






Classification of Wandering Patterns in the Elderly using Machine Learning and Time Series Analysis

Daniel Ramos-Rivera , † Arnoldo Díaz-Ramírez , *Senior Member IEEE*, Leonardo Trujillo ,
Juan P. García-Vázquez , and Pedro Mejía-Álvarez 

Abstract—Dementia has emerged as a significant health concern due to global aging trends. A degenerative brain disorder, dementia leads to cognitive decline, memory loss, impaired communication skills, reduced abilities, and shifts in personality and mood. Dementia lacks a definitive cure, but accurate diagnosis and treatment can improve the quality of life for those affected. Wandering behavior is common in patients, and a link between wandering patterns and the severity of the disease has been established. This work addresses the challenge of detecting dementia-related wandering behaviors. The proposed strategy utilizes data imputation methods and feature extraction with the Discrete Wavelet Transformation applied to a recently developed and comprehensive dataset. Machine learning algorithms are used to perform the final detection, and hyperparameter optimization is also evaluated. Experiments show that performance achieves an accuracy of approximately 98% using the Random Forest classifier. Results are competitive with the state-of-the-art in time series classification, with improved efficiency. The proposed methodology can be used for the development of applications for dementia related research and care.

Link to graphical and video abstracts, and to code: <https://latam.ieceer9.org/index.php/transactions/article/view/8969>

Index Terms—Dementia, Wandering Patterns, Feature Extraction, Discrete Wavelet Transform, Machine Learning

I. INTRODUCTION

Wandering behavior is a common symptom among patients that suffer from dementia, with some studies finding a prevalence of 67% [1]. Dementia patients experience significantly more neuronal loss and impairment than the average person as they age, which leads to symptoms that include memory loss, confusion, difficulty articulating thoughts, disorientation and wandering [2]. According to the

The associate editor coordinating the review of this manuscript and approving it for publication was Carlos Thomaz (*Corresponding author: Leonardo Trujillo*).

This paper was funded by CONAHCYT grant CF-2023-I-724, CONAHCYT scholarship 2023-000002-01NACF-02559, TecNM grant 19400.24-P, and by grants from the Secretaría de Economía e Innovación de Baja California and AWS.

D. Ramos-Rivera and † A. Díaz-Ramírez are with Tecnológico Nacional de México/IT de Mexicali, Mexicali, BC, México (e-mail: danielr@itmexicali.edu.mx). A. Díaz-Ramírez is a posthumous author of this work, deceased October 25th 2023.

L. Trujillo is with Tecnológico Nacional de México/IT Tijuana, Tijuana, B.C., México (e-mail: leonardo.trujillo@tectijuana.edu.mx).

J. P. García-Vázquez is with Universidad Autónoma de Baja California, MyDCI, Mexicali, B.C., México (e-mail: pablo.garcia@uabc.edu.mx).

P. Mejía-Álvarez is with CINESTAV-GUADALAJARA, Zapopan, Jal, México (e-mail: pmalvarez@cs.cinvestav.mx).

National Institutes of Health in the United States, dementia affects many people, particularly the elderly; around one-third of individuals aged 85 or older experience its effects [2]. It is important to emphasize that dementia is not a normal part of aging, and while it cannot be cured, measures can be taken to enhance the quality of life for patients and their loved ones. This involves considering the individual's current abilities and monitoring changes over time that may signal the early stages of dementia.

In the work by Martino-Saltzman et al. [3], the authors found a correlation between wandering patterns and the extent of dementia, corroborated in other works [4]. These patterns can be classified into four categories: direct, pacing, lapping, and random; illustrated in Fig. 1. The direct pattern involves straightforward movement between two points without deviation. The pacing pattern entails back-and-forth walking within an area. Lapping involves semicircular movement, often repeated. The random pattern sees the patient moving haphazardly toward various points without remaining within a specific area or following a recurrent pattern. The authors found that higher levels of dementia are associated with patients displaying lapping patterns, followed by those with random patterns. Conversely, patients exhibiting pacing patterns showed lower levels of dementia. The direct pattern exhibited no correlation with any level of dementia.

Various systems have been proposed to detect wandering behaviors [5], [6], which often require dedicated infrastructure in indoor environments. For instance, using ultrasound, Wi-Fi, ultra-wideband, cameras, and other sensors [7]. These systems are often costly since they require specialized hardware and infrastructure, as well as the creation of an indoor map, commonly known as a fingerprint [8], which is subsequently used to estimate an individual's movement within an environment. However, creating these fingerprints is time-consuming, and the approach has limited flexibility because it requires a special configuration for each room or area.

The present work employs Inertial Measurement Units (IMUs), which include accelerometers, gyroscopes, and magnetometers, often used for pedestrian tracking [9]. Micro Electro-Mechanical Systems (MEMS) that implement IMUs are widely available and commonly found in modern smartphones. Due to their cost-effectiveness, compact size, efficiency, and low power consumption, MEMS are an ideal choice for implementing systems that calculate pedestrian trajectories, also known as Pedestrian Dead-Reckoning [10], to create a pattern recognition model for indoor walking without the need for dedicated infrastructure [11], [12]. Other

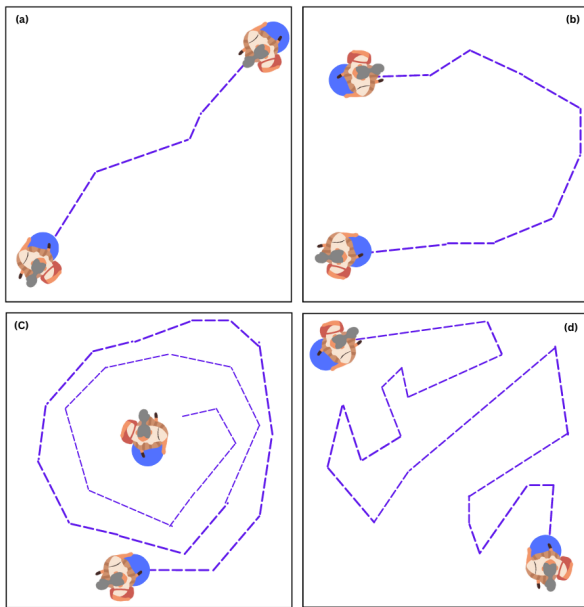


Fig. 1. Examples of wandering patterns defined in [3]: a) direct, b) pacing, c) lapping and d) random.

issues that any such system must account for include ethical concerns, governance and privacy, before considering real-world deployment [13]. While these issues are not studied in the present work, it is possible to address them with current technologies and frameworks [14].

The main contribution of this work concerns the identification of wandering patterns using IMU readings. The dataset was proposed in [15], which is noteworthy in terms of class balance and the number of samples, relative to previous works. It considers the four wandering patterns of interest when characterizing the behavior of patients that suffer from dementia, summarized in Fig. 1. The wandering patterns are recorded as the trajectories of the patients within a room, and three different time series representations are used. This is the first work that uses the dataset from [15] to build machine learning (ML) models to detect wandering patterns performing feature extraction using the Discrete Wavelet Transform, and classification using the K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF) algorithms, along with Bayesian optimization for hyperparameter tuning. To the best of our knowledge, this methodology has not been previously employed in this domain. Performance achieved is competitive with the state-of-the-art in time series classification [16], [17].

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 outlines the methodology, Section 4 presents the experimental results, and Section 5 presents the conclusions and discusses future work.

II. RELATED WORKS

In [19] an active RFID system is used to detect wandering patterns. The system detects room-to-room movements, allowing for the automatic classification of movement patterns. The goal was to assist in managing the wandering behavior of

people with dementia. The work considered all the wandering patterns of the Martino-Saltzman model. The study employed ML to classify the patterns, using eight classical algorithms, including RF, KNN, and SVM. The results consider sensitivity, specificity, precision, recall, F1-measure and latency, with the RF classifier achieving 92.3%, 92.3%, 92.2%, 92.3%, 92.2%, and 0.03s, respectively.

In [20] wireless devices and omnidirectional antennas operating at S-band frequencies are used to monitor the wandering behavior of patients. The technique involves measuring variances of amplitude and calibrated phase information to identify movement patterns, capturing the deviations relative to an ideal signal. Each pattern exhibits unique signatures in both Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS) signal propagation. A SVM is used for classification, achieving close to 90% accuracy. In [21], sensor event streams are used as features extracted from the CASAS dataset [22]. The work applied a Long Short-Term Memory (LSTM) network and the Gated Recurrent Unit (GRU) network to deal with the raw sensory data. Results achieved an accuracy of 98.6%. In this case the wandering patterns were predefined using static segments and the approach relies on non-trivial sensors and infrastructure.

In [6], an Ultra-wideband (UWB) localization system was used, along with SVM and KNN classifiers. Features included the coordinates of the patterns within a large 2D map. The dataset comprised 400 samples (10 subjects \times 4 patterns \times 10 repetitions) for both the x and y axis, yielding accuracy results ranging from 95% to 99%. However, the wandering experiments were conducted in highly controlled environments; e.g., only two trajectories of the random pattern were explicitly and statically defined, which does not resemble a real scenario. The authors evaluated their models by partitioning the data into training and test sets in four ways: Group-1 (60% train, 40% test), Group-2 (70% train, 30% test), Group-3 (80% train, 20% test), and Group-4 (90% train, 10% test). The work in [18] used data from mobile devices and deep learning models. Data is collected in controlled environments using 14 healthy subjects, where users had to directly interact with the mobile devices to begin data capture. The dataset contained 1,354 samples, and the best accuracy was approximately 85%.

In [15] the authors monitor wandering behavior considering various sensing techniques, such as infra-red, ultrasound, WiFi, ultra-wideband, vision, and IMUs. In particular, IMUs were used to extract the x and y coordinates of the movements within a room over time, which is the data considered in the present study. The authors created a data set of 1600 samples, 400 each for the 4 wandering patterns of the Martino-Saltzman model. Moreover, preliminary results using ML were obtained using a the common data partition of 70% for training and 30% for testing, and achieved an accuracy of approximately 90%. The study evaluated the use of time-series data techniques, such as autocorrelation and partial autocorrelation functions, linear discriminant analysis, multivariate Gaussian models, adaptive boosting, and KNN.

The dataset developed in [15] is a noteworthy contribution, using realistic patterns that did not rely on a set of pre-defined trajectories. The lapping and random patterns pose distinct

TABLE I

A COMPARISON OF RECENT DATASETS USED IN THIS PROBLEM DOMAIN. NOMENCLATURE: SUBJECTS WITH DEMENTIA (DS), SUBJECTS WITHOUT DEMENTIA (NDS), SECONDS (S), NOT SPECIFIED (NS), AND 'ALL' PATTERNS REFERS TO THE DIRECT, PACING, LAPPING AND RANDOM PATTERNS [3]. (*) WHILE [18] CONTAINS UNBALANCED CLASSES, THE PERCENTAGE DIFFERENCES IS SMALL BETWEEN CLASSES.

Year/Ref.	Population	Samples	Classes	Patterns	Time (s)	Comments
2014/[19]	5 ds	220	Unbalanced	ALL	9 to 1,545	Non-controlled environment, using RFID.
2018/[20]	NS	120	NS	Lapping, Pacing and Random	212	Controlled environment, using S-Band.
2020/[6]	10 nds	400	Balanced	ALL	30 to 60	Controlled environment, using UWB.
2020/[21]	1 nds	129,706	Unbalanced	ALL	15	Non-controlled environment, various ambient sensors.
2020/[18]	14 nds	1,354	Unbalanced*	ALL	22.5	Controlled environment, mobile devices.
2022/[15]	3 nds	1,600	Balanced	ALL	20 to 25	Controlled environment, using IMU.

challenges for feature extraction and classification, and are therefore often simplified in previous works. The work in [19], for instance, achieved a higher accuracy compared to [15]. However, that work, like others, utilized a relatively small dataset of 220 wandering patterns, where 97 corresponded to the direct pattern, whereas the remaining patterns were represented by no more than 36 samples. For the experiments used in [6], only two trajectories for the random pattern were defined. In [21], the wandering patterns were predefined and characterized by static segments. Moreover, that approach requires previously installed infrastructure within the rooms. The datasets reviewed so far are compared in more detailed in Table I.

The present work employs the dataset from [15] to derive a ML pipeline to detect wandering patterns. The proposal is general and not contingent on predefined patterns. While efficient inference is preferable, the proposed system does not require real-time response to provide useful feedback for caregivers. This work is directly related to a broader group of studies that perform human activity recognition, particularly those that use IMUs and ML [11], [12].

III. METHODOLOGY

The methodology employed can be described in four stages. The first was defining a dataset based on [15]. The second stage considers two approaches to characterize the wandering patterns. The first approach considers three different time series representations. The most basic representation uses the x and y coordinates of each pattern. Moreover, two derived time series are extracted, namely a Slope time series and the Path Efficiency time series, described in Section III-C. In this case the classifiers must detect patterns directly from the raw time series. The second approach applies feature extraction to each time series, to facilitate classification. The Discrete Wavelet Transform is employed and various statistical descriptors are computed, namely central tendency measures, measures of central deviation, percentiles (5%, 25%, 75%, and 95%), entropy, and the amount of non-zero crossing and non-mean crossing indices. In the third phase, ML is used to generate classification models, comparing both the raw time series and the extracted features. Finally, the fourth stage involves employing Bayesian Optimization (BO) for hyperparameter tuning. Experimental work evaluates the use of default hyperparameters as well.

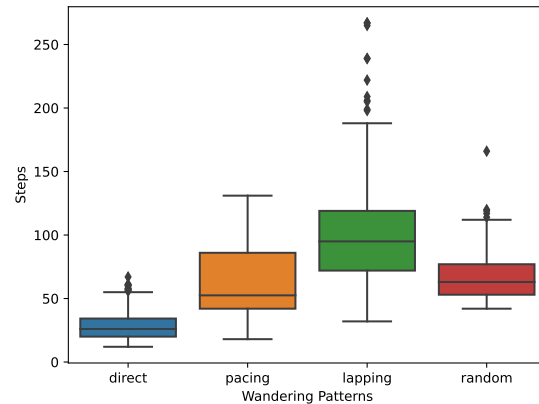


Fig. 2. Steps per wandering pattern in the dataset.

A. Dataset

The dataset contains 1,600 wandering patterns, 400 from each of the four types [3]: direct, pacing, lapping, and random [15]. There is variation in the length and duration of each pattern. In [15], they estimate the number of steps from the sensor readings, with an average step length of 0.4m, and the number of steps varies from 10 to 200 (excluding outliers), whereas duration varies from 10 to 25 seconds. Fig. 2 summarizes the number of steps for each pattern using boxplots.

B. Data Imputation

In order to have the same number of readings on each wandering experiment, a limit of 80 readings (steps) per experiment was set using the following criteria. Consider n as the total number of steps in a wandering episode, r as the result of $80 - n$ and $r/2$ as the result of $n - 80$, then:

- Case 1: if the reading has less than 80 steps, then h steps are added at the start and h' at the end, where

$$h = \lfloor \frac{r}{2} \rfloor, \quad (1)$$

$$h' = \begin{cases} h, & \text{if } r \text{ is even} \\ h + 1, & \text{otherwise.} \end{cases} \quad (2)$$

Essentially, the goal is to divide the remainder into two equal parts. Starting from this premise, the amount of

steps to be added is divided by two and rounded down to work exclusively with integer values. For h' , if h is even then h and h' are equal. Otherwise, if h is not even, due to rounding error, h' is equal to h plus one.

- Case 2: If the reading has more than 80 steps, then m number steps are subtracted at the start and m' at the end of the reading, similarly to Case 1, such that

$$m = \lfloor \frac{r2}{2} \rfloor, \quad (3)$$

$$m' = \begin{cases} m, & \text{if } r2 \text{ is even} \\ m + 1, & \text{otherwise} \end{cases} . \quad (4)$$

- Case 3: If the reading has 80 steps, then do nothing.

C. Feature Extraction

For this study, the proposed system focuses on the Cartesian coordinates of each step during the trajectory of each wandering pattern, as described in [15], organizing the data points in chronological order to create a time series. The following time series were extracted from the IMU readings.

- Coordinates (Coords): the raw coordinate data is extracted from the IMU sensors as described in [15]. These are two time series, for the x and y positions respectively.
- Slope: The slope of a line is the measure of its steepness or incline and represents the rate at which the dependent variable changes concerning the independent variable. Mathematically, the slope (m) is calculated as the ratio of the vertical change (Δy) to the horizontal change (Δx) between two points on the line: $m = \frac{\Delta y}{\Delta x}$.
- Path Efficiency (Path-Eff): ratio of the optimal path length and the actual path length, based on [23].

Fig. 3 shows examples from each time series. The plots show the Coords time series (both the x component, denoted as *Coords x* and the y component denoted as *Coords y*, as well as the Slope and Path-Eff for a single sample from each class (direct, lapping, pacing and random). Each time series is normalized to the range $[-1, 1]$ in these plots.

Feature extraction is a significant part of any ML system [24]. This work employs the Discrete Wavelet Transform (DWT), which can be used to quantify trends, inflection points, discontinuities, and self-similarities. The method leverages a set of basis functions localized within the time and frequency domains [25], and can be implemented efficiently [26]. Various families of wavelets exist, but the present work uses the Biorthogonal Wavelet rbio3.1, given its strong performance in similar domains [27].

The DWT is employed to extract descriptive features from the time series data, with the features represented by computed wavelet coefficients. The following procedure is carried out to extract a total of 12 features. First, select a time series (raw coordinates, Slope, or Path Efficiency) as an input signal. Second, apply the DWT to the time series, and extract the following features. *Entropy*, which refers to a measure of the level of disorder, randomness, or uncertainty in a dataset. Moreover, the following statistical descriptors: the 5%, 25%, 75%, and 95% percentiles; median, mean, standard deviation,

variation, and quadratic mean. The following descriptors known as *crossings* are also computed. *No-Zero Crossing Indices*, which refers to the count of instances where a signal crosses the zero axis without actually being equal to zero. It can provide information about oscillatory behavior in the signal. *No-Mean Crossings Indices*, which refers to the count of instances where a signal crosses its mean value. It offers insights into the signal's behavior around its central tendency. Finally, all of the data is scaled using min-max normalization.

D. Machine Learning Algorithms

KNN employs a distance function to identify the k nearest neighbors for classifying a new sample. In KNN classification the outcome relies on the maximum vote from its neighbors, with the number of neighbor's k determined by the user. An optimal combination of these two hyperparameters significantly impacts the model's performance. The SVM classifier (SVC) construct a decision boundary, also referred to as a hyperplane in feature space, between two classes. This hyperplane is oriented in a manner that maximizes the distance from the nearest data points of each class, termed support vectors [28]. Initially devised to tackle linear problems, SVMs developed to address non-linear problems through kernel functions. These functions introduce additional dimensions to the data, enhancing separability and leading to more efficient calculations [29]. Prominent kernel functions include polynomial, Sigmoid, and Radial Basis Function (RBF). The RF classifier combines multiple decision trees (DT) which are (weak) classifiers. Each classifier originates from a randomly sampled vector independent of the input vector, and each DT contributes a vote to classify an input sample [30].

E. Bayesian Optimization

Hyperparameter tuning was performed using BO [31], which generates posterior information through Bayesian inference for black-box optimization using the Tree Parzen Estimator (TPE). In this study TPE was implemented with $n_{\text{trials}} = 100$ and using $k = 5$ stratified cross-validation, allowing for a sufficient number of iterations to refine hyperparameters effectively and a robust evaluation for increased reliability of the performance assessments.

F. Experimental Settings

1) *Instrumentation and Sensor Specifications*: Data was acquired utilizing a Xiaomi Mi A1 smartphone equipped with; a Bosch accelerometer (model 2062600) with a resolution of 0.002 m/s^2 ; an AKM magnetometer (model 1) with a resolution of $0.01 \text{ } \mu\text{T}$; a Bosch gyroscope (model 2062600) with a resolution of 0.06 deg/s ; and other sensors [15]. The sensors were calibrated, and the sampling frequency was 500 Hz. To calibrate the gyroscope the smartphone was placed on a flat surface without moving it and then, the low pass filter was applied. To calibrate the magnetometer, the phone was slowly moved following a trajectory similar to an 8 number, and bias was removed. The sample rate was 500 Hz: more details are provided in [15].

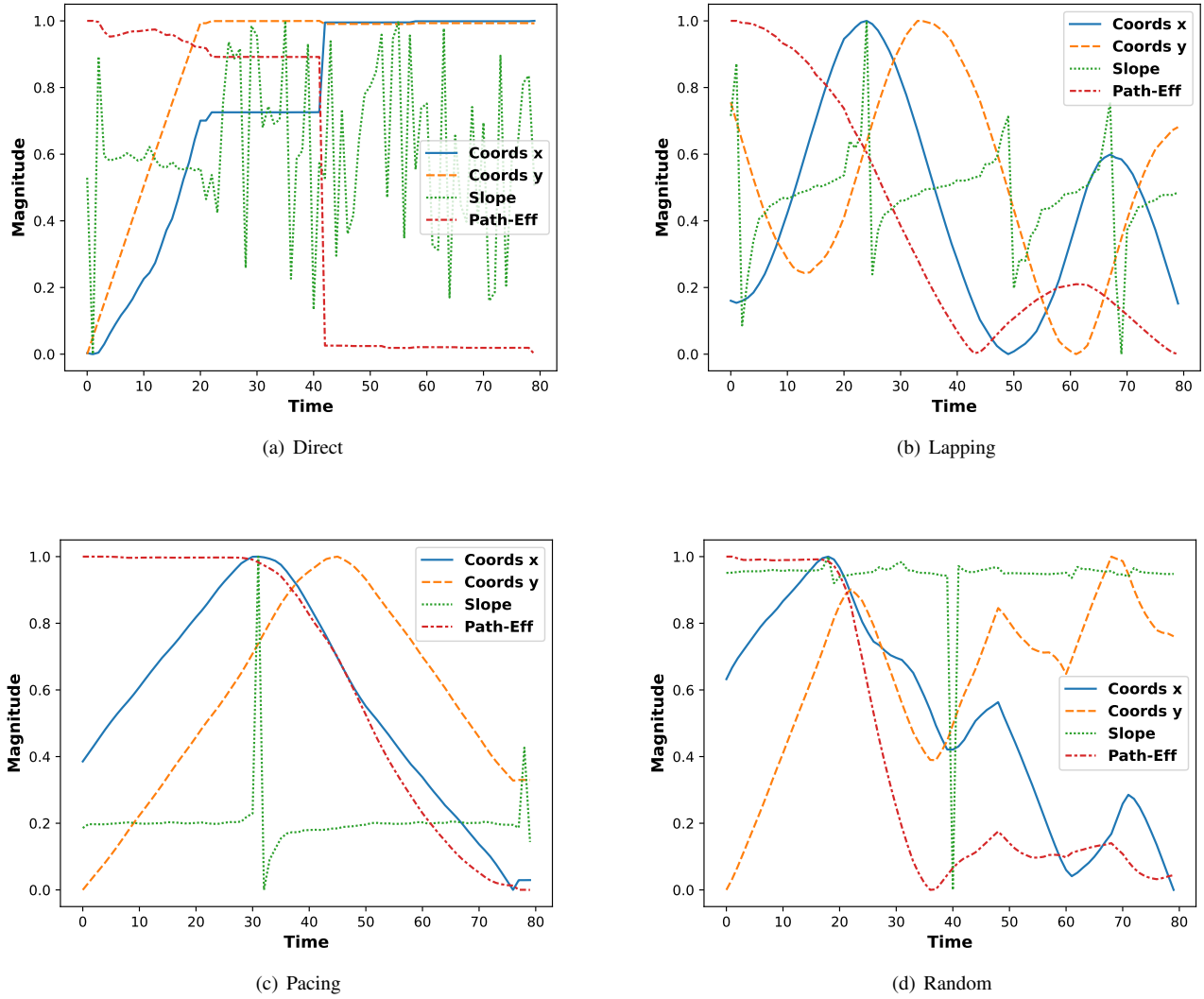


Fig. 3. Example time series for a single sample from each class: (a) Direct; (b) Lapping; (c) Pacing; and (d) Random.

2) *Hardware used for ML Training:* Experiments were conducted in the AWS cloud environment using the Amazon SageMaker service on an ml.c5.2xlarge instance, which offers 8 virtual CPUs with 16 GiB of RAM. C5 instances are equipped with either the 1st or 2nd generation Intel Xeon Platinum 8000 series processor [32].

3) *Software and Libraries:* Model implementation was carried out using the Python programming language. For the development and training of ML models Scikit-learn was employed [33]. Graphical representations were generated using Matplotlib [34] and Seaborn [35]. For BO the Hyperopt library was used [36], and for DWT the PyWavelet library¹.

4) *Evaluation metrics:* A total of 10 runs of 5-fold cross-validation were performed; i.e., 50 runs for each configuration.

For hyper-parameter optimization an additional level cross-validation (nested cross-validation) was used with three folds. Experiments were evaluated in terms of average performance, based on accuracy and F1 score, as well as macro-averages of precision, recall, and the receiver operator characteristic (ROC) area under the curve (AUC) is computed using one-vs-rest computation. The confusion matrix is computed for the best experimental configurations for additional analysis and detail.

IV. EXPERIMENTS, RESULTS, AND DISCUSSION

All experimental trials employed scaled data with the Min-Max scaler [33], all features in the range [0, 1]. Table II summarizes all the results on the test data for each experimental condition. The table presents the average score achieved in each metric, computed over all 50 runs (10 repetitions of

¹<https://pywavelets.readthedocs.io>

TABLE II
 AVERAGES OF EACH PERFORMANCE MEASURE FOR EACH EXPERIMENTAL CONDITION, SHOWING: ACCURACY, F1 SCORE, PRECISION, RECALL AND ROC-AUC. BOLD INDICATES BEST RESULTS FOR EACH GROUP OF RESULTS

Classifier	Hyperparameters	Features	Time Series	Accuracy	F1	Precision	Recall	AUC
SVC	Default	Raw	Coords	0.943	0.943	0.943	0.943	0.997
SVC	Default	Raw	Coords-Slope	0.268	0.161	0.281	0.282	0.701
SVC	Default	Raw	Path-Eff	0.848	0.848	0.865	0.848	0.964
SVC	Default	Raw	Slope	0.226	0.092	0.057	0.250	0.486
RF	Default	Raw	Coords	0.962	0.961	0.961	0.962	0.999
RF	Default	Raw	Coords-Slope	0.962	0.961	0.961	0.962	0.999
RF	Default	Raw	Path-Eff	0.891	0.890	0.893	0.891	0.981
RF	Default	Raw	Slope	0.797	0.799	0.804	0.798	0.944
KNN	Default	Raw	Coords	0.956	0.956	0.956	0.956	0.998
KNN	Default	Raw	Coords-Slope	0.685	0.680	0.683	0.686	0.887
KNN	Default	Raw	Path-Eff	0.872	0.871	0.883	0.872	0.964
KNN	Default	Raw	Slope	0.528	0.500	0.509	0.529	0.794
SVC	BO	Raw	Coords	0.962	0.962	0.962	0.962	0.999
SVC	BO	Raw	Coords-Slope	0.348	0.237	0.210	0.353	0.548
SVC	BO	Raw	Path-Eff	0.893	0.892	0.895	0.893	0.980
SVC	BO	Raw	Slope	0.241	0.097	0.060	0.250	0.491
RF	BO	Raw	Coords	0.961	0.961	0.961	0.962	0.999
RF	BO	Raw	Coords-Slope	0.961	0.961	0.961	0.962	0.999
RF	BO	Raw	Path-Eff	0.892	0.891	0.894	0.892	0.983
RF	BO	Raw	Slope	0.810	0.811	0.815	0.810	0.950
KNN	BO	Raw	Coords	0.961	0.960	0.961	0.961	0.998
KNN	BO	Raw	Coords-Slope	0.899	0.899	0.900	0.899	0.980
KNN	BO	Raw	Path-Eff	0.876	0.875	0.884	0.876	0.958
KNN	BO	Raw	Slope	0.557	0.537	0.540	0.558	0.781
SVC	Default	DWT	Coords	0.961	0.961	0.961	0.961	0.997
SVC	Default	DWT	Coords-Slope	0.963	0.963	0.964	0.963	0.998
SVC	Default	DWT	Path-Eff	0.923	0.922	0.924	0.923	0.991
SVC	Default	DWT	Slope	0.709	0.705	0.708	0.710	0.899
RF	Default	DWT	Coords	0.961	0.961	0.961	0.961	0.997
RF	Default	DWT	Coords-Slope	0.982	0.982	0.982	0.982	0.999
RF	Default	DWT	Path-Eff	0.924	0.923	0.925	0.924	0.989
RF	Default	DWT	Slope	0.842	0.841	0.843	0.842	0.965
KNN	Default	DWT	Coords	0.952	0.952	0.952	0.952	0.991
KNN	Default	DWT	Coords-Slope	0.950	0.950	0.951	0.950	0.992
KNN	Default	DWT	Path-Eff	0.901	0.899	0.905	0.900	0.971
KNN	Default	DWT	Slope	0.624	0.611	0.611	0.625	0.829
SVC	BO	DWT	Coords	0.960	0.960	0.960	0.960	0.995
SVC	BO	DWT	Coords-Slope	0.963	0.963	0.964	0.963	0.996
SVC	BO	DWT	Path-Eff	0.924	0.924	0.924	0.924	0.989
SVC	BO	DWT	Slope	0.707	0.703	0.703	0.708	0.855
RF	BO	DWT	Coords	0.962	0.962	0.962	0.962	0.997
RF	BO	DWT	Coords-Slope	0.982	0.982	0.982	0.982	0.999
RF	BO	DWT	Path-Eff	0.924	0.923	0.925	0.924	0.990
RF	BO	DWT	Slope	0.842	0.842	0.845	0.842	0.966
KNN	BO	DWT	Coords	0.958	0.957	0.959	0.958	0.992
KNN	BO	DWT	Coords-Slope	0.961	0.960	0.962	0.960	0.993
KNN	BO	DWT	Path-Eff	0.911	0.909	0.914	0.911	0.971
KNN	BO	DWT	Slope	0.693	0.687	0.693	0.692	0.889
BOSS	Default	Raw	Coords-Slope	0.939	0.937	0.941	0.939	0.996
ROCKET	Default	Raw	Coords	0.984	0.984	0.984	0.984	0.989
HIVE-COTE 2.0	Default	Raw	Coords	0.966	0.966	0.966	0.966	0.997

five-fold cross-validation). Three classifiers are used (SVC, RF, KNN), with (BO) and without (Default) hyperparameter optimization, using feature extraction (DWT) or the raw time series (Raw). Four time series are considered, Coords, Slope, Path Eff and combining the Coords and Slope (Coords-Slope) time series. Results are summarized into four large blocks in Table II, considering the use (or not) of hyperparameter optimization and feature extraction. The hyperparameter settings for each classifier are summarized in Table III. Code to replicate the experiments is available: <https://github.com/>

DanielRR10/wandering-patterns-time-series.

Results of models that use the raw data with default hyperparameter values are the first group in Table II. The model displaying the highest performance was the RF classifier utilizing the Coords time series, achieving an average accuracy of 0.962 and AUC of 0.999. The second group of results in Table II are also based on raw time series, but with hyperparameter tuning. The accuracy of most algorithms improves. For example, using the Coords-Slope time series with the KNN classifier accuracy improved by 20%. However, in most cases the improvement

TABLE III
HYPERPARAMETER VALUES USED FOR EACH CLASSIFIER,
SHOWING DEFAULT VALUES AND THE SEARCH RANGE
USED BY BO

Algorithm	Hyperparameter	Default	BO Range
KNN	k-neighbors	5	[3, 20]
KNN	Metric	minkowski	{manhattan, euclidean, chebyshev, minkowski}
KNN	Weights	uniform	{uniform, distance}
KNN	Algorithm	auto	{ball-tree, kd-tree, brute, auto}
SVC	C	1	$[10^{-2}, 10]$
SVC	Gamma	scale	$[10^{-2}, 10]$
SVC	Degree	3	[2, 5]
SVC	Kernel	rbf	{rbf, sigmoid, poly, linear}
Random Forests	Criterion	gini	{entropy, gini, log-loss}
Random Forests	Max-depth	None	[1, 21]
Random Forests	Max-features	auto	{sqrt, log2, None}
Random Forests	N-estimators	100	[50, 210]

was lower. Once again, RF achieved the highest AUC of 0.99, while the SVC classifier achieved the best performance on most metrics.

The following two groups of results use feature extraction, which led to accuracy improvements with most configurations surpassing 90% accuracy. In these groups hyperparameter optimization was even less significant than before. The best classifier, RF, was insensitive to hyperparameter tuning, with default values achieving the same performance. RF utilizing the Coords-Slope time series attained the highest accuracy of 98.2%. To illustrate the impact of the DWT feature extraction, a further analysis of two datasets is performed. The raw signal and the DWT feature extraction of the Slope time series. Consider the SVC results in Table II for both cases. Using the raw time series accuracy is 22.6% with default hyperparameters and 24.1% with BO. On the other hand, with DWT feature extraction the accuracy scores are 70.9% and 70.7%, respectively, a large performance increase in both cases. To understand the reasons, 2D visualizations of both datasets are generated using t-SNE [37], shown in Fig. 4. For the raw time series there is a very large overlap between the classes, indicating a difficult classification task. Conversely, using DWT features each class is projected to more separable multimodal clusters, explaining the performance increase.

Overall, the best results were obtained with the RF classifier. Fig. 5 presents a detailed representation of the best configurations, namely: 1) RF classifier using raw Coords time series and default hyperparameters (KNN-RD); 2) RF classifier with raw Coords-Slope time series with hyperparameter optimization (RF-RO); 3) RF classifier with DWT features and default hyperparameters (RF-FD); and 4) RF classifier with DWT features and hyperparameter optimization (RF-FO). Fig. 5 presents the average confusion matrix for each classifier. In general, performance is very strong, showing that: hyperparameter optimization is not crucial, and that DWT-based feature extraction achieves improved performance, but the raw time series can also be used to attain competitive results. The proposed approach not only achieved results closely aligned with those from [6], but also operated on a dataset that is notable and unique in this domain [15].

The last three rows of Table II show the performance of ML methods specifically designed for time series classification from SKTIME². The methods are BOSS [38], ROCKET [16] and HIVE-COTE 2.0 [17]. These methods are run without hyperparameter optimization, showing only the configuration that achieved the best results. The RF-FD configuration is extremely competitive, outperforming what is considered to be the state-of-the-art algorithm, HIVE-COTE 2.0. Moreover, while training times using the RF-FD configuration was on average 31 seconds, BOSS, ROCKET and HIVE-COTE 2.0, respectively required 1,885, 80 and 11,223 seconds. Feature extraction is not a factor, since extracting the DWT feature required no more than six few seconds for the entire dataset.

V. CONCLUSIONS AND FUTURE WORK

This paper proposes a methodology to classify four different types of wandering patterns that are highly correlated with dementia using the recent dataset from [15]. Experimental work proposes three different representations of each wandering pattern, expressed as time series. The methodology also applies the DWT to extract descriptive features and employs three ML classifiers, using default and optimized hyperparameter settings. Experimental results show that improved performance is achieved using DWT features instead of the raw time series, but the difference is less than initially anticipated. The RF classifier outperforms other ML methods, and best performance is achieved using the combined Coords-Slope time series. For the RF classifier hyperparameter optimization did not produce notable improvements. Some of the main limitations of the current work is the relative simplification of the overall problem by using a dataset derived from healthy persons, and still not addressing privacy and security issues that would be faced during real-world deployment. Therefore, future work will focus on the integration of the proposed methodology within a system that can capture and model data collected within a real-world clinical environment, fully considering the issues mentioned above. Nonetheless, results are highly competitive with the state-of-the-art in time series classification, and shows that good performance can be achieved classifying wandering patterns based on IMUs without requiring methods that require large amounts of data or computing resources.

ACKNOWLEDGMENTS

Research supported by CONAHCYT and TecNM projects CF-2023-I-724 and 19400.24-P, respectively; and Red-CI Baja projects *Detección de daño en palas de aerogeneradores con técnicas de clasificación de series de tiempo* and *Clasificación de patrones de deambulaje en personas afectadas con demencia utilizando aprendizaje automático* from the *Secretaría de Economía e Innovación* and AWS. First author supported by CONAHCYT scholarship 2023-000002-01NACF-02559.

²<https://www.sktime.net>

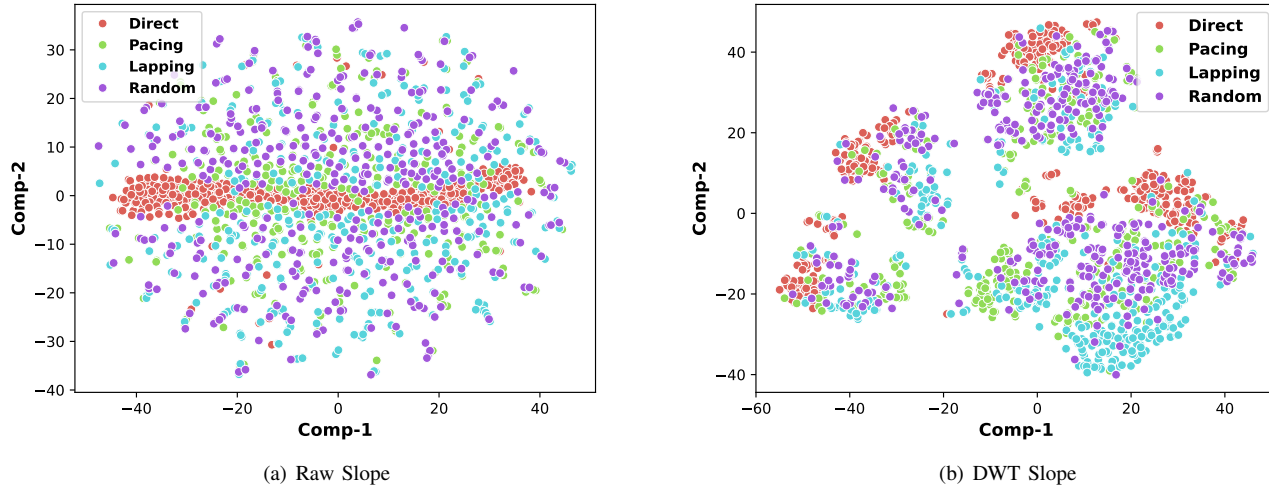


Fig. 4. Two-dimensional projections of the Slope time series data, considering: (a) the raw time series and (b) DWT feature extraction. The plots are generated using the t-SNE method.

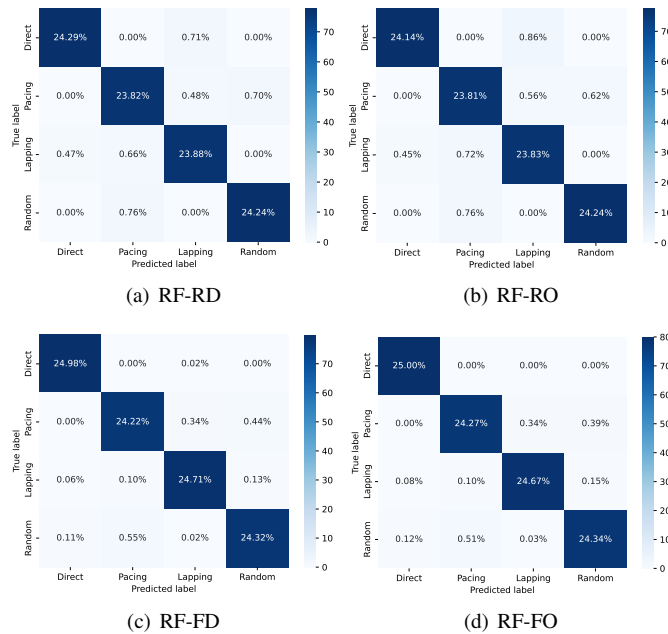


Fig. 5. Confusion matrix plots of performance scores for the best experimental configurations from the four groups of methods summarized in Table II. In each case we show the Accuracy, F1 score, Precision, Recall and ROC AUC computed on the test sets.

REFERENCES

[1] J. Wan, C. A. Byrne, M. J. O’Grady, and G. M. P. O’Hare, “Managing wandering risk in people with dementia,” *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 6, pp. 819–823, 2015. [Online]. Available: <https://doi.org/10.1109/THMS.2015.2453421>

[2] National Institute on Aging, “What is dementia? symptoms, types, and diagnosis,” 12 2022. [Online]. Available: <https://www.nia.nih.gov/health/what-is-dementia>

[3] D. Martino-Saltzman, B. B. Blasch, R. D. Morris, and L. W. McNeal, “Travel behavior of nursing home residents perceived as wanderers and nonwanderers,” *Gerontologist*, vol. 31, no. 5, pp. 666–672, Oct. 1991. [Online]. Available: <https://doi.org/10.1093/geront/31.5.666>

[4] A. Nakaoka, S. Suto, K. Makimoto, M. Yamakawa, K. Shigenobu, and K. Tabushi, “Pacing and lapping movements among institutionalized patients with dementia,” *American Journal of Alzheimer’s Disease & Other Dementias*, vol. 25, no. 2, p. 167–172, Jan. 2010. [Online]. Available: <https://doi.org/10.1177/1533317509356688>

[5] A. Hammoud, M. Deriaz, and D. Konstantas, “Wandering behaviors detection for dementia patients: a survey,” in *2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech)*, 2018, pp. 1–5. [Online]. Available: <https://ieeexplore.ieee.org/document/8448329>

[6] A. Barua, C. Dong, F. Al-Turjman, and X. Yang, “Edge computing-based localization technique to detecting behavior of dementia,” *IEEE Access*, vol. 8, pp. 82 108–82 119, 2020. [Online]. Available: <https://doi.org/10.1109/ACCESS.2020.2988935>

[7] C. Huang, Z. Liao, and L. Zhao, “Synergism of ins and pdr in self-contained pedestrian tracking with a miniature sensor module,” *IEEE Sensors Journal*, vol. 10, no. 8, pp. 1349–1359, 2010. [Online]. Available: <https://doi.org/10.1109/JSEN.2010.2044238>

[8] B. Kearns, D. Algase, D. Moore, and S. Ahmed, “Ultra wideband radio: A novel method for measuring wandering in persons with dementia,” *Gerontechnology*, vol. 7, 01 2008. [Online]. Available: <https://doi.org/10.4017/gt.2008.07.01.005.00>

[9] J. Zheng, M. Qi, K. Xiang, and M. Pang, “IMU performance analysis for a pedestrian tracker,” in *Intelligent Robotics and Applications*, Y. Huang, H. Wu, H. Liu, and Z. Yin, Eds. Cham: Springer International Publishing, 2017, pp. 494–504. [Online]. Available: https://doi.org/10.1007/978-3-319-65289-4_47

[10] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, “A comparison of pedestrian dead-reckoning algorithms using a low-cost MEMS IMU,” in *2009 IEEE International Symposium on Intelligent Signal Processing*, 2009, pp. 37–42. [Online]. Available: <https://doi.org/10.1109/WISP.2009.5286542>

[11] B. B. V. B. S. Navita Kumari, Amulya Yadagani and S. Mohanty, “Human motion activity recognition and pattern analysis using compressed deep neural networks,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 12, no. 1, p. 2331052, 2024.

[12] S. K. Challa, A. Kumar, V. B. Semwal, and N. Dua, “An optimized deep learning model for human activity recognition using inertial

- measurement units,” *Expert Syst.*, vol. 40, no. 10, Dec. 2023. [Online]. Available: <https://doi.org/10.1111/exsy.13457>
- [13] *Ethics and governance of artificial intelligence for health: WHO guidance*. Genève, Switzerland: World Health Organization, June 2021. [Online]. Available: <https://www.who.int/publications/i/item/9789240029200>
- [14] B. Liu, M. Ding, S. Shaham, W. Rahayu, F. Farokhi, and Z. Lin, “When machine learning meets privacy: A survey and outlook,” *ACM Comput. Surv.*, vol. 54, no. 2, mar 2021. [Online]. Available: <https://doi.org/10.1145/3436755>
- [15] A. Díaz-Ramírez, J. E. Miranda-Vega, D. Ramos-Rivera, D. A. Rodríguez, W. Flores-Fuentes, and O. Sergiyenko, “Time series data processing for classifying wandering patterns in people with dementia,” *IEEE Sensors Journal*, vol. 22, no. 11, pp. 10 196–10 206, 2022. [Online]. Available: <https://doi.org/10.1109/JSEN.2021.3123543>
- [16] A. Dempster, F. Petitjean, and G. I. Webb, “Rocket: exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, vol. 34, no. 5, p. 1454–1495, July 2020. [Online]. Available: <https://doi.org/10.1007/s10618-020-00701-z>
- [17] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, and A. Bagnall, “Hive-cote 2.0: a new meta ensemble for time series classification,” *Machine Learning*, vol. 110, no. 11–12, p. 3211–3243, Sept. 2021. [Online]. Available: <http://dx.doi.org/10.1007/s10994-021-06057-9>
- [18] N. K. Vuong, Y. Liu, S. Chan, C. T. Lau, Z. Chen, M. Wu, and X. Li, “Deep learning with long short-term memory networks for classification of dementia related travel patterns,” in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2020, pp. 5563–5566. [Online]. Available: <https://doi.org/10.1109/EMBC44109.2020.9175472>
- [19] N. K. Vuong, S. Chan, and C. T. Lau, “Automated detection of wandering patterns in people with dementia,” *Gerontechnology*, vol. 12, pp. 127–147, 2014. [Online]. Available: <https://api.semanticscholar.org/CorpusID:62610392>
- [20] X. Yang, S. A. Shah, A. Ren, N. Zhao, D. Fan, F. Hu, M. Ur Rehman, K. M. von Deneen, and J. Tian, “Wandering pattern sensing at s-band,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 6, pp. 1863–1870, 2018. [Online]. Available: <https://doi.org/10.1109/JBHI.2017.2787595>
- [21] A. Chaudhary, H. P. Gupta, K. K. Shukla, and T. Dutta, “Sensor signals-based early dementia detection system using travel pattern classification,” *IEEE Sensors Journal*, vol. 20, no. 23, pp. 14 474–14 481, 2020. [Online]. Available: <https://doi.org/10.1109/JSEN.2020.3008063>
- [22] W. S. University, “Casas datasets.” [Online]. Available: <https://casas.wsu.edu/datasets/>
- [23] A. Kumar, C. T. Lau, S. Chan, M. Ma, and W. D. Kearns, “A unified grid-based wandering pattern detection algorithm,” in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016, pp. 5401–5404. [Online]. Available: <https://doi.org/10.1109/EMBC.2016.7591948>
- [24] K. AlSharabi, Y. Bin Salamah, A. M. Abdurraqueeb, M. Aljalal, and F. A. Alturki, “EEG signal processing for alzheimer’s disorders using discrete wavelet transform and machine learning approaches,” *IEEE Access*, vol. 10, pp. 89 781–89 797, 2022. [Online]. Available: <https://doi.org/10.1109/ACCESS.2022.3198988>
- [25] P. Mao and R. Aggarwal, “A novel approach to the classification of the transient phenomena in power transformers using combined wavelet transform and neural network,” *IEEE Transactions on Power Delivery*, vol. 16, no. 4, pp. 654–660, 2001. [Online]. Available: <https://doi.org/10.1109/61.956753>
- [26] A. Ukil and A. Barlocher, “Implementation of discrete wavelet transform for embedded applications using tms320vc5510,” in *2007 International Symposium on Industrial Embedded Systems*, 2007, pp. 357–360. [Online]. Available: <https://doi.org/10.1109/SIES.2007.4297361>
- [27] S.-H. Wang, T.-M. Zhan, Y. Chen, Y. Zhang, M. Yang, H.-M. Lu, H.-N. Wang, B. Liu, and P. Phillips, “Multiple sclerosis detection based on biorthogonal wavelet transform, rbf kernel principal component analysis, and logistic regression,” *IEEE Access*, vol. 4, pp. 7567–7576, 2016. [Online]. Available: <https://doi.org/10.1109/ACCESS.2016.2620996>
- [28] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, “Applications of support vector machine (SVM) learning in cancer genomics,” *Cancer Genomics Proteomics*, vol. 15, no. 1, pp. 41–51, Jan. 2018. [Online]. Available: <https://doi.org/10.21873/cgp.20063>
- [29] W. S. Noble, “What is a support vector machine?” *Nature Biotechnology*, vol. 24, no. 12, pp. 1565–1567, Dec 2006. [Online]. Available: <https://doi.org/10.1038/nbt1206-1565>
- [30] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [31] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, “Hyperparameter optimization for machine learning models based on Bayesian optimization b,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019. [Online]. Available: <https://doi.org/10.11989/JEST.1674-862X.80904120>
- [32] AWS, “Amazon EC2 C5 Instances — Amazon Web Services (AWS).” [Online]. Available: <https://aws.amazon.com/ec2/instance-types/c5/>
- [33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>
- [34] J. D. Hunter, “Matplotlib: A 2D graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007. [Online]. Available: <https://doi.org/10.1109/MCSE.2007.55>
- [35] M. L. Waskom, “Seaborn: statistical data visualization,” *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021. [Online]. Available: <https://doi.org/10.21105/joss.03021>
- [36] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, “Hyperopt: a Python library for model selection and hyperparameter optimization,” *Computational Science and Discovery*, vol. 8, no. 1, p. 014008, 2015. [Online]. Available: <https://doi.org/10.1088/1749-4699/8/1/014008>
- [37] L. van der Maaten and G. E. Hinton, “Visualizing high-dimensional data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [38] P. Schäfer and M. Höggqvist, “SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets,” in *Proceedings of the 15th International Conference on Extending Database Technology*, ser. EDBT ’12. New York, NY, USA: Association for Computing Machinery, 2012, p. 516–527. [Online]. Available: <https://doi.org/10.1145/2247596.2247656>



Daniel Ramos-Rivera received the B.S. degree in computer systems with the Tecnológico Nacional de México/IT de Mexicali, Mexico in 2021. He is currently pursuing the M.Sc. degree in computer science at Tecnológico Nacional de México/IT de Mexicali. His research interests include deep learning, data analytics, e-health, cloud computing, and software systems.



Arnoldo Díaz-Ramírez (Member, IEEE) received the bachelor’s degree in computer sciences from CETYS Universidad, Mexicali, Mexico, in 1988, and the Ph.D. degree in computer sciences from the Universitat Politècnica de Valencia, Spain, in 2006, with a focus on scheduling of real-time systems. From 1992 to 2023 he worked for the Tecnológico Nacional de México/Instituto Tecnológico de Mexicali (TecNM/ITM) campus, as a Research Professor and coordinator of the Industrial Informatics Research Group, TecNM/ITM. His research interests included real-time systems, cyber-physical systems, ubiquitous computing, ambient assisted living, e-health, artificial intelligence, and wireless sensor networks. Professor Díaz-Ramírez passed away on October 25th 2023.



Leonardo Trujillo received a doctorate in computer science from the CICESE Research Center in Mexico. He is currently a Professor at the Tecnológico Nacional de México/IT Tijuana, in Tijuana, Mexico. His work focuses on genetic programming and machine learning, working on the development of new learning methods and applications on a variety of problem domains. He has been the PI of several national and international research grants, receiving several distinctions from the Mexican Science Council (CONAHCYT). His work has been published in

over 150 publications, and he is currently Editor-in-Chief of the Genetic Programming and Evolvable Machines journal (Springer).



Juan P. García-Vázquez received the degree of Doctor of Sciences from the Institute of Engineering at the Autonomous University of Baja California (UABC). Currently, he works in the Engineering Faculty of UACB, for the program of Computer Systems. Previously, he completed a postdoctoral stay at the Institute of Higher and Technological Studies of Monterrey (ITESM), in the Intelligent Environments laboratory. He is a member of the National System of Researchers Level 1 and a member of the Mexican Computing Academy (AMEXCOMP)

and has the PRODEP Desirable Profile. His interests are: Human-Computer Interaction, Machine Learning, Artificial Intelligence, Ubiquitous Computing and Computer Vision.



Pedro Mejía-Álvarez received the B.S. degree in computer systems from ITESM, Queretaro, Mexico, in 1985, and the Ph.D. degree in informatics from the Polytechnic University of Madrid, Spain, in 1995. He was a PosDoctoral fellow and visiting Professor at the University of Pittsburgh in 1999-2000. He has been Professor at Cinvestav-IPN, since 1997. His main research interests are mobile computing, real-time systems scheduling, adaptive fault tolerance, and software engineering.