ PSP (Post-processing)

Translation	M16/M15	M18	M19
En-to-As	9.16	9.42	10.52
As-to-En	12.28	12.56	13.93

5.3 Post-processing

The post-processing step is used to handle out-of-vocabulary issue. It arises due to the named-entities, compounds, technical terms and misspelled words [2]. The OOV is of two types: Completely Out-of-Vocabulary (COOV) and Sense Out-of-Vocabulary (SOOV).

If the words are not present in the training data, then it is known as COOV, on the other hand SOOV are those words which are present in the training data with different usage or sense from the test set words. NMT generates <unk> (unknown) tokens against OOV.

Furthermore, NMT shows weakness in case of rare word translation since fixed-size vocabulary, which forces producing <unk> [27]. The authors [40] introduced byte pair encoding (BPE) to handle the OOV issue. Likewise, we have used BPE and proposed to use a post-processing step.

The post-processing step contains two key components: Bilingual Dictionary and Transliteration Module **Bilingual Dictionary:** We have prepared a bilingual English - Assamese dictionary since there is lack of available dictionary data for En-As pair.

In our previous work [22], we have collected 200,151 a number of En-As parallel sentences from an online dictionary, namely, Glosbe.

Moreover, we have extracted 10, 103, 84 phrase pairs from the train set (as discussed in Section 5.1). We have used both (Glosbe and phrase pairs) to filter out single and double parallel words.

In the prepared dictionary, the total number of parallel single/double words are 464,586, wherein 87,024 from Glosbe and rest are from phrase pairs. We have filtered parallel noun phrases from the phrase pairs using two steps: first, extracted noun phrases from the English side of phrase pairs using NLTK[11] tool and then mapped those sentences in the phrase pairs to collect corresponding Assamese noun phrases.

The bilingual dictionary is used to replace the <unk> tokens with the appropriate target words concerning source words. **Transliteration Module:** We have used this module to source words which are not present in the bilingual dictionary.

It is mainly used to handle the unseen tokens that produce <unk>. We have used indic-trans[12] [6] to convert the source word into the target word script in the predicted sentence for both En-to-As and As-to-En transliteration.

6 Experiment and Result

In this section, we briefly present experimental setup and reported quantitative results with error analysis.

6.1 Experimental Setup

We have employed two setups in the baseline system experiments, namely, phrase-based SMT (PBSMT) and NMT. For PBSMT, the Moses[13] [18] toolkit is used, wherein, GIZA++ [31] and IRSTLM [9] are used to extract phrase pairs to build the translational model and language model, following default settings of Moses.

The NMT experiments are carried out using the publicly available Marian [13] toolkit in three basic operations, data preprocessing, training and testing.

[11] https://www.nltk.org/

[12] https://github.com/libindic/indic-trans

[13] http://www.statmt.org/moses/

Table 19. BLEU scores on different sentence group distribution for Test Set-1, SG: Sentence Group, NM-1: M7, NM-2: M16 (En-to-As) / M15 (As-to-En), NM-3: NM-2 + PLM

SG	Length	No. of Sentences	NM-1	NM-2	NM-3
1	1-15	1344	En-to-As: 15.98	En-to-As: 17.72	En-to-As: 17.96
			As-to-En: 18.98	As-to-En: 23.32	As-to-En: 23.96
2	16-30	944	En-to-As: 10.52	En-to-As: 11.22	En-to-As: 11.36
			As-to-En: 12.16	As-to-En: 17.38	As-to-En: 17.87
3	31-45	179	En-to-As: 9.32	En-to-As: 10.52	En-to-As: 11.69
			As-to-En: 11.47	As-to-En: 15.26	As-to-En: 15.57
4	46-80	33	En-to-As: 4.40	En-to-As: 7.48	En-to-As: 9.29
			As-to-En: 7.34	As-to-En: 9.54	As-to-En: 11.38

Table 20. Comparative quantitative results on test set-1 in terms of automatic evaluation scores

Translation	Model	BLEU	TER	RIBES	METEOR	F-measure
En-to-As	M2 (baseline-3)	6.92	93.1	0.2878	0.1043	0.2106
	M19 (best)	16.02	79.4	0.4226	0.2712	0.6346
As-to-En	M2 (baseline-3)	12.84	88.1	0.4284	0.1477	0.2876
	M19 (best)	20.04	74.5	0.4738	0.3846	0.7584

Table 21. Comparative quantitative results on test set-2 in terms of automatic evaluation scores

Translation	Model	BLEU	TER	RIBES	METEOR	F-measure
En-to-As	M2 (baseline-3)	7.66	91.3	0.3032	0.1286	0.2306
	M19 (best)	10.52	88.4	0.4027	0.1406	0.2798
As-to-En	M2 (baseline-3)	10.49	89.5	0.4098	0.1384	0.2796
	M19 (best)	13.93	82.4	0.3826	0.2394	0.2847

In the data preprocessing step, the word-segmentation technique, namely, byte pair encoding (BPE) [40] with $32k$ merge operations is utilized. The vocabulary size of English and Assamese are $32,404$ and, $31,920$ at sub-word level (BPE).

Moreover, we have used GloVe [35] word embeddings as subword level, wherein, the pretraining is performed up to 100 iterations with embedding vector size 200. We have named AsGloVe for custom Assamese word embeddings on Assamese side monolingual data.

For RNN-based NMT, we have investigated RNN and bidirectional RNN in our previous work [23, 24] and it is observed that the bidirectional RNN-based NMT shows better translational accuracy.

Thus, we have considered bidirectional RNN-based NMT in baseline-2, where, 0.3 drop-out in two-layer LSTM-based encoder-decoder architecture is used [26].

The default configuration of six layers with eight attention heads, drop-out of 0.1, and Adam optimizer with a learning rate of 0.001 are used in the training process of NMT and LM.

A single NVIDIA Quadro P2000 GPU is utilized to train the models with early stopping criteria, i.e., the model training is halted if it does not converge on the validation set for more than 10 epochs.

6.2 Result and Error Analysis

We have used automatic evaluation metrics, namely, BLEU, TER, RIBES, METEOR,

Table 22. Human evaluation scores on test set-1, AD: Adequacy, FL: Fluency, OR: Overall Ratings

Translation	Model	AD	FL	OR
En-to-As	M2 (baseline-3)	2.56	3.26	2.91
	M19 (best)	4.92	5.84	5.38
As-to-En	M2 (baseline-3)	2.96	3.76	3.36
	M19 (best)	5.12	6.26	5.69

Table 23. Human evaluation scores on test set-2, AD: Adequacy, FL: Fluency, OR: Overall Ratings

Translation	Model	AD	FL	OR
En-to-As	M2 (baseline-3)	1.36	2.02	1.69
	M19 (best)	2.04	3.12	2.58
As-to-En	M2 (baseline-3)	1.84	2.38	2.11
	M19 (best)	2.12	3.18	2.65

F-measure, and human evaluation scores to evaluate the quantitative results of predicted translations.

Table 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, reports the comparative BLEU score results of exploring the transformer-based NMT in different configurations i.e, with or without domain-specific parallel data, phrase pairs augmentation, synthetic parallel data, prior alignment, pretrained LM and along with the post-processing step on test set-1 and test set-2.

Furthermore, we have reported statistical significance in Table 19, wherein, BLEU scores are evaluated on the test set-1 in four groups of sentence length. It is noticed that translation accuracy decreases as the increase in sentence length (number of words).

The effect of LM is realized in the sentences of group 4 (length: 46 - 80). Table 20 and 21 presents comparative results in terms of different automatic evaluation metrics of our best model (enhanced transformer-based NMT) over the baseline transformer model (baseline-3).

Figure 2 presents comparative results of our best model over the existing works [23, 24] in terms of BLEU scores. All facets of translation accuracy cannot be evaluated using the automatic evaluation measures. Thus, the human evaluation (HE) or manual evaluation metric is taken into account. It consists of two aspects: adequacy and fluency.

The adequacy factor measures how well the predicted translation, which corresponds to the reference sentence, is contextually represented. Whereas, fluency is a different criterion that determines whether the predicted translation is well-formed or not.

The overall rating[14] of HE is calculated by the average score of adequacy and fluency. For example, if a reference sentence is: *“He is coming to the park”* and the predicted sentence is: *“He is a good boy.”* Here, the predicted sentence, inadequate with respect to the reference sentence.

But, the predicted sentence is fluent since it is a well-formed or grammatical well-structured sentence. We have hired three human evaluators who possess linguistic knowledge of both the languages, i.e., English and Assamese, and considered the assessment criteria on a scale of 1-5 on randomly selected 100 sample sentences following [33]. Table 22, 23 report the manual evaluation results of transformer model (baseline) and the best model, wherein, the average scores of three human evaluators are presented.

From the quantitative results, it is observed that our best model (M19) attains higher translation accuracy than the baseline models.

[14] https://nlp.amrita.edu/mtil_cen/\#results

Also, it is observed that As-to-En translation attains higher translational performance than En-to-As.

It is because of the presence of more number of tokens in En side as compared to As (as mentioned in Table 2) and as a result, more number of En tokens are encoded by the encoder and the decoder can produce a better translation for As-to-En translation.

From Table 8, it is observed the NMT performance lowers in M1 (without domain-specific parallel train set) [19], therefore, by contributing domain-specific parallel corpus in this work, NMT translational performance improves for both directions of translation covering various domains.

To closely analyse the effect of domain-specific parallel data, the sample predicted sentences of best model and with Google[15] and Bing[16] translation are discussed using the following notations:

— SS: Source sentence.

— TT: Reference / Target sentence.

— PT1: Predicted sentence by the best model (En-to-As).

— PT2: Predicted sentence by the best model (As-to-En).

— BT: Bing translation.

— GT: Google translation.

1. (a) Example-1 (Agriculture): En-to-As

SS: *More than 50 percent of these bamboos are found across the North East including Assam.*

TT: ইয়াৰে ৫০ শতাংশৰো অধিক বাঁহ অসমকে ধৰি সমগ্ৰ উত্তৰ পূত্ৰ্বাঞ্চলত পোৱা যায় । (*yare 50 shatangshu adhik banh asomake dhari samagra uttar purtwanchalat poua jai*)

PT1: প্ৰায় ৫০টাতকৈ অধিক বাঁহবিলাক ধৰি উত্তৰপূৰ্বাঞ্চলত পোৱা যায় । (*praiy 50tatki adhik banhhbilak dhari uttarapurbanchalat poua jai*)

BT: ইয়াৰে ৫০ শতাংশতকৈও অধিক বাঁহ অসমকে ধৰি উত্তৰ-পূৰ্বাঞ্চলত পোৱা যায় । (*yare 50 shatanshtkio adhik banh asamake dhari uttar-purbanchalat poua jai*)

GT: অসমকে ধৰি সমগ্ৰ উত্তৰ পূৰ্বাঞ্চলত ৫০ শতাংশতকৈ অধিক বাঁহ পোৱা যায় (*asamake dhari samagr uttar purbanchalat 50 shatanshtaki adhik banh poua jai*)

1. (b) Example-1 (Agriculture): As-to-En

SS ইয়াৰে ৫০ শতাংশৰো অধিক বাঁহ অসমকে ধৰি সমগ্ৰ উত্তৰ পূত্ৰ্বাঞ্চলত পোৱা যায় । (*yare 50 shatangshu adhik banh asomake dhari samagra uttar purtwanchalat poua jai*)

TT: *More than 50 percent of these bamboos are found across the North East including Assam.*

PT2: *More than 50 per cent bamboo available in the state of North East India.*

BT: *More than 50 per cent of these bamboos are found in the entire north-eastern region including Assam.*

GT: *More than 50 per cent of this bamboo is found in the entire North East including Assam.*

Discussion: In the above examples, both directions of predicted translation of PT1 and PT2 are fluent like BT and GT. However, predicted translations are partial adequate unlike BT and GT, since PT2 misses "including Assamese", and plural form of "bamboo". Whereas, PT1 misses "অসমকে"

2. (a) Example-2 (Social Media): En-to-As

SS: *The moon hangs in the sky like a huge plate.*

TT: আকাশত প্ৰকাণ্ড থালিখনৰ দৰেই জোনবাইজনী ওলমি আছে। (*akashat prakando thalikhanar darei jonvaijani ulomi aache.*)

PT1: আকাশত ওলমি থকা চন্দ্ৰ। (*akashat ulomi thaka chandra*)

BT: চন্দ্ৰটো এটা ডাঙৰ প্লেটৰ দৰে আকাশত ওলমি আছে। (*chandrato eta dangor plater dore aakasot ulomi ase*)

GT: বিশাল প্লেটৰ দৰে আকাশত ওলমি আছে চন্দ্ৰ। (*vishal plator dore akashat ulomi ase chandra*)

[15] https://translate.google.co.in/ (Accessed on 30.09.2022)
[16] https://www.bing.com/translator(Accessed on 30.09.2022)

2. (b) Example-2 (Social Media): As-to-En

SS: আকাশত প্ৰকাণ্ড থালিখনৰ দৰেই জোনবাইজনী ওলমি আছে। (*akashat prakando thalikhanar darei jonvaijani ulomi aache.*)

TT: *The moon hangs in the sky like a huge plate.*

PT2: *Jonbil is hanging in the sky.*

BT: *The zonbaijani is hanging in the sky like a huge thali.*

GT: *The moon hangs in the sky like a huge plate.*

Discussion: Here, both PT1 and PT2 generate partially adequate translation, but sentences are fluent like BT and GT. Also, unlike GT, BT unable to produce correct word (“zonbaijani”, “thali”) for As-to-En translation.

3. (a) Example-3 (Judiciary): En-to-As

SS: *The respondent asserted that after show cause notice dated 15th June 2001 was replied by the petitioner by letter dated 8th July.*

TT: উত্তৰদাতাই দৃঢ়তাৰে কৈছিল যে 15 জুন 2001 তাৰিখৰ কাৰণ দৰ্শোৱাৰ জাননীৰ পিছত আবেদনকাৰীয়ে 8 জুলাই তাৰিখৰ পত্ৰৰ দ্বাৰা ইয়াৰ উত্তৰ দিছিল ।

PT1: উত্তৰদাতাই স্বীকাৰ কৰিছিল যে 8 জুন 2001 তাৰিখৰ কাৰণ দৰ্শোৱাৰ জাননী জাৰী কৰা হৈছিল ।

BT: উত্তৰদাতাই দৃঢ়তাৰে কৈছিল যে 15 জুন 2001 তাৰিখৰ কাৰণ দৰ্শোৱাৰ জাননীৰ পিছত আবেদনকাৰীয়ে 8 জুলাই তাৰিখৰ পত্ৰৰ দ্বাৰা উত্তৰ দিছিল ।

GT: প্ৰতিবাদীয়ে দৃঢ়তাৰে কয় যে ২০০১ চনৰ ১৫ জুন তাৰিখৰ কাৰণ দেখুৱাৰ পিছত আবেদনকাৰীয়ে ৮ জুলাই তাৰিখৰ পত্ৰযোগে উত্তৰ দিছিল ।

3. (b) Example-3 (Judiciary): As-to-En

SS: উত্তৰদাতাই দৃঢ়তাৰে কৈছিল যে 15 জুন 2001 তাৰিখৰ কাৰণ দৰ্শোৱাৰ জাননীৰ পিছত আবেদনকাৰীয়ে 8 জুলাই তাৰিখৰ পত্ৰৰ দ্বাৰা ইয়াৰ উত্তৰ দিছিল ।

TT: *The respondent asserted that after show cause notice dated 15th June 2001 was replied by the petitioner by letter dated 8th July .*

PT2: *The respondent asserted that after the issuance of the show cause notice dated 15 June 2001 the petitioner submitted its reply by the letter dated 8th July .*

BT: *The respondent asserted that after the show cause notice dated June 15, 2001, the petitioner had replied to it by letter dated July 8 .*

GT: *The respondent asserted that after the show cause notice dated 15 June 2001, the petitioner replied to it by letter dated 8 July .*

Discussion: In the above examples, PT1 and PT2 show weakness in adequacy since both unable to produce correct translation of last sub-phrase “by the petitioner by letter dated 8th July” / পিছত আবেদনকাৰীয়ে 8 জুলাই তাৰিখৰ পত্ৰৰ দ্বাৰা ইয়াৰ উত্তৰ দিছিল unlike BT and GT. However, fluency is fine in all the predicted translations.

4. (a) Example-4 (Government Office): En-to-As

SS: *Official receiver or assignee in insolvency proceedings*

TT: দেউলিয়া প্ৰক্ৰিয়াত অফিচিয়েল ৰিচিভাৰ বা আৱণ্টনকাৰী

PT1: চৰকাৰী অনুসন্ধানৰ কাৰ্যক্ৰমসমূহৰ ৰিচিভাৰ

BT: দেউলিয়া প্ৰক্ৰিয়াত অফিচিয়েল ৰিচিভাৰ বা আৱণ্টনকাৰী

GT: ইনছলভেন্সি প্ৰক্ৰিয়াত অফিচিয়েল ৰিচিভাৰ বা এচাইনী

4. (b) Example-4 (Government Office): As-to-En

SS: দেউলিয়া প্ৰক্ৰিয়াত অফিচিয়েল ৰিচিভাৰ বা আৱণ্টনকাৰী

TT: *Official receiver or assignee in insolvency proceedings*.

PT2: *Official resource in bankruptcy proceeding or the allocation*.

BT: *Official receiver or allottee in insolvency proceedings*.

GT: *The official receiver or allocator in bankruptcy proceedings*.

Discussion: Here, PT1 and PT2 produce inadequate as well as not fluent translations, unlike BT ang GT.

5. (a) Example-5 (Tourism): En-to-As

SS: *Taj Mahal ticket to increase by Rs 200*.

TT: তাজমহলৰ টিকট ২০০ টকা বৃদ্ধি হ'ব

PT1: ২০০ টকা খৰচ বৃদ্ধি কৰাৰ বাবে তাজমহলৰ টিকটৰ

BT: তাজমহলৰ টিকট ২০০ টকা বৃদ্ধি হ'ব

GT: ২০০ টকা বৃদ্ধি হ’ব তাজমহলৰ টিকট

5. (b) Example-5 (Tourism): As-to-En

SS: তাজমহলৰ টিকট ২০০ টকা বৃদ্ধি হ'ব

TT: *Taj Mahal ticket to increase by Rs 200*.

PT2: *200 Rs will increase in Taj Mahal*.

BT: *Taj Mahal tickets to be increased by Rs 200*.

GT: *Tickets for the Taj Mahal will be increased by Rs*.

Discussion: Here, PT2 missed the word “ticket”, that leads to inadequate translation Unlike BT. Whereas, GT unable produce “200” in output. However, PT1 produce correct translation like BT and GT in terms of both adequacy and fluency factors of translation.

6. (a) Example-6 (COVID-19): En-to-As

SS: *The fresh order comes amid concerns in the government about the Covid19 lockdown disrupting the supply chain of essential goods.*

TT: কভিড১৯ লকডাউনে অত্যাৱশ্যকীয় সামগ্ৰীৰ যোগান শৃংখলা ব্যাহত কৰাৰ বিষয়ে চৰকাৰত উদ্বেগৰ মাজতে নতুন নিৰ্দেশটো আহিছে ।

PT1: চৰকাৰে কঠোৰ সামগ্ৰীৰ যোগান ব্যাহত কৰাৰ বাবে চৰকাৰৰ কোভিড১৯ লকডাউনৰ বিষয়ে উদ্বেগ প্ৰকাশ কৰে ।

BT: কভিড১৯ লকডাউনে অত্যাৱশ্যকীয় সামগ্ৰীৰ যোগান শৃংখলা ব্যাহত কৰাৰ বিষয়ে চৰকাৰত উদ্বেগৰ মাজতে নতুন নিৰ্দেশটো আহিছে।

GT: Covid19 লকডাউনে অত্যাৱশ্যকীয় সামগ্ৰীৰ যোগান শৃংখলত ব্যাঘাত জন্মাবলৈ চৰকাৰত উদ্বেগ প্ৰকাশ কৰাৰ সময়তে এই সতেজ নিৰ্দেশ ।

6. (b) Example-6 (COVID-19): As-to-En

SS: কভিড১৯ লকডাউনে অত্যাৱশ্যকীয় সামগ্ৰীৰ যোগান শৃংখলা ব্যাহত কৰাৰ বিষয়ে চৰকাৰত উদ্বেগৰ মাজতে নতুন নিৰ্দেশটো আহিছে ।

TT: *The fresh order comes amid concerns in the government about the Covid19 lockdown disrupting the supply chain of essential goods.*

PT2: *The new orders have come when the Covid19 lockdown avoids an essential commotion.*

BT: *The new order comes amid concerns in the government about the Covid-19 lockdown disrupting the supply chain of essential commodities.*

GT: *The new directive comes amid concerns in the government that the lockdown has disrupted the supply chain of essential commodities.*

Discussion: Both PT1 and PT2 yield fluent translation like BT and GT. But partially adequate translation in PT1 and PT2, unlike BT and GT.

7. (a) Example-7 (Sports): En-to-As

SS: *Indian boxers to start practice for Olympics from June 10.*

TT: ভাৰতীয় বক্সাৰসকলে ১০ জুনৰ পৰা অলিম্পিকৰ বাবে আৰম্ভ কৰিব অনুশীলন

PT1: ভাৰতীয় বক্সাৰসকলে ১০ জুনৰ পৰা অলিম্পিকৰ বাবে আৰম্ভ কৰিব

BT: ভাৰতীয় বক্সাৰসকলে ১০ জুনৰ পৰা অলিম্পিকৰ বাবে অনুশীলন আৰম্ভ কৰিব

GT: ১০ জুনৰ পৰা অলিম্পিকৰ বাবে অনুশীলন আৰম্ভ কৰিব ভাৰতীয় বক্সাৰসকলে

7. (b) Example-7 (Sports): As-to-En

SS: ভাৰতীয় বক্সাৰসকলে ১০ জুনৰ পৰা অলিম্পিকৰ বাবে আৰম্ভ কৰিব অনুশীলন

TT: *Indian boxers to start practice for Olympics from June 10.*

PT2: *Indian boxers should start on Olympics from June 10.*

BT: *Indian boxers to start training for Olympics from June 10.*

GT: *Indian boxers will start training for the Olympics from June.*

Discussion: Both PT1 and PT2 missed the word “practice” or অনুশীলন in output, that lead to partially adequate unlike GT and BT. However, all the sentences are fluent.

8. (a) Example-8 (Literature): En-to-As

SS: *A practice called Mizwah has been prevalent among Jewish people.*

TT: ইহুদি ধৰ্মাৱলম্বী লোকসকলৰ মাজত মিজ্ৱাহ নামৰ এটা প্ৰথা প্ৰচলিত হৈ আহিছে ।

PT1: মিজুলুই ইহুদী লোকসকলৰ মাজত মিলুই প্ৰচলিত কৰিছে।

BT: ইহুদী লোকসকলৰ মাজত মিজৱাহ নামৰ এটা প্ৰথা প্ৰচলিত হৈ আহিছে।

GT: ইহুদী লোকসকলৰ মাজত মিজৱা নামৰ এটা প্ৰথা প্ৰচলিত হৈ আহিছে

8. (b) Example-8 (Literature): As-to-En

SS: ইহুদি ধৰ্মাৱলম্বী লোকসকলৰ মাজত মিজ্ৱাহ নামৰ এটা প্ৰথা প্ৰচলিত হৈ আহিছে ।

TT: *A practice called Mizwah has been prevalent among Jewish people.*

PT2: *A practice of worship is prevalent among Jewish people.*

BT: *There has been a custom called Mizwah among the Jewish people.*

GT: *There is a custom called mizvah among the Jewish people.*

Discussion: Like BT and GT, both PT1 and PT2 generate fluent translation. However, inadequate translation in case of PT1 and PT2 unlike BT and GT.

7 Conclusion and Future Work

In this work, we have contributed domain-wise parallel corpus into previous developed dataset, EnAsCorp1.0 [23], we have improved NMT to cover different domains, such as, Agriculture, Social Media, Judiciary, Government Office, COVID-19, Sports, Tourism, Literature for En-As pair translation.

By data augmentation via phrase pairs in addition to the original parallel corpus, more token alignment information is passed into the training model. Also, utilization of synthetic parallel sentences via pretrain and fine-tune steps, we have handled the data scarcity issues for En-As pair translation. It improves translational performance for both directions of translation.

By injecting prior alignment information with pretrained multilingual contextual embeddings-based alignment technique i.e., SimAlign in the transformer-based NMT attains higher translation accuracy than the FastAlign-based prior alignment information or without alignment information.

Moreover, the backward direction, i.e., As-to-En achieves better translational performance than the forward direction En-to-As. Therefore, we have proposed to use reverse order (As-to-En) alignment information in the forward direction (En-to-As) and it shows enhancement in the forward direction of translation i.e., En-to-As.

With custom pretrained LM, translation accuracy is higher in the long-type sentences (as mentioned in Table 19). However, it is inadequate since contextual meaning is different from the source sentence, but fluency is better in the case of the best model for both directions of translation. The domain-wise parallel data will be increased in future work, and attempt to apply the multilingual transfer learning-based approach for further research.

Acknowledgments

The authors are thankful to the Department of Computer Science and Engineering and Center for Natural Language Processing (CNLP), Artificial Intelligence (AI) Lab at the National Institute of Technology, Silchar for providing the requisite support and infrastructure to execute this work.

References

1. **Abdulmumin, I., Galadanci, B. S., Garba, A. (2019).** Tag-less back-translation. DOI: 10.48550/ARXIV.1912.10514.

2. **Aminian, M., Ghoneim, M., Diab, M. (2014).** Handling OOV words in dialectal Arabic to English machine translation. Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants, Association for Computational Linguistics, pp. 99–108.

3. **Bahdanau, D., Cho, K., Bengio, Y. (2015).** Neural machine translation by jointly learning to align and translate. 3rd International Conference on Learning Representations. Conference Track Proceedings, pp. 1–15. DOI: 10.48550/ARXIV.1409.0473.

4. **Barman, A., Sarmah, J., Sarma, S. (2014).** Assamese wordnet based quality enhancement of bilingual machine translation system. Proceedings of the Seventh Global Wordnet Conference, pp. 256–261.

5. **Baziotis, C., Haddow, B., Birch, A. (2020).** Language model prior for low-resource neural machine translation. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, pp. 7622–7634. DOI: 10.48550/ARXIV.2004.14928.

6. **Bhat, I. A., Mujadia, V., Tammewar, A., Bhat, R. A., Shrivastava, M. (2014).** Iiit-h system submission for fire2014 shared task on transliterated search. Proceedings of the Forum for Information Retrieval Evaluation, Association for Computing Machinery, pp. 48—53. DOI: 10.1145/2824864.2824872.

7. **Denkowski, M., Neubig, G. (2017).** Stronger baselines for trustable results in neural machine translation. Proceedings of the First Workshop on Neural Machine Translation, Association for Computational Linguistics, pp. 18–27. DOI: 10.48550/ARXIV.1706.09733.

8. **Dutta, H. (2019).** Assamese Orthography: An Introduction and Some Applications for Literacy Development. Springer International Publishing, pp. 181–194. DOI: 10.1007/978-3-030-05977-4_10.

9. **Federico, M., Bertoldi, N., Cettolo, M. (2008).** IRSTLM: an open source toolkit for handling large scale language models. INTERSPEECH, ISCA, pp. 1618–1621. DOI: 10.21437/interspeech.2008-271.

10. **Garg, S., Peitz, S., Nallasamy, U., Paulik, M. (2019).** Jointly learning to align and translate with transformer models. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, pp. 4452–4461. DOI: 10.18653/v1/D19-1453.

11. **Isozaki, H., Hirao, T., Duh, K., Sudoh, K., Tsukada, H. (2010).** Automatic evaluation of translation quality for distant language pairs. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 944–952.

12. **Jalili Sabet, M., Dufter, P., Yvon, F., Schütze, H. (2020).** SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, Association for Computational Linguistics, pp. 1627–1643. DOI: 10.48550/ARXIV.2004.08728.

13. **Junczys Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., et al. (2018).** Marian: Fast neural machine translation in C++. Proceedings of ACL 2018, System Demonstrations, pp. 116–121. DOI: 10.48550/ARXIV.1804.00344.

14. **Kanchan Baruah, K., Das, P., Hannan, A., Sarma, S. K. (2014).** Assamese-English bilingual machine translation. International Journal on Natural Language Computing (IJNLC).

15. **Khenglawt, V., Laskar, S. R., Pakray, P., Manna, R., Khan, A. K. (2022).** Machine translation for low-resource English-Mizo pair encountering tonal words. Computación y Sistemas, Vol. 26, No. 3.

16. **Kocmi, T. (2020).** Exploring benefits of transfer learning in neural machine translation. DOI: 10.48550/ARXIV.2001.01622.

17. **Koehn, P., Axelrod, A., Birch, A., Callison-Burch, C., Osborne, M., Talbot, D. (2005).** Edinburgh system description for the 2005 IWSLT speech translation evaluation. 2005 International Workshop on Spoken Language Translation, ISCA, pp. 68–75.

18. **Koehn, P., Hoang, H., Birch, A., Burch, C. C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007).** Moses: Open source toolkit for statistical machine translation. ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, The Association for Computational Linguistics, pp. 177—180.

19. **Koehn, P., Knowles, R. (2017).** Six challenges for neural machine translation. Proceedings of the First Workshop on Neural Machine Translation, Association for Computational Linguistics, pp. 28–39. DOI: 10.18653/v1/W17-3204.

20. **Koehn, P., Och, F. J., Marcu, D. (2003).** Statistical phrase-based translation. Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 127–133.

21. **Lalrempuii, C., Soni, B., Pakray, P. (2021).** An improved English-to-Mizo neural machine translation. ACM Transactions on Asian and Low-Resource Language Information Processing, Vol. 20, No. 4, pp. 1–21. DOI: 10.1145/3445974.

22. **Laskar, S. R., Faiz Ur Rahman Khilji Darsh Kaushik, A., Pakray, P., Bandyopadhyay, S. (2021).** EnKhCorp1.0: An English-Khasi corpus. Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021), Association for Machine Translation in the Americas, pp. 89–95.

23. **Laskar, S. R., Khilji, A. F. U. R., Pakray, P., Bandyopadhyay, S. (2020).** EnAsCorp1.0: English-Assamese corpus. Proceedings of the 3rd Workshop on Technologies for MT of Low Resource Languages, Association for Computational Linguistics, pp. 62–68.

24. **Laskar, S. R., Ur Rahman Khilji, A. F., Pakray, P., Bandyopadhyay, S. (2022).** Improved neural machine translation for low-resource English–Assamese pair. Journal of Intelligent and Fuzzy Systems, Vol. 42, No. 5, pp. 4727–4738.

25. **Lavie, A., Denkowski, M. J. (2009).** The meteor metric for automatic evaluation of machine translation. Machine Translation, Vol. 23, No. 3, pp. 105—115. DOI: 10.1007/s10590-009-9059-4.

26. **Luong, T., Pham, H., Manning, C. D. (2015).** Effective approaches to attention-based neural machine translation. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 1412–1421.

27. **Luong, T., Sutskever, I., Le, Q., Vinyals, O., Zaremba, W. (2015).** Addressing the rare word problem in neural machine translation. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Vol. 1, pp. 11–19. DOI: 10.48550/ARXIV.1410.8206.

28. **Mahanta, S. (2012).** Assamese. Journal of the International Phonetic Association, Vol. 42, No. 2, pp. 217–224. DOI: 10.1017/s0025100312000096.

29. **Megerdoomian, K., Parvaz, D. (2008).** Low-density language bootstrapping: the case of tajiki Persian. Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08),

European Language Resources Association (ELRA), pp. 3293–3298.

30. **Nguyen, T., Nguyen, L., Tran, P., Nguyen, H. (2021).** Improving transformer-based neural machine translation with prior alignments. Complexity, Vol. 2021, pp. 1–10. DOI: 10.1155/2021/5515407.

31. **Och, F. J., Ney, H. (2003).** A systematic comparison of various statistical alignment models. Computational Linguistics, Vol. 29, No. 1, pp. 19–51. DOI: 10.1162/089120103321337421.

32. **Papineni, K., Roukos, S., Ward, T., Zhu, W. J. (2002).** BLEU: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 311–318. DOI: 10.3115/1073083.1073135.

33. **Pathak, A., Pakray, P., Bentham, J. (2018).** English–Mizo machine translation using neural and statistical approaches. Neural Computing and Applications, Vol. 30, pp. 1–17.

34. **Pathak, D., Nandi, S., Sarmah, P. (2022).** Reduplication in Assamese: Identification and modeling. Transactions on Asian and Low-Resource Language Information Processing, Vol. 21, No. 5, pp. 1–18. DOI: 10.1145/3510419.

35. **Pennington, J., Socher, R., Manning, C. D. (2014).** Glove: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP. A meeting of SIGDAT, a Special Interest Group of the ACL, Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/d14-1162.

36. **Probst, K., Brown, R., Carbonell, J., Lavie, A., Levin, L. S., Peterson, E. (2001).** Design and implementation of controlled elicitation for machine translation of low-density languages. pp. 3293–3298.

37. **Ramesh, G., Doddapaneni, S., Bheemaraj, A., Jobanputra, M., Sharma, A., Sahoo, S., Diddee, H., Kakwani, D., Kumar, N. (2021).** Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. Transactions of the Association for Computational Linguistics, Vol. 10, pp. 145–162.

38. **Sen, S., Hasanuzzaman, M., Ekbal, A., Bhattacharyya, P., Way, A. (2020).** Neural machine translation of low-resource languages using SMT phrase pair injection. Natural Language Engineering, Vol. 27, No. 3, pp. 271–292. DOI: 10.1017/s1351324920000303.

39. **Sennrich, R., Haddow, B., Birch, A. (2016).** Improving neural machine translation models with monolingual data. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 86–96. DOI: 10.48550/ARXIV.1511.06709.

40. **Sennrich, R., Haddow, B., Birch, A. (2016).** Neural machine translation of rare words with subword units. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1715–1725. DOI: 10.48550/ARXIV.1508.07909.

41. **Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J. (2006).** A study of translation edit rate with targeted human annotation. Proceedings of Association for Machine Translation in the Americas, pp. 223–231.

42. **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017).** Attention is all you need. In Advances in Neural Information Processing Systems. pp. 5998–6008.

Article received on 09/07/2022; accepted on 07/10/2022.
Corresponding author is Partha Pakray.

ISSN 2007-9737

Agent-Based Modeling for Evaluation of Transportation Mode Selection in the State of Guanajuato, Mexico

David Salas-Rodríguez[1], Luis Arturo Rivas-Tovar[2]

[1] Instituto Tepeyac,
Mexico

[2] Instituto Politécnico Nacional,
ESCA STO,
Mexico

{davino66, larivas33}@hotmail.com

Abstract. One of the negative consequences of the industrialization of Mexico favored by the North American Free Trade Agreement (NAFTA), is the emergence of huge industrial corridors associated with the demand for mobility by commuters who move to their workplace. The demand produces mobility patterns that have a serious impact on air pollution in five cities in the state of Guanajuato that, despite being medium in size, outnumber Mexico City in pollution. The objective of this work is to model a data-driven agent based on the beliefs-desires-intentions model, to predict the selection of transport modes using a J48 decision tree algorithm that was designed from data from the 2015 national census (INEGI). The method is mode based l agent programmed in Net logo. The results show that: it is possible to predict the demand of transport considering the: gender, level of education, transfer times and age in the five cities of Guanajuato, in a horizon of three years. With changes in public policies related to mobility and changes in transportation patterns, air pollution would be reduced. The proposed model could be used to support public policies that improve mobility and positively impact air quality in five cities in the state of Guanajuato.

Keywords. Data-driven, agent-simulation, J48, kappa-index, MCCI, Air pollution, Guanajuato Mexico-

1 Introduction

The North American Free Trade Agreement (NAFTA) started on January 1, 1994, as a consequence, Mexican exports grew steadily. In 22 years, the Mexican economy has been transformed: 82% of its foreign sales are manufactured products. in 1993 oil exports represented 10% of the Gross Domestic Product (GDP) but 40% of government incomes, in 2018 was only 6% and 8 % government incomes At present, Mexico's export profile is made up of aerospace parts and equipment, optical equipment, machinery, electrical and electronics, and author and auto parts.

In 2022, Mexico is seventh producer worldwide. 25 years after the signing of the NAFTA by Mexico, 89.7% of Mexico's exports were manufactured. Mexico went from being an exporting country of oil and raw materials to being an industrialized country.

One of the most negative externalities of NAFTA has been the pollution or the air of its communities as a result of its successful industrialization. Mexicali, Toluca, Ecatepec, Tlalnepantla, Netzahualcoyotl, Salamanca, Leon, Celaya, Irapuato, Mexico City and Monterrey are the most polluted cities in Mexico with PM2.5 particles, which present a medium and long-term health risk.

They exceeded the limit of PM2.5, which is from 0 to 10 µg / m³ of the WHO [37]. The first nine cities plus Monterrey are home to the industrial plant that was created after the industrial boom that transformed the country after 1994.

While the air pollution in Mexico City is explained by its 5.5 million cars, the others cities of this sad and black ranking, are medium, its pollution is associated with its accelerated process of industrialization as a result of NAFTA, now a days T- MEC.

doi: 10.13053/CyS-26-4-3989

1.1 State of Art in Data-Driven Agent-based Modelling

Agent-based modelling (ABM) aims to simulate human behavior in systems using a programming language. Fundamental to this methodology is the abstraction of human behavior with a prediction function based on quantitative and/or qualitative data, and such approaches are often considered most effective when they are kept as simple as possible (principle by Terano) [34].

One example research that used agent-based modelling was the simulation of pedestrian evacuation in the event of seismic risk at an urban scale in a study that modelled human behavior based on observations from video recordings of real events [5].

Ng, Eheart, Cai, & Braden used agent-based modelling with a Bayesian inference to analyses decision-making among farmers regarding water quality impacts at a watershed scale in carbon emission markets fielding a second-generation biofuel crop [23].

Azar & Menassa developed a model behavior occupant in commercial buildings and their impact on energy use combined a quantitative method of measuring energy use with qualitative techniques for identifying occupant behavior [2].

Klein, Kwak, Kavulya, Jazizadeh, & Becerik-Gerber proposed other model of building occupant behavior, combined sensor data and electronic building controls to reduce energy use using a multi-objective Markov decision problem to determine occupant [14].

Finally, Arel, Liu, Urbanik, & Kohls [1] designed a model of traffic signal control, governed by an autonomous intelligent agent was modelled using neural networks.

Dynamics in ABM have been presented in two dimensions: at microscopic level, which the agent level is represented by the evolution of its dynamic attributes (variables and models) caused by its interaction with other agents or the environment.

At macroscopic level is a consequence of the microscopic dynamics and its mathematical interactions agent representation is not a trivial issue and the model can only address the interaction as the functional relationship between states and parameters [25].

In recent research based on agents, this microscopic dynamic is modelled by deterministic empirical models [26]; theoretical models as objective functions and evolutionary algorithms [9, 19, 20], theoretical empirical such as interviews and algorithms [17]; logit regression [41] or novel theoretical approaches to limited rationality such as Frank-Wolfe's linearization method with the Generalized Benders' Decomposition method [21].

A novel approach is the data-driven agent model [13] that lies between the extremes of totally empirical and totally abstract models; it consists of empirical models elaborated from the data for the dynamics at the micro level using big data and the KDD process (Knowledge Discovery on Database) [8].

Recent research uses big data and machine learning for agent models such as neural networks [6, 40] and complex networks [35].

The aim of this paper is describe beliefs-desires-intentions (BDI) using a data-driven agent-based model to predict the behavioral changes of commuters who must select a means of transportation and thereby affect the dynamic demands on transportation infrastructure using novel data-driven approach with census data and J48 algorithm machine learning tool proposed by Witten & Frank [39].

This is the first work related to five cities of Guanajuato State in Mexico; in order to understand dynamics at the micro level in the selection of the means of transport and at the macro level as the emergent mobility behavior in the five cities.

1.2 Five Cities on the State of Guanajuato

Guanajuato state is located in the center of Mexico with an area of 30,460 km^2, which represents 1.6% of the national territory.

The state comprises 46 municipalities with a total population of 6,166,934 inhabitants, Guanajuato ranks 6th nationally for its number of inhabitants [10], Guanajuato more polluted cities are Leon, Silao, Salamanca, Irapuato and Celaya. Together forms one of the more important industrial cluster in America.

Leon. Located in the East of the State with a population of 1 578 626 inhabitants [11], it is the largest metropolis in the state; in the last decade it has shown strong growth.

With data from the intercensal count [10] of the inhabitants who move from home to work, the use of public transport predominates with 37%, followed by the use of private cars with 30% and the use of labor transport labor by 7% (labor transport is the means of transportation that companies provide their employees using leased buses).

It is the municipality in the State that has the most extensive bike path network with a total of 108 kms. [18].

In Leon, there is a unique integrated transport system known as Optibus; this is formed by a combination of a bus transport subsystem and a bus rapid transit subsystem. Silao de la Victoria.

Located in the East of the State bordering León and Irapuato and with a population of 189,567, it is rapidly industrializing due to having one of the first automotive plants in the State.

It has an index of registered motor vehicles of 250 (This index is calculated as the number of motor vehicles registered in circulation divided by the estimated total number of population multiplied by 1000. Less is better, the 2015 State average is 270 [12], 20% of commuters use private vehicles, 19% use public transport, and 27% use labor transport.

Irapuato. Located in the south-central part of the State, there is a mixture of agricultural, commercial and industrial activity, with a population of 574 344 and an index of registered motor vehicles of 307, the use of private vehicles predominates with 35%, public transport with 22% followed by 13% of labor transport.

Salamanca. Located in the central zone of the State, there is a mixture *of* agricultural and industrial activity, with the large PEMEX refinery Antonio M. Amor. It has a population of 273 271 inhabitants and an index of registered motor vehicles of 352. Commuters*' use* of private vehicles predominates with 34%, followed by public transport with 22% and 11% for labor transport.

Celaya. Located in the southwestern area of the state, its main economic activity is trade and services followed by industry. It has 494 304 inhabitants, being the third largest metropolis in the state. Its index of registered motor vehicles is 338 and the use of the private vehicle predominates with 34%, followed by public transport with 30% and only with 5% for labor transport.

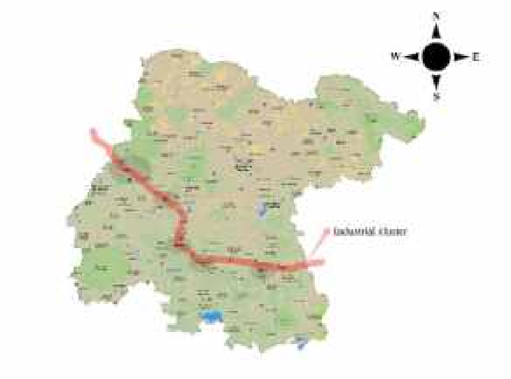


Fig. 1. Map of Guanajuato State

Figure 1 shows a map of the state, and Table 1 summarizes the population density and the number of observations used for this study [30].

2 Method

We define "agent" as a real entity that has the ability to assimilate information from its environment (input), reason (logical processes), and respond to the environmental input with a decision (output) that results in a behavior.

In operative terms an agent is a resident and commuter of the five Guanajuato cities who moves from home to work.

Consider now all the inhabitants of a city who need to move from their homes to their workplaces. Each of these commuter's reasons is based on a series of personal attributes to arrive at a decision regarding which means of transport to use, resulting in a series of behaviors that will contribute to patterns that arise with respect to the mobility of that city's residents.

Thus, a single agent exhibits individual behaviors that will cause the emergence of patterns in a system of multiple agents.

Agent-based modelling is a technique arising from systems engineering that allows modelling complex systems formed by categorized agents who are related to each other and whose interactions cause the emergence of observable patterns.

The methodology is used to simulate emerging behaviors resulting from the dynamics of socio-ecological systems [32].

The foundational elements for modelling the relationships and interactions among the agents in the system are: the data (agent attributes) and the functions that will determine the changes in those attributes in each iteration in the simulation process.

The feedback from each iteration determines the dynamics of the modelled system, starting from a given initial state [33].

The BDI agent-based model [24] includes (1) a belief system that represents the agent's values or knowledge of the environment as attributes the agent develops over time, (2) a system of desires that represents the agent's established goals related to those beliefs (here considered to be the desire to get to and from work), and (3) a system of intentions that represents the actions aimed at achieving the desired objective (here considered to be the intention to use a specific transport mode to move to and from work).

2.1 Algorithm and Language for Data-Driven ABM

Weka is a software platform developed by the machine learning group at the University of Waikato. It includes a collection of machine learning algorithms for data mining models [36].

Net Logo is a multi-agent programmable modelling environment developed by Uri Wilensky at the Centre for Connected.

Learning and Computer-Based Modelling, Northwestern University [38]. This environment allow the programming of models based on machine learning algorithms through its control structures.

The Net Logo environment enables agent-based modelling by facilitating the programming of agent intentions and their execution in a very simple way, based on code defined by the user and data that inform the creation of the intentions list.

Further details on the manner in which a BDI model is implemented in Net Logo, can be found in Sakellariou [29].

Table 1. Summary of data sample size

Metropolitan area	**Population**	**INEGI[1] commuter agents for ABM (sample size)**
Leon	1 578 626	37 400
Silao de la Victoria	189 567	6 692
Irapuato	574 344	11 405
Salamanca	273 271	7 430
Celaya	494 304	9 997
Total	3 110 112	72 924

A classifier algorithm is needed to provide an abstract data-driven model of the reasoning process based on categorical and numerical attributes.

The J48 algorithm allows quantitative and qualitative predictor variables to be used to construct a classifier tree to predict the dependent variable.

Unlike other classification algorithms such as the a priori method (also provided in Weka), in which only binary variables can be introduced to produce association rules in the form $p \to q$, the J48 classifier was programmed using if ... then... else control structures as the transition function used to determine the agent's selection of a mode of transportation.

The research used an official data sample source that included 72 924 surveys of the inhabitants of Guanajuato state's five mega-cities [10].

To model the agents, the method of Drogoul, Vamderue and Meurisse [7] is adopted for: a) abstract transportation mode selection based on the J48 decision tree, b) formal transportation mode selection, and c) the application of the model in the programming language used for the simulation. Figure 2 shows the methodology for the agent-based model.

[1] INEGI is the national institute of statistics geography and informatics.

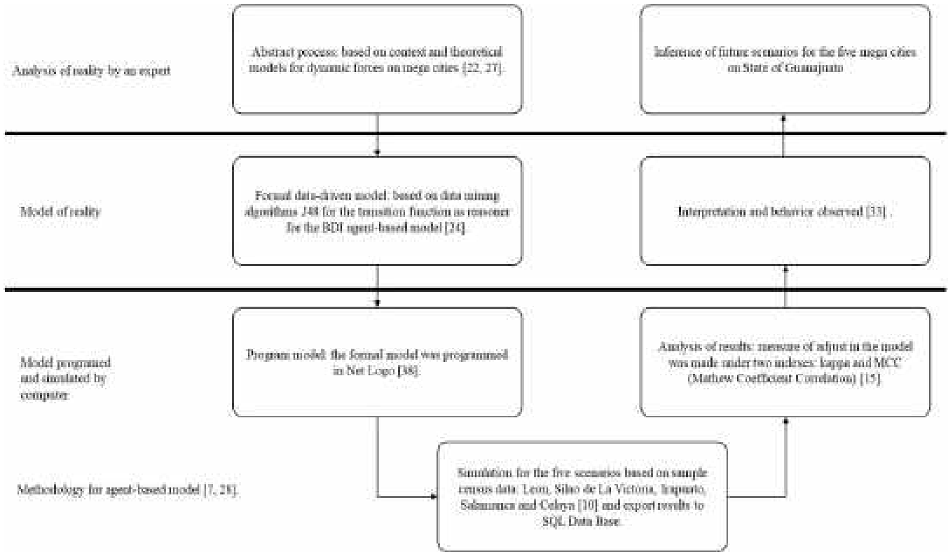

Analysis of reality by an expert
Abstract process: based on context and theoretical models for dynamic forces on mega cities [22, 27].
Inference of future scenarios for the five mega cities on State of Guanajuato
Model of reality
Formal data-driven model: based on data mining algorithms J48 for the transition function as reasoner for the BDI agent-based model [24].
Interpretation and behavior observed [33] .
Model programed and simulated by computer
Program model: the formal model was programmed in Net Logo [38].
Analysis of results: measure of adjust in the model was made under two indexes: kappa and MCC (Mathew Coefficient Correlation) [15].
Methodology for agent-based model [7, 28].
Simulation for the five scenarios based on sample census data: Leon, Silao de La Victoria, Irapuato, Salamanca and Celaya [10] and export results to SQL Data Base.

Fig. 3. Methodology for transport demand by an agent-based data-driven modelling

2.2 Scenarios for the ABM for Commuter Patterns

The initial scenarios were configured with agents whose maximum life span was 90 years, modelled independently by city, and the simulations were run to model a horizon of three years, in one-year intervals. The commuter is represented as an independent agent in the model who lives in one of the five cities.

The *initial state* of simulation are 72 924 agents modelled as the commuter types with the attributes of gender, academic level completed, age, means of transport with time of transfer for each, and the city where the agent lives.

The overall choice for transport was: Private vehicle 22 542; Bus, taxi or collective 22 225; Walk 10 835; Bicycle 9 429; Work´s transport 5 694; Not Specified 2 116 and BRT or light rail 83.

The means of transport is identified in the model as the initial means of transport for each agent and is used for comparison with the simulated means of transport at the end of the simulation. Of these agents, those that express the use of more than one means of mobility were 1,951 for two options, and 114 for three options.

The agent-based model simulates the means of transport for each independent agent based on a J38 decision tree modelled from census data. Table 2 shows a summary of the 75 070 observations of the census data used as a set for machine learning training of agents' formal predictive function using WEKA.

At each iteration (tick) of the agent-based model in Net Logo, the reasoned agent function (formal predictive function) uses as scenario input parameter: academic grade, gender, age and city and predicts the mean of transport (as output).

At the *final state* of simulation, a file is generated with, for each agent, its initial means of transport as well as those predicted by the model (can be more than one) and the accumulated transfer time for each one.

Each iteration of the program corresponds to one hour of the day, using a NetLogo extension that establishes date/time utilities and discrete event scheduling to simulate the dynamics in transportation demand [31].

The time that each agent takes from leaving home until arriving at the workplace is established randomly, considering the normal distribution specified in the census data.

Table 2. Data modelled travelers in the five cities studied

	Celaya		Irapuato		Leon		Salamanca		Silao	
Mean of transport	male	female	male	female	male	female	male	female	male	female
Not Specified	142	95	179	117	360	262	149	103	102	52
Bicycle	1 422	137	1 489	70	4 164	156	1 372	66	869	46
Walk	747	654	985	720	3 625	2 917	522	399	794	598
*** Bus, taxi or collective**	1 413	1 566	1 052	1 431	7 624	6 255	674	923	734	543
BRT or light rail	0	0	0	0	58	63	0	0	0	0
Other	58	30	110	21	245	48	42	6	35	7
Work´s transport	407	133	1 110	415	881	248	645	181	1 203	648
*** Private vehicle**	2 349	1 172	2 767	1 258	7 751	3 705	1 724	864	995	363
Totals	6 538	3 787	7 692	4 032	24 708	13 654	5 128	2 542	4 732	2 257
							Total of commuters in census dataset			75 070

* Predominant means of transport in the five cities

The time spent at work is determined randomly, with start times between 7:00 and 9:00 in the morning for work days from Monday to Saturday, and work schedules assigned randomly to between five and ten hours.

2.3 Validation for the Selection of Different Values for Parameters in Data-Driven ABM

The data-driven BDI agent-based model applied in this study is based on INEGI 2015 survey-derived demographic characteristics: gender, academic level completed, city, time of transfer and age. In a section of each survey, the inhabitant expressed: the city where he lives, gender, age, academic level completed, time of transfer to his place of work and up to three Decision.

The selection of the variables of the census questionnaire was carried out by calculating the information gain, selecting the next: Academic level with gain of 0.1752, Gender with 0.0734, City with 0.0637 and Age with 0.0373.

Trees are algorithms used by Maimon and Rokach [16] to predict an output variable based on predicates, which in this case are qualitative and quantitative agent attributes.

The most well-known algorithm of this type is C4.5, which in its most recent version is referred to as J48, implemented in the Weka data mining tool [54].

This tool is efficient and capable of handling large training sets. The precision of this classifier tree model is measured with the kappa (κ) statistic like Landis and Koch [15].

Table 3. Confusion matrix of the J48 decision tree results as output of WEKA for initial scenario

Bus, taxi, or collective	**Private vehicle**	**Bicycle**	**Walk**	**Work's transport**	**Other**	**Not specified**	**BRT or light rail**	**<-- classified as[2]**
14 491	4 884	1 843	351	646	0	0	0	Bus, taxi or collective
5 993	14 184	2 034	178	559	0	0	0	Private vehicle
4 064	1 319	**3 789**	233	386	0	0	0	Bicycle *
7 253	1 805	1 804	518	581	0	0	0	Walk
1 529	1 709	1 037	234	1 362	0	0	0	Work´s transport
276	132	154	14	26	0	0	0	Other
664	507	269	41	80	0	0	0	Not Specified
85	33	3	0	0	0	0	0	BRT or light rail
🟩 TP Bicycle: 3 789								
🟥 TN Bicycle: 4 925								
🟦 FP Bicycle: 426								
⬜ FN Bicycle: 619								

This index is used to evaluate inter-observer agreement for categorical data, similar to the ANOVA applied to quantitative data, values less than zero are poor, values between 0.21 and 0.4 are fair, between 0.6 and 0.8 is desirable, values greater than 0.81 are almost perfect adjustments.

In this research, this classifier corresponds to the prediction of a commuter's selection for his transportation mode. The Matthews correlation coefficient (MCC) [4] is used for classifications between classes (predicates) in unbalanced data (data with disproportionate frequencies).

An unbalanced data is a data set where the values are grouped by categories and the frequencies are not proportional.

The MCC provides a measure of classification performance in a set of categorical data and can be seen as a discretization of the Pearson correlation for binary variables.

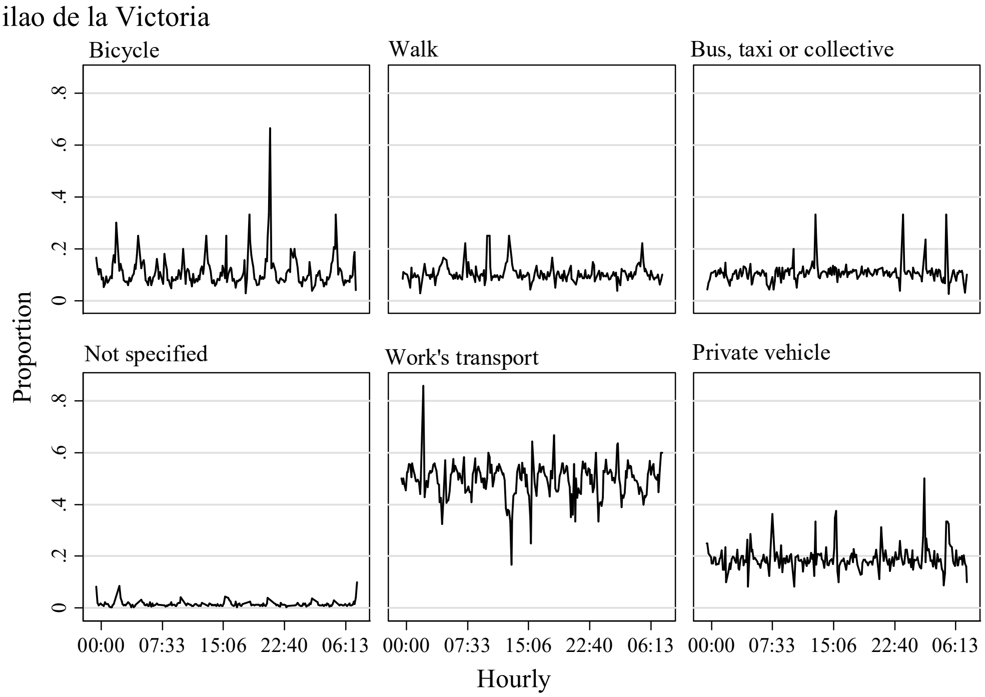
The coefficient is always between -1 and +1, a value of -1 indicates total disagreement with the classification and 1 a perfect classification; value of 0 is for completely random classification (prediction) [3, 4].

For this research, this measure is applied for the measure classification performance in the formal model and ABM model. One of the central aspects of our model is that the algorithm assumes that agents have different transport preferences depending on gender, age, city and educational level.

Thus, women and young people studying at university will be modelled as using public transport while a middle-aged man with university education will be modelled as using private transported means of transport used for that purpose.

The agent decision that the program simulates is the selection of an intended transportation mode for commuting purposes. In the program, the agent's age is a variable that advances, possibly resulting in the selection of a different transportation mode.

[2] The figure 4 shows the simulation of transportation choices in the city of Silao de la Victoria. Which was chosen to represent an intermediate case of the 5 cities. Due to space problems, the other 4 are not shown, however, the agent model developed predicts the behavior of the inhabitants of the other four cities.



10 days of simulation by the model.

Fig. 4. Diurnal oscillatory behavior results for Silao de la Victoria

For each agent, the model stores as beliefs a variable representing the cumulative total hours of use of each transportation mode in each iteration of the simulation.

The manner in which the states of agents who the simulation failed to categorize was determined as follows. Each time the model evolved, the program counted an hour for the category each agent selected as a belief.

For example, during the simulation, an agent who first walks and then uses a bicycle will have accumulated the first hours for walking and the next for bicycling.

In addition, the initial and final categories were stored as variables (InitialID and id-group, respectively), and an algorithm was applied to compare these values such that the counted beliefs in each category corresponded with the total accumulated hours.

Errors were identified when the initial or final categories hour accumulator is equal to 0 on results. Two states of the model were considered: the initial state q_i corresponding to initial data (Tables 2 and 3) and the final state corresponding to the data exported from the model (Table 4).

The final state consists in a dataset who stores the agent's attributes and the hour counter by each category on mobility that evolve in the simulation.

3 Results

In state of the art review, we could not find an investigation that allows us to compare the findings with previous works, however behavior observed in the hours of use by transportation mode indicated an oscillatory system over a day.

The classifier (or prediction) algorithm produces four outcomes: (1) True Positive (TP), which corresponds to the cases that are correctly classified in the predicted commuter means of transport; for example, it is a commuter that in the observations shows that it uses a bicycle and the algorithm predicts it as such, (2) True Negative

Table 4. Observed and predicted results

	Initial State	**Final State**	**Percent change**	**Sensitivity (TP ratio)**		**Percent change**	**MCC**		**Percent change**
	q_i	q_f	ⅆ	q_i	q_f	ⅆ	q_i	q_f	ⅆ
Walk	10 835	1 224	-89%	0.043	0.064	49%	0.068	0.137	101%
Bicycle	9 429	9 770	4%	0.387	0.41	6%	0.265	0.309	17%
BRT or light rail	83	0	-100%	0	0	0%	0	0	0
Bus taxi or collective	22 225	32 194	45%	0.652	0.677	4%	0.253	0.29	15%
Work's transport	5 694	3 568	-37%	0.232	0.278	20%	0.249	0.304	22%
Private vehicle	22 542	26 078	16%	0.618	0.661	7%	0.411	0.46	12%
Not specified	2 116	41	-98%	0	0.02	0%	0	0.137	14%
Frequency observed in demand for mobility in final state data									
One option for mobility	70 859	69 591	-2%						
Two options	1 951	3 184	63%						
Three options	114	100	-12%						
Statistics									
Po	0.458	0.495	8%						
Pe	0.262	0.268	2%						
Kappa (κ)	0.265	0.311	17%						

*q_i indicates the initial state and q_f indicates the final state of simulation

(TN), which corresponds to the sum of cases which are correctly classified as are not for the predicted category.

For example, they are all the commuters that in the observations show that they do not use a bicycle and the algorithm predicts it as such.

False Positive (FP) are the sum of cases which are classified in a predicted category but not correspond to it, as an example are all commuters that in the observations present that they do not use bicycles and the algorithm predicts that this is the medium they use and False Negative (FN) are the sum of cases which are classified as are not for the predicted category but correspond to it. As an example are all the commuters that in the observations show that they use a bicycle and the algorithm predicts another means of transport.

These values are obtained from confusion matrix as shown in table 3.

For example, the formal model predicts the transportation mode selection for commuters that live in the five cities summarized in the Table 2.

3.1 Predicted Commuter Transportation Mode Selection Patterns to Simulate the Evolution of Transport Activity

Each leaf of decision tree corresponds to a predicted transportation mode selection modelled as control structure in Net Logo program language.

For example, if the agent has an academic grade equal to primary and gender is equal to women, and lives in (Celaya, Silao de la Victoria, Leon-Salamanca, Irapuato)[3] then use bus, taxi or collective; this predicted result corresponds to one leaf of 101 possible predicted results by the model for the five cities.

The 75 070 instances correspond to commuters in census dataset used to build the tree model in Weka, distributed in the categories of transport: predominates Bus, taxi or collective 22 215 and Private vehicle 22 948; Not Specified 1 561; Bicycle 9 791; Walk 11 961; Work´s transport 5 871; BRT or light rail 121 and Other 602.

However, a little subset of 38 of commuters from 121 have academic grade of secondary, 14 women with an age mean of 29 years and standard deviation of 11 and 24 men with an age mean of 31 years and standard deviation of 10; this subset commuter description is a brief example as each one is modelled independently.

The proposed data-driven model generates a confusion matrix describing the mismatch between modelled results and survey answers. In Table 3, a Commuter classified as a walker, green and red color are correctly (True predictions) predicted, blue and grey are errors (False predictions).

In the matrix, the rows correspond to the survey answer, as an example, the bicycle means of transport is used, the sum of the row corresponds to 9 791 commuters who in the survey used this means of transport; the columns correspond to the value predicted by the algorithm, means that only 3 789 were correctly predicted, the rest of the columns were erroneous predictions: 4 064 were predicted as Bus taxi or collective, 1 319 as Private vehicles, 233 as Walk and 386 as Work's transport.

4 Discussion

Figure 4 shows ten days average simulation results for one of the five cities Silao de la Victoria[4]. With the behavior of each agent in the use of the means of transport predicted by the model and the time of use modelled in a random way, it is possible to observe the dynamic behavior of the transport demand. Demand for each category of transport is shown as the proportion of commuters in that category who are in transit.

As we can see, the use of work's transport predominates and the lower peak close to 15:00 hours explain that about 20% of commuters who use this means of transport are traveling at that time.

This sample city was selected to show the behavior in the dynamics of transport demand exhibit reality congruences as smallest of the five studied cities where industrial activity predominates and this category of transportation is used to move its employees. The BDI agent-based model showed an improvement in the kappa and MMC indices compared to table 3 and table 4 of the formal model.

This improvement, although it is a fair value in relation to Kappa index showed in Table 4, means that the agent model predicts as final state the means of transport better than the formal model J48 as initial state.

The improvement in the MMC means that the prediction of the means of transport in each agent shows an improvement compared to the initial state, observing that the model predicts each category in a moderate way as Private vehicle with 0.46.

Table 4 summarizes the results, where q_i represents the initial state and q_f represents the final state; ⅆ indicates the percent change after the simulation. The sensitivity of the J48 algorithm, indicated by the true positive (TP) ratio, is also shown. MCC was calculated using model result values.

The commuter demand for mobility predicted by the agent-based model indicates changes in the demand for transport over time. As the result of the use of a data-driven agent-based model, the prediction of the transportation mode varies as a function of the attributes of the agent.

If the age changes during the simulation, for example in the case of a 17 years old Irapuato commuter (agent modelled), at the time of

[3] The model generates predictions and a total of 101 sheets describing the future possible states of travellers

exceeding 17 years it changes from walking to the use of bus taxi or collective simulating the evolution in transport dynamics.

The observed dynamics show that commuter preferences converge towards a single transportation mode over time, with some commuters selecting two options and a minimum of commuters selecting three options.

This change in agent preferences was identified by the reasoning function modelled in the program.

The transportation demand over the three-year horizon indicated an increase in the use of taxi, bus or collective of 45% and an increase of 16% in the use of a private vehicle (automobile, van or motorcycle), which are the two main commuter transportation modes.

In the case of the city of Leon, there is an integrated transport system known as Optibus.

This is formed by a combination of a bus transport subsystem and a bus rapid transit subsystem, however commuters perceive it as transportation by bus more than a real BRT as see in table 4 because the model predicted 0 out of 83 observed.

5 Conclusion

The results show that the patterns of selection of mode of transport in the five cities with the highest air pollution in the state of Guanajuato are predicted.

The model based on can predict the decisions of selecting the mode of transport using variables age, gender, academic level and city of residence. The evolution of transport demand in five cities (macro level) indicated an increase in the use of two main modes of transport: bus and Optibus (in the case of León), taxi or collective by 45% and private vehicle (automobile), truck or motorcycle)) by 16%.

These results indicate that the citizens of the cities studied prefer to use private transport since the supply of public transport is of poor quality. However, if the supply of public transport improved citizens would dispense with their cars.

Research shows that models based on data-based agents can help us understand the evolution of a system in which a large number of people make independent decisions.

Likewise, our study shows that it is possible to construct a population-based model based on demographic surveys to predict transport variables considering: gender, level of education, transfer times and age in the five largest cities of Guanajuato, it is also able to predict the preferred means of transportation for the inhabitants of the five largest cities of Guanajuato.

Finally, it is possible to create a data-based algorithm that can simulate day changes in the choice of transport mode at the micro level. These results are similar to the findings of Witten and Frank [39].

The kappa index and the MCC used to measure the adjustments in the model observed for the algorithm based on data, in the final model (predicted scenario) and its comparison, allow observing that the adjustment values vary between regular and moderate.

The proposed model has the feasibility of being applied not only in other Mexican megalopolises with more than 1 million inhabitants the size of Leon, but also in medium - sized cities that have serious pollution problems, favoring the redesign of the transportation systems that are used in Mexico.

They find concessions to powerful interest groups that have prevented the orderly planning of a fundamental public service such as transportation and the mobility of citizens, which defines the quality of life in a city in such a relevant way.

References

1. **Arel, I., Liu, C., Urbanik, T., Kohls, A. G. (2010).** Reinforcement learning-based multi-agent system for network traffic signal control. IET Intelligent Transport Systems, Vol. 4 No. 2, pp. 128–135. DOI: 10.1049/iet-its.2009.0070.
2. **Azar, E., Menassa, C. C. (2012).** Agent-based modeling of occupants and their impact on energy use in commercial buildings. Journal of Computing in Civil Engineering, Vol. 26, No. 4, pp. 506–518. DOI: 10.1061/(ASCE)CP.1943-5487.0000158.
3. **Baldi, P., Brunak, S., Chuvin, Y., Andersen, C. A., Nielsen, H. (2000).** Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, Vol.

6, No. 5, pp. 412–424. DOI: 10.1093/bioinformatics/16.5.412.

4. **Boughorbel, S., Jarray, F., El-Anbari, M. (2017).** Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PloS one, Vol. 12, No. 6. DOI: 10.1371/journal.pone.0177678.
5. **D'Orazio, M., Spalazzi, L., Quagliarini, E., Bernardini, G. (2014).** Agent-based model for earthquake pedestrians' evacuation in urban outdoor scenarios: Behavioural patterns definition and evacuation paths choice. Safety science, Vol. 62, pp. 450–465. DOI: 10.1016/j.ssci.2013.09.014.
6. **Drchal, J., Čertický, M., Jakob, M. (2019).** Data-driven activity scheduler for agent-based mobility models. Transportation Research Part C, Vol. 98, pp. 370-390. DOI: 10.1016/j.trc.2018.12.002.
7. **Drogoul, A., Vamderue, D., Meurisse, T. (2002).** Multi agent based simulation: Where are the agents? Lecture notes in computer science, Vol. 2581. DOI: 10.1007/3-540-36483-8_1.
8. **Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996).** From data mining to knowledge discovery in databases. AI Magazine, Vol. 17, No. 13, pp. 37–54. DOI: 10.1609/aimag.v17i3.1230.
9. **Hernández, C. A., Castilla, G., López, A., Mancilla, J. E. (2016).** A multi-objective algorithm NSGA-II for programming of lamination steps in hot steel. Research in Computing Science, Vol. 120, pp. 65–80.
10. **INEGI. (2020).** Inhabitants number. https://www.cuentame.inegi.org.mx/monografias/informacion/gto/poblacion/default.aspx.
11. **INEGI. (2018).** Information by entity. http://cuentame.inegi.org.mx/monografias/informacion/gto/default.aspx?tema=me&e=11.
12. **IPLANEG. (2015).** Index of registered motor vehicles in circulation, 2015.
13. **Kavak, H., Padilla, J. J., Lynch, C. J., Diallo, S. Y. (2018).** Big data, agents, and machine learning: towards a data-driven agent-based modeling approach. In: Proceedings of the Annual Simulation Symposium, Baltimore: Society for Computer Simulation International, pp. 12.
14. **Klein, L., Kwak, J. Y., Kavulya, G., Jazizadeh, F., Becerik-Gerber, B. (2012).** Coordinating occupant behavior for building energy and comfort management using multi-agent systems. Automation in Construction, Vol. 22, pp. 525–536. DOI: 10.1016/j.autcon.2011.11.012.
15. **Landis, J. R., Koch, G. G. (1977).** The measurement of observer agreement for categorical data. Biometrics, Vol. 33, No. 1, pp. 159–174. DOI: 10.2307/2529310
16. **Maimon, O., Rokach, L. (2010).** Data mining and knowledge discovery handbook. New York: Springer. DOI: 10.1007/978-0-387-09823-4.
17. **Martínez, L. M., Correia, G. H., Moura, F., Mendes Lopes, M. (2017).** Insights into carsharing demand dynamics: Outputs of an agent-based model application to Lisbon, Portugal. International Journal of Sustainable Transportation, Vol. 11, No. 2, pp. 148159. DOI: 10.1080/15568318.2016.1226997.
18. **IMPLAN. (2016).** Cycleways master plan. https://implan.gob.mx/pdf/estudios/movilidad/plan-maestro-de-ciclovias-2016.pdf.
19. **Mogale, D. G., Kumar, S. K., Tiwari, M. K. (2016).** Two stage Indian food grain supply chain network transportation-allocation model. IFAC-PapersOnLine, Vol. 49, No. 12, pp. 49–12. DOI: 10.1016/j.ifacol.2016.07.838.
20. **Mogale, D. G., Kumar, S. K., Márquez, F. P., Tiwari, M. K. (2017).** Bulk wheat transportation and storage problem of public distribution system. Computers & Industrial Engineering, Vol. 104, pp. 80–97. DOI: 10.1016/j.cie.2016.12.027.
21. **Mogale, D., Lahoti, G., Jha, S., Shukla, M., Kamath, N., Tiwari, M. (2018).** Dual market facility network design under bounded rationality. Algorithms, Vol. 11, No. 4, pp. 54–74. DOI:10.3390/ a11040054.
22. **Molina, L. T., Molina, M. J. (2002).** Air quality in the Mexico megacity. An integrated assesment. Kluwer Academic Publishers. Vol. 2, DOI: 10.1007/978-94-010-0454-1.

23. **Ng, T. L., Eheart, J. W., Cai, X., Braden, J. B. (2012).** An agent-based model of farmer decision-making and water quality impacts at the watershed scale under markets for carbon allowances and a second-generation biofuel crop. Water Resources Research, Vol. 47, No. 9, DOI: 10.1029/2011WR 010399.
24. **Norling, E., Sonenberg, L., Rönnquist, R. (2000).** Enhancing multi-Agent based simulation with human-like decision making strategies. Multi-Agent-Based Simulation, pp. 214–228. https://doi.org. 10.1007/3-540-445 61-7_16.
25. **Pereda, M., Zamarreño, J. (2015).** Agent based model: an aproach from system engineering. Ibero-American Magazine of Automatics and Industrial Computing, Vol. 12, No. 3, pp. 304–312. DOI: 10.1016/j.riai. 2015.02.007.
26. **Pijoan, A., Kamara-Esteban, O., Alonso-Vicario, A., Borges, C. (2018).** Transport choice modeling for the evaluation of new transport policies. Sustainability, Vol. 10, No. 4, pp. 1230–1252. DOI: 10.3390/su10041230.
27. **Pinhas, A., Shvainshtein, O., Kishcha, P. (2012).** AOD Trends over megacities based on space monitoring using MODIS and MISR. American Journal of Climate Change, Vol. 1, No. 3, pp. 17–131. DOI: 10.4236/ajcc.20 12.13010.
28. **Rivas-Tovar, L. A. (2017).** Preparation of thesis: structure and metodology. Trillas.
29. **Sakellariou, I. (2010).** Agents with beliefs and intentions in Netlogo. http://users.uom. gr/~iliass/projects/NetLogo/AgentsWithBeliefs AndIntentionsInNetLogo.pdf.
30. **Salas-Rodríguez, D., Rivas, L. A. (2017).** Air pollution in five cities in Guanajuato state (México). Conference on Complex Systems.
31. **Sheppard, C. (2018).** A NetLogo extension that brings date/time utilities and discrete event scheduling to NetLogo. https://github.com/ colin sheppard.
32. **Smajgla, A., Brow, D. G., Valbuena, D., Huigene, M. (2011).** Empirical characterisation of agent behaviours in socio-ecological systems. Environmental Modelling & Software, Vol. 27, No. 7, pp. 837–844. DOI: 10.1016/j.envsoft.2011. 02.011.
33. **Sterman, J. D. (2000).** Business dynamics systems thinking and modeling for a complex world. McGraw-Hill.
34. **Terano, T. (2008).** Beyond the KISS principle for agent-based social simulation. Journal of Socio-Informatics, Vol. 1, No. 1, pp. 175–187
35. **Tomasello, M. V., Vaccario, G., Schweitzer, F. (2017).** Data-driven modeling of collaboration networks: a cross-domain analysis. EPJ Data Science, No. 22. DOI: 10.1 140/epjds/s13688-017-0117-5.
36. **University of Waikato. (2022).** Downloading and installing Weka.https://www.cs.waikato .ac.nz/~ml/weka/downloading.html.
37. **WHO. (2017).** WHO Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide. World update 2005. Summary of risk assessment. Suiza: OMS. http://apps. who.int/iris/bitstream/10665/69478/1/WHO_S DE_PHE_OEH_06.02_spa.pdf.
38. **Wilensky, U. (2017).** Net Logo. https://ccl.northwestern.edu/netlogo/.
39. **Witten, I., Frank, E. (2005).** Data mining practical machine learning tools and techniques. San Francisco: Elsevier. Vol. 31, No. 1.
40. **Zhang, Y., Grignard, A., Lyons, K., Aubuchon, A., Larson, K. (2018).** Real-time machine learning prediction of an agent-based model for urban decision-making. Proceedings of the 17th international conference on autonomous agents and multiAgent systems Stockholm: International Foundation for Autonomous Agents and Multiagent Systems. pp. 2171–2173.
41. **Zhao, C., Li, S., Wang, W., Li, X. D. (2018).** Advanced parking space management strategy design: An agent-based simulation optimization approach. Transportation Research Record, Vol. 2672, No. 8. DOI: 10.1177/0361198118758671.

Article received on 15/06/2021; accepted on 18/08/2022. Corresponding author is David Salas-Rodríguez.