



HAL
open science

POPAyI: Muscling Ordinal Patterns for low-complex and usability-aware transportation mode detection

Isadora Cardoso-Pereira, João B Borges, Aline Carneiro Viana, Antonio A. F. Loureiro, Heitor S Ramos

► **To cite this version:**

Isadora Cardoso-Pereira, João B Borges, Aline Carneiro Viana, Antonio A. F. Loureiro, Heitor S Ramos. POPAyI: Muscling Ordinal Patterns for low-complex and usability-aware transportation mode detection. IEEE Internet of Things Journal, inPress. hal-04417507

HAL Id: hal-04417507

<https://inria.hal.science/hal-04417507>

Submitted on 25 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

POPAyI: *Muscling* Ordinal Patterns for low-complex and usability-aware transportation mode detection

Isadora Cardoso-Pereira, João B. Borges, Aline C. Viana, Antonio A. F. Loureiro, Heitor S. Ramos

Abstract—Detecting transportation modes’ usability in spatiotemporal urban trajectories can provide valuable insights into the mobility preferences of urban populations, helping epidemic prevention and urban quality-of-life improvement. With this goal, we introduce *POPAyI*, a strategy that bases its design on the Ordinal Pattern (OP) transformation applied to mobility-related time series. *POPAyI* can quantify time-series dynamics with a low-complex cost, *muscling* time series’ characteristics without the need for high computational and methodological complexities as the current Machine Learning (ML) and Deep Learning (DL) literature. *POPAyI* uses polar representation and captures amplitude information in time series, bringing the multivariate capability to the standard 1D OP transformation. Our experiments show that *POPAyI*: (i) perfectly adapts to multi-dimensional mobility time series and natural non-linear mobility behavior. (ii) presents consistent detection results in any considered number of transportation mode’s classes with efficiency in terms of storage and computation complexity, using fewer features than ML approaches and computational resources than DL methods, e.g., reaching 10000 fewer parameters than a lightweight DL approach while increasing by 3% the F1-score.

Index Terms—Multivariate ordinal patterns, transportation mode detection, pattern recognition, time series classification

I. INTRODUCTION

Transportation Mode Detection (TMD) involves classifying mobility traces to identify the corresponding transport mode. It can provide valuable insights into the mobility preferences of urban populations and, in a broader sense, help tackle the consequences of urbanization, such as helping epidemic prevention, traffic management, and improving quality of life (e.g., road safety and carbon footprint control) [1, 2].

Most TMD literature relies on hand-crafted features (reaching hundreds of features [1, 3, 4]) or computationally- and data-intensive DL methods [5, 6], which pose challenges for resource-constrained Internet of Things (IoT) scenarios [2, 7]. We tackle TMD differently by leveraging mobility analysis with Ordinal Patterns (OP). OP provides a symbolic representation of time series dynamics, capturing intrinsic characteristics without relying on predefined models or significant computational resources. This lightweight solution is well-suited for edge computing applications, allowing on-device analytics and preserving privacy by processing data locally [7].

I. Cardoso-Pereira, A. A. F. Loureiro, and H. S. Ramos are with Federal University of Minas Gerais (UFMG), Brazil. E-mail: {isadoracardoso,loureiro,ramosh}@dcc.ufmg.br.

A. C. Viana is with Inria, France. E-mail: aline.viana@inria.fr.

J. B. Borges is with Federal University of Rio Grande do Norte (UFRN), Brazil. E-mail: joao.borges@ufrn.br.

Copyright (c) 2024 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

However, OP has its limitations. Initially designed for 1D time series, it may not directly adapt to multidimensional data such as mobility. Previous proposals [8–11] converted 2D into 1D data using dimensionality reduction or distance-based projections, which is often time-consuming, complex, and do not capture non-linear movement. Another limitation is not registering amplitude, rendering it oblivious to magnitude variations; e.g., the points (2, 5) and (2, 500) form the same OP symbol. This oversight may lead to lower performance, as amplitude carries valuable information, particularly in mobility, where it reflects transport displacement. Including amplitude can significantly enhance the ability to distinguish between transports and provide insights into urban conditions over time. However, literature incorporates it as a correcting factor in features extracted from OP, limiting its applicability to other OP-based features and representations [12, 13].

In this context, we propose *POPAyI* (Polar Ordinal Patterns with Amplitude Information), a multivariate OP approach. It extracts linear and non-linear motion using the polar form of 2D coordinates and incorporates amplitude information by inferring entity displacement through coordinate distances. *POPAyI* is the first study that includes amplitude into a multivariate OP transformation. These strategies make *POPAyI* effective in capturing transportation behaviors, *muscling* OP, and bringing low-complex and usability-aware TMD.

In summary, our contributions are: (1) we design POP, a multivariate OP approach for TMD that uses polar coordinates, being more suitable for mobility data by considering movement aspects. (2) We incorporate amplitude information into POP to obtain *POPAyI* leveraging TMD results. (3) We validate *POPAyI* using two well-known mobility datasets: Cologne and GeoLife. We compare *POPAyI* with seven ML- and DL-based literature proposals, showing that it is a lightweight alternative with competitive results and lower complexity. *POPAyI* can achieve equivalent results to DL approaches using 10^3 to 10^6 times fewer parameters, e.g., we can increase the F1-score by 3% using 10000 fewer parameters than a lightweight DL approach. We can also perform similar classification to ML methods with about 90 fewer features.

We organize this work as follows. Section II provides essential preliminary definitions (e.g., TMD problem statement) and OP theory. Section III reviews relevant prior work in TMD. Section IV introduces *POPAyI*. Section V discusses our results. Section VI concludes this work and explains future directions.

II. RATIONALE

Section II-A presents essential preliminary definitions of TMD. Sections II-B and II-C, respectively, introduce OP

transformation and OP limitations for TMD, which we tackle with our approach.

A. Preliminary Definitions

Our technique relies on the ability to capture intricate details and changes in transportation modes from 2D spatiotemporal data. Therefore, we prioritize datasets that offer high precision and are sampled at a frequency of few seconds. Specifically, datasets that provide samplings of 2D geodesic coordinates.

In this scenario, a *trajectory* is the mobility record in chronological order as $\mathbf{T}(t) = \{p_1, \dots, p_n\}$. Each point $p_i = (x_i, y_i, t_i)$, $i \in \mathbb{N}$, represents a geospatial coordinate set and a corresponding timestamp [1], indicating the location of a mobile entity at a particular time. Since x and y vary together over time, we consider a trajectory as a multivariate time series $T = \{X, Y\}$, where $\mathbf{X}(t) = \{x_1, \dots, x_n\}$ and $\mathbf{Y}(t) = \{y_1, \dots, y_n\}$. We also consider trajectories of varying lengths n generated by distinct transportation modes.

TMD involves two fundamental steps: segmentation and classification. In *segmentation*, a trajectory is divided into distinct non-overlapping segments $\mathbf{S}(t) = \{x_1, \dots, x_k\}$, where $k \leq n$. Each segment corresponds to a consecutive sequence of points associated with a single transportation mode, and a point belongs to only one segment. To ensure comparability with existing literature, we split trajectories as described in Zheng et al. [1], at either a stationary point (where the interval between two consecutive points exceeds 20 minutes) or the end of the trajectory. We exclusively consider trajectories (and segments) with a single associated transportation mode, relying on ground-truth data for this determination. Various segmentation methods can be employed to extract these segments, such as fixed-length segments [5] or change-point detection [1].

The second step, *classification*, focuses on predicting the transportation mode for each segment. Preprocessing techniques are employed to enhance distinctive characteristics, facilitating accurate identification. Again, for consistency with existing literature and to uphold feature quality, we adhered to the same preprocessing strategies utilized in Dabiri and Heaslip [5], Zheng et al. [1], and Xiao et al. [3]: exclusion of segments with fewer than 10 points due to limited information and removal of out-of-range coordinate points and outliers in speed and acceleration to mitigate the potential for inaccurate measurements that could impact TMD results.

B. Ordinal Patterns Transformation

OP can be applied to any time series, as it does not rely on specific model assumptions, merely replacing values in the same neighborhood with patterns based on their sequence [14]. It needs two parameters: *embedding dimension* $D \in \mathbb{N}$ to determine the length of the patterns, and *embedding delay* $\tau \in \mathbb{N}$ to define the interval between consecutive data points [14]. Formally, OP is defined as follows. Considering a time series $\mathbf{X}(t)$, at each time instant $t = \{1, \dots, n - (D - 1)\tau\}$, there is a sliding window $s_t \subseteq x$ of size D such as $s_t = \{x_t, x_{t+\tau}, \dots, x_{t+(D-2)\tau}, x_{t+(D-1)\tau}\}$. In other words, OP obtains each element within the sliding window in the

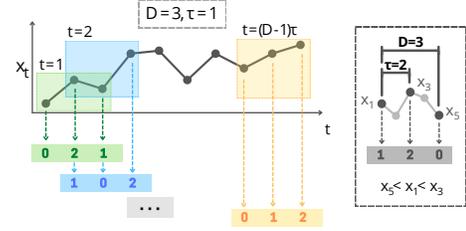


Fig. 1: Transforming time series into OP symbols.

time $t, \dots, t + (D - 1)\tau$ by sampling the time series at evenly spaced intervals, separated by intervals of size τ . In each t , OP determines an ordinal relationship between the points within the sliding window: the necessary reordering to sort these points in ascending order. Hence, the time series is converted to a set of ordinal patterns, $\Pi = \{\pi_1, \dots, \pi_m\}$, where $m = n - (D - 1)\tau$ and each π_m represents a pattern of the possible permutation set $D!$ [15].

Figure 1 depicts how OP works. In the highlight, we see the parameters: each sliding window contains D points with a interval of τ . For instance, for $D = 3$, we have the first sliding window in $t = 1$, that is, $s_1 = \{1, 5, 2\}$, forming the pattern $\pi_m = 021$. The choice of D must satisfy the condition $n \gg D!$ for the sake of reliability of the statistics estimated by the technique [15, 16]. For practical purposes, Bandt and Pompe [14] recommend values in the range $3 \leq D \leq 7$.

1) **Ordinal Patterns Probability Distribution:** Other representations can be obtained after extracting the ordinal patterns from the time series, such as a probability distribution. It depicts the number of appearances of a particular pattern π_m in the symbolic time series. Thus, the histogram of the probability distribution $P \equiv \{p(\pi_m)\}$ is defined as $p(\pi_m) = |d_{\pi_m}|/m$, where $|d_{\pi_m}| \in \{0, \dots, m\}$ is the number of observed patterns.

2) **Ordinal Patterns Transition Network:** Another representation derived from OP transformation is the Ordinal Patterns Transition Network (OPTN). It is defined as a weighted directed graph $G_{\pi_m} = (V, E)$, with vertices $v_{\pi_i} \in V = \{v_{\pi_i} : i = 1, \dots, D!\}$ and edges $E = \{(v_{\pi_i}, v_{\pi_j}) : v_{\pi_i}, v_{\pi_j} \in V\}$ representing the observed patterns π_m and the transitions between two sequential patterns π_i and π_j , respectively. The edge weights $w : E \rightarrow \mathbb{R}$ are the probability of a specific transition occurring, given by $w(v_{\pi_i}, v_{\pi_j}) = \frac{|\Pi_{\pi_i, \pi_j}|}{m-1}$, where $|\Pi_{\pi_i, \pi_j}| \in \{0, \dots, m-1\}$ is the number of transitions between π_i and π_j . Additionally, it satisfies $\sum_{v_{\pi_i}, v_{\pi_j}} w(v_{\pi_i}, v_{\pi_j}) = 1$.

From these new representations, it is possible to extract features, such as Information Theory quantifiers, which we can use to characterize the time series dynamics [16].

C. OPs' Limitations for TMD

Considering TMD, OP exhibits linear complexity, making it suitable for resource-limited scenarios (e.g., online detection and IoT applications) and large time series processing. OP is also robust to observational and dynamic noise, making it resilient to measurement errors. Furthermore, OP is invariant to non-linear monotonic transformations, i.e., it is insensitive

to amplitude variation [15–17]. Despite these benefits, OP has certain disadvantages for TMD, such as:

- **Absence of amplitude information.** The invariance to non-linear monotonic transformation is helpful against noise, but it can be harmful when amplitude contains essential information. For instance, using the standard OP transformation ($D = 3$ and $\tau = 1$) in two vehicles segments with speeds (in m/s) $v_1 = \{2, 5, 8\}$ and $v_2 = \{2, 10, 19\}$ result in the same pattern $\pi_m = 012$ (i.e., speed increases over time), though their noticeable amplitude differences. Hence, amplitude absence may impact TMD since speed gradient diversity and spatial movement dynamism are not captured.
- **Originally defined for univariate time series.** The standard OP works for univariate time series. However, many phenomena have more than one temporal component, and we cannot observe these components in isolation. In mobility, we have latitude and longitude, which depend on time and each other. Hence, when considering two points x_i and x_{i+1} , where each $x_i = (lat_i, long_i)$, though their information changes in time, there is no intuitive way to establish an ordinal relationship between the components. So OP cannot be trivially generalized to this new scenario.

POPAYI overcomes these limitations by using a multivariate approach and incorporating amplitude information, resulting in improved performance and a more comprehensive representation of spatiotemporal mobility behavior. We will examine *POPAYI* in detail in section IV.

III. RELATED WORK

While some studies have explored incorporating external information into sensor-sourced data to enhance TMD accuracy, this approach is often impractical due to the frequent collection of additional knowledge required [18, 19]. Instead, *POPAYI* utilizes high-resolution spatiotemporal mobility data, such as GPS traces, to extract features that capture the distinctive characteristics of transportation modes. Consequently, in our literature review, we prioritize works that align with our non-invasive approach in the same data kind and perform feature extraction as a crucial component of their methodology. We also discuss 1D and 2D OP approaches, evaluating their effectiveness in analyzing multivariate spatiotemporal data.

1) **Transportation Mode Detection:** Most TMD literature focuses on feature extraction for ML. Traditional ML methods, such as tree-based ensemble algorithms, have demonstrated high accuracy, reaching up to 90% [3, 4]. However, these approaches often involve extracting numerous statistical and domain-specific features, which can range from dozens [1] to hundreds [3], which can be time-consuming, require domain knowledge, and suffer from the curse of dimensionality.

To avoid using these hand-crafted features, some studies employ supervised [2, 5, 6, 20–23] and semi-supervised [24–26] DL techniques to extract multiple layers of features automatically, often yielding comparable or even superior results than ML. Still, they demand significant computational resources and large volumes of training segments with equal length, which require interpolation or padding in real-world data. Additionally, their extracted features are highly abstract and non-intuitive, making interpretation challenging. Recently, DL with

image-based features emerged in TMD, aiding the capture of spatial information, local patterns, and global context [27–31], but entailing temporal information loss, increased computational requirements, and more data preprocessing complexity. Therefore, we aim for a method that achieves high detection results using minimal features and computational resources, striking a balance between effectiveness and practicality.

2) **Ordinal Patterns:** OP [14] is a vital contribution to studying time series dynamics in several domains [15–17], including TMD. For instance, Zhang et al. [32] extracted Permutation Entropy (PE) [14] from OP and used it along with statistical features to identify the transports that generated GPS trajectories, qualifying PE as a great feature in this task due to its low computational complexity. Our previous work [11] extended such investigation to other features, such as Statistical Complexity (SC) and probability of self-transition, extracted from OPTN. We showed that features extracted from OP and OPTN help identify transports in scenarios with fewer data, such as IoT contexts. However, as OP is originally for univariate time series, these mentioned studies need to transform mobility into a one-dimensional space, which can be time-consuming and complex.

Regarding the multivariate OP transformations, studies suggested joining the OP representation of each temporal component in a matrix [8, 33], which increases OP time and space complexity and demands more extensive time series to extract reliable statistics, making it impractical in certain domains. Another proposal combined dimensions into a 1D projection with PCA [8], creating abstract time series that are hard to interpret. Alternatively, researchers calculated Euclidean and Manhattan distances between time series points and a reference point [9], which may lose essential information by incorrectly modeling mobility (e.g., not considering turns).

Furthermore, amplitude information is crucial in various fields, including mobility, where it represents the displacement of entities and plays a vital role in TMD. Neglecting this information can significantly degrade classification performance [34]. Some studies incorporated amplitude directly in PE [12, 13, 35], limiting their application in other features and OP representations. Sun et al. [36] created amplitude symbols by dividing the time series plane into equal regions, which can lead to large distributions and affect the representativeness of extracted statistics. No studies consider amplitude information in multivariate OP to the best of our knowledge.

3) **POPAYI positioning:** In *POP transformation*, we used *polar coordinates* instead of the original 2D geodesic coordinates, effectively capturing the non-linear aspects of spatiotemporal mobility without increasing OP complexity, unlike previous studies. We then enhance POP with amplitude information, enabling its integration into any feature or OP-based transformation with low computational costs. To the best of our knowledge, *POPAYI is the first work to leverage amplitude information in a multivariate OP approach*. We will discuss *POPAYI's* design in the following section.

IV. POPAYI DESIGN

Section IV-A introduces *POPAYI*, an extension of standard 1D OP transformation for 2D data, specifically focused on

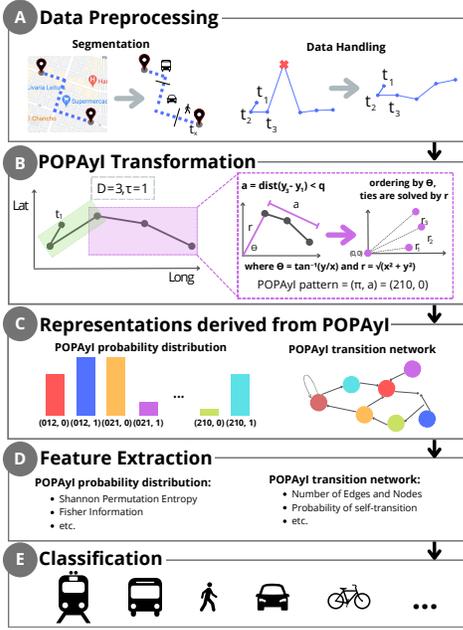


Fig. 2: *POPAYI*'s methodology.

TMD. The components of *POPAYI*, i.e., the polar-like multivariate OP transformation and amplitude information, are detailed in Sections IV-B and IV-C, respectively.

A. Methodology

Figure 2 presents our methodology, described as follows.

(A) Data Preprocessing. This stage aims to transform the raw trajectory into a better format for further analysis, as follows:

- **Segmentation.** To segment the trajectories, we detect stationary points with a minimum standing time of 20 minutes (i.e., the interval between two consecutive points is greater than 20 minutes) or the end of the trajectory.
- **Data Handling.** Data sampling is commonly influenced by various context conditions (e.g., weather), leading to inaccurate measurements that affect TMD results. To prevent this issue, we remove coordinate points with out-of-range values and discard trajectories with fewer than 10 points to avoid generating low-quality traces.

(B) *POPAYI* Transformation. OP extracts a symbolic pattern from a sliding window of a size determined by the parameters D and τ . For instance, in Figure 2, we use $D = 3$ and $\tau = 1$, indicating a sliding window with 3 coordinate points and a time interval of 1 (details in Figure 1). Generalizing OP to 2D, in each sliding window, we order the points based on their polar angle (cf. Section IV-B) and calculate the amplitude by the distance from the first to last point (cf. Section IV-C).

Algorithm 1 presents the pseudo-code for the *POPAYI* transformation. Lines 2 to 13 iterate through all segment points and perform the following operations: First, we sample a sliding window from the segment that will undergo the *POPAYI* transformation, with cost $O(1)$. Lines 3 to 5 check if the segment contains only one dimension, indicating a 1D time series. If so, a new dimension with zero values is added, transforming

it from $\mathbf{S}(t) = \{x_1, \dots, x_n\}$ to $\mathbf{S}(t) = \{(0, x_1), \dots, (0, x_n)\}$. It allows us to use the *POPAYI* transformation for both 1D and 2D data due to *POP* characteristics, detailed in Section IV-B. The time complexity to perform *POP* transformation in line 6 is $O(D \log D)$, but since D is typically small, it can be treated as $O(1)$. Finally, the amplitude variation is calculated from lines 7 to 11, with a time complexity of $O(1)$. This step is explained in detail in Section IV-C. Therefore, the overall time complexity of the *POPAYI* transformation is $O(n)$.

Algorithm 1 *POPAYI* Transformation

```

1: procedure POPAYI_TRANSFORMATION(segment, D, τ, q)
2:   for each sliding window in segment do
3:     if sliding window contains only one dimension then
4:       add new dimension with zero values (each x is now (0, x))
5:     end if
6:     πm ← apply POP on 2D sliding window
7:     if distance from first to last point in sliding window ≥ q then
8:       am ← 1
9:     else
10:      am ← 0
11:    end if
12:    add the tuple (πm, am) to a list
13:  end for
14: return list containing all tuples (πm, am) that represent the segment
15: end procedure

```

(C) Representations derived from *POPAYI*. Although *POPAYI* symbols are tuples, it is possible to extract the exact representations described in Sections II-B1 and II-B2, namely the OP's *probability distributions and transition network*. At the first representation, we count the number of occurrences of each tuple composed of POP and amplitude. At the latter, the nodes in the transition network are the tuples, and the edges are the transition between patterns that appeared sequentially.

(D) Feature Extraction. From each representation, we extract different features. For *probability distribution*, the features are:

- **Shannon Permutation Entropy (PE):** This is a variation from Shannon's classical entropy and quantifies the probability distribution's randomness. Hence, its maximum occurs when all possible permutations of $D!$ have the same probability of occurring (i.e., a uniform distribution), indicating a completely random time series. In contrast, low PE values represent a deterministic time series [16].
- **Statistical Complexity (SC):** This feature computes the degree of regularity present in time series by comparing the difference between the analyzed probability and uniform distribution. Hence, SC captures the relationship between dynamical components (such as determinism and randomness) while measuring their disequilibrium [16, 37].
- **Fisher Information (FI):** It measures the amount of information one observation carries about an unknown parameter, e.g., the probability of observing multiple occurrences of patterns in a trajectory. In other words, FI captures how dispersed are the distribution values, usually evidenced by the curve's shape, wideness, or skewness. Hence, the higher the FI, the sharply peaked will be the curve describing the distribution values and easy to find "real" representative patterns. Contrarily to the population-like focus of PE, FI presents a locality property since it reflects the differences among consecutive probabilities of distributions [37].

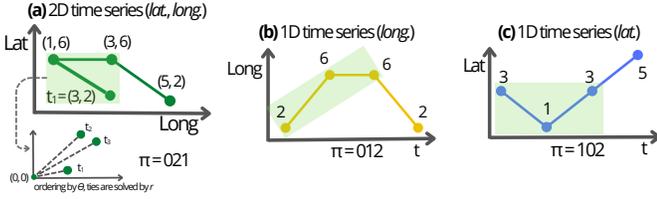


Fig. 3: POP compared to standard 1D OP transformation

The extracted *OP transition network's* features are:

- **Avg. and Standard Deviation of Edge Weights:** transition probabilities' central tendency and dispersion, respectively.
- **Probability of self-transition:** Self-transitions, or loops, are the edges from a vertex to itself, meaning the consecutive occurrences of a pattern. Borges et al. [17] showed that it is a valuable indicator of time series' main characteristics.
- **Number of Edges:** It is the density of a graph in edge connectivity, i.e., the ratio between the existing edges and the maximum number of edges a graph can contain. It can be a vital indicator of temporal dynamics. For instance, as the randomness of a process increases, there are more chances for all possible transitions to occur. Hence, a high number of edge values means more randomness in the system, while low values occur in more deterministic time series.
- **Number of Nodes:** Previous studies have shown that deterministic time series (regardless of size) may have forbidden ordinal patterns. Consequently, their graphs have forbidden nodes. In contrast, stochastic time series contain all possible patterns (and possible nodes) if long enough [17]. Therefore, we measure the node density (using the maximum number of patterns as the upper bound of nodes).

In summary, we consider 11 features: 3 extracted from *POPAyI* probability distribution and 8 extracted from the transition network (i.e., PE, SC, FI, average and standard deviation of edge weights, probability of self-transition, number of edges and nodes). In our experiments, these features show promising results in discriminating transports, but still are not enough to reach state-of-the-art discrimination. For this reason, we go further by muscling *POPAyI* with statistical metrics (i.e., mean, variance, maximum, and minimum) extracted from both motion-related features, i.e., distance and speed. Hence, a total of 19 features are considered in our classifier.

(E) Classification. Next, the extracted features are classified using the Extreme Gradient Boosting (XGBoost), a tree-based traditional ML approach, which achieved the best result in our experiments, as shown in Section V-D5.

B. Polar Ordinal Patterns: POP

Extending standard 1D OP to higher dimensional time series poses challenges in preserving its interpretability while maintaining the relationship among points (cf. Section II). In this context, *POPAyI* introduces *POP (Polar Ordinal Patterns)*, which uses polar coordinates to extract ordinal patterns from 2D data points while retaining their meaningful relationship, effectively addressing previous methods' limitations and enabling the capture of mobility dynamics.

1) POP definition: POP is an adaptation of the Graham scan algorithm [38], designed for planar convex hull computation. It uses polar coordinates to determine the collinearity and relative orientation of points to a reference point. In POP, we leverage this Graham scan's fundamental concept but modify it to calculate the ordinal relationship between points. This adaptation enriches Graham's analysis by incorporating ordinal ordering in addition to collinearity, clockwise, and counterclockwise orientation.

Algorithm 2 presents the POP transformation. First, we transform from geodesic (lat, long) to polar coordinates. To this, we utilize the equations $r = \sqrt{(\text{lat} - a)^2 + (\text{long} - b)^2}$ and $\theta = \tan^{-1}((\text{long} - b)/(\text{lat} - a))$, where (a, b) denotes the reference point. Since we use $(0, 0)$ as our reference point, we can apply the equations as shown in lines 3 and 4.

To obtain the angle θ , we use the `atan2` function, which returns results in the range $-\pi$ to π . However, we must adjust when the calculated angle is less than 0 since negative angles can introduce ambiguity when ordering points based on their polar angles. Thus, in lines 5 to 7 of the algorithm, we add 2π to the angle when negative, maintaining ordering consistency by using only positive angles.

Once each geodesic coordinate is transformed into polar coordinates, we calculate in line 9 the indices that would sort the array based on θ , providing the ordinal relationship between the points. In tie cases, i.e., when points have the same θ , we use the distance r to order. Sorting the points based on their polar angles allows us to preserve the relative order of the points, enabling subsequent analysis and interpretation.

Algorithm 2 POP Transformation

```

1: procedure POP(2D sliding window)
2:   for  $x, y$  in 2D sliding window do
3:      $r \leftarrow \sqrt{x^2 + y^2}$ 
4:      $\theta \leftarrow \tan^{-1}(y/x)$ 
5:     if  $\theta < 0$  then
6:        $\theta \leftarrow 2\pi + \theta$ 
7:     end if
8:   end for
9:    $\pi_m \leftarrow$  ordering by  $\theta$ , ties are solved by  $r$ 
10: return  $\pi_m$ 
11: end procedure

```

For instance, as illustrated in Figure 3, for $D = 3$, $\pi_m = 021$ signify a turn in a specific direction or along a particular route, which is captured by POP. Relying solely on a single 1D coordinate (lat. or long.) limits our ability to extract precise motion details, obtaining only linear behavior, such as north/south and back/forward.

Moreover, the patterns $\pi_m = 012$ and $\pi_m = 210$ represent collinear points. Considering the ordinal ordering, we can distinguish between them based on the sequential relationship of the points, indicating increasing or decreasing trends. Similarly, $\pi_m = 021$ (illustrated in Figure 3) and $\pi_m = 120$ indicate clockwise orientation, while $\pi_m = 201$ and $\pi_m = 102$ (also in Figure 3) signify counterclockwise orientation.

Hence, by introducing ordinal ordering in Graham scan's idea, POP enhances the interpretability and analysis of mobility patterns in 2D time series data, acquiring essential aspects such as the direction of movement and changes in

intensity along the direction. Therefore, POP brings several advantages while inheriting OP benefits (shown in Sec. IV-D) and avoiding the limitations of linear projections that fail to capture mobility points relationships, e.g., turns made in travel.

Finally, we claim that POP is a suitable generalization of the standard OP transformation to 2D data. To this, consider a 1D sliding window $s_m = \{3, 1, 3\}$. Since we cannot directly extract polar coordinates from it, we employ POP as follows: **(1)** Each univariate point y is converted to a 2D point $(0, y)$, as outlined in Algorithm 1; in our example, $s_m = \{(0, 3), (0, 1), (0, 3)\}$. **(2)** The polar angle θ is calculated, yielding the same value for all points (all of them are $\theta = \tan^{-1}(0)$). The distance r (i.e., the original y) resolves these ties. By consistently preserving the values and ordinal relationships from the 1D to the 2D representation, *POP effectively captures the essential characteristics and ordinal relationships of the 1D and 2D time series*. This capability allows for the robust analysis of mobility data.

C. Amplitude-enhanced ordinal patterns

While speed and amplitude provide similar information in equally-spaced trajectory samples, real-world data often presents complex and irregular sampling patterns. To tackle this challenge, *POPAYI* introduces a mechanism to calculate amplitude, capturing the displacement of entities over specific timeframes and encompassing variations in movement speed and direction. By incorporating amplitude, *POPAYI* facilitates more comprehensive analysis, particularly in scenarios with non-uniform sampling, ultimately enhancing accuracy and enabling a deeper understanding of the underlying dynamics.

We capture the amplitude by the distance between the first and last geodesic coordinates within each sliding window and then binarize it using a user-defined threshold value q , as shown in Algorithm 1. If the Euclidean distance between the window's last and first points exceeds q , $a_m = 1$; otherwise, $a_m = 0$. This way, in *POPAYI*, the time series is transformed into a set of patterns with their corresponding amplitudes, $\Pi = \{(\pi_1, a_1), \dots, (\pi_m, a_m)\}$, where each (π_m, a_m) represents a combination of the possible permutation set of $D!$.

For example, consider the points in Figure 3 within the first sliding window $s_1 = \{(3, 2), (1, 6), (3, 6)\}$ and using an amplitude threshold of $q = 5$. In this case, the Euclidean distance is 4 and the resulting pattern is $\pi_1 = (021, 0)$. Now, if we consider another sliding window $s_m = \{(3, 2), (1, 6), (9, 6)\}$ with the same amplitude threshold, it leads to $\text{dist} = 7.2$ and the pattern $\pi_m = (021, 1)$. This example demonstrates how incorporating amplitude information in *POPAYI* allows for distinctive patterns with different amplitudes, providing more nuanced insights into mobility dynamics.

One could expand this approach to more amplitude values, but this would cost an increased complexity and the need for additional parameters to determine amplitude thresholds. Instead, in this work, we adopt binarization to compare amplitude values and prevent the occurrence of infinite or massive amplitude distributions. Furthermore, binarization makes amplitude more reliable regarding outliers since a pattern in *POPAYI* with an abnormal value will not appear much, thus having minor importance in the corresponding distribution.

Using *POPAYI*, we extract features and representations from tuples (π_m, a_m) as symbol patterns, preserving the linear time complexity of the standard OP transformation and slightly increasing space complexity to $O(2D!)$.

D. POPAYI advantages to TMD

POPAYI, as the representations derived from it, inherits the benefits of the OP transformation and introduces several advantages for TMD. These include:

- **Lightweight approach.** *POPAYI* transformation depends on the time series size n and the embedding dimension D , while the construction of its derived transformations involves counting the number of patterns and transitions in $m - 1$ steps. It results in a time complexity of $O(nD \log D)$ with efficient sorting algorithms (e.g., merge sort) or $O(nD^2)$ with simpler algorithms (e.g., selection sort). However, as D is typically small and constant, the time complexity of *POPAYI* is $O(n)$ [15, 17], which is particularly suitable for resource-limited scenarios.
- **Scalability.** *POPAYI* is computationally efficient, with linear time complexity that enables its application to large time series. Also, its derived transformations are independent of the time series size, needing a space of $2D!$ ($3 \leq D \leq 7$).
- **Resilience to trajectory size.** Unlike DL literature [2, 5, 6], OP has the advantage of being resilient to trajectories of different sizes, ranging from small to large. Hence, it adapts to the person's real life, in which mobility behavior can describe trajectories of different trip sizes.
- **Robustness.** *POPAYI* is robust to observational and dynamic noise (e.g., GPS measurement errors) [15, 16].
- **Enhanced representation.** *POPAYI* captures the rich 2D temporal dynamics of time series, providing a more comprehensive representation than standard OP. By incorporating amplitude information, *POPAYI* enables better discrimination and characterization of different spatiotemporal mobility behaviors, enhancing the accuracy of TMD.

V. EXPERIMENTS

This section shows the validation results of *POPAYI*. We present the datasets used in Section V-A. Section V-B shows how we select the *POPAYI*'s hyperparameters used in the experiments. We analyze the Cologne dataset in Section V-C, demonstrating that amplitude information aid in capturing mobility behavior. In Section V-D, we use the Geolife dataset to evaluate *POPAYI* in TMD tasks with different levels of detection complexity, from distinct to closely associated transports. We compare it to existing methods, evaluating their performance across various metrics, such as accuracy, F1 score, and computational complexity.

A. Datasets

In Section V-C, we use the realistic large-scale **Cologne**¹ dataset describing vehicle trajectories generated by Uppoor et al. [39] with a temporal resolution of one second. It contains

¹<http://kolntrace.project.citi-lab.fr/>

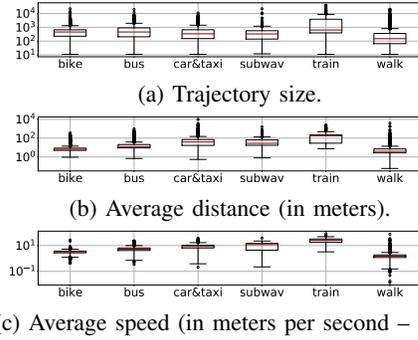


Fig. 4: GeoLife dataset statistics: (a) trajectory size, (b) avg. distance, and (c) avg. speed of each transportation mode.

24 hours weekday traffic of 700,000 individual vehicle trips in a 400 km² region in Köln, Germany. Each dataset’s row informs the timestamp, the vehicle identification, the corresponding two-dimensional position (geodesic coordinates) x and y in meters, and speed (m/s).

Section V-D uses the real-world **GeoLife² dataset** [1] containing 182 users’ trajectories over five years (from April 2007 to August 2012), in which 73 have transportation mode information besides the latitude and longitude. Figure 4 shows boxplots of trajectory size, average distance, and average speed extracted from the transports of these 73 users employed in our evaluation. This extracted set contains the transportation modes required to ensure comparability with related studies.

B. POPAyI’s hyperparameters selection

In our experiments, we tailored the systematic search for the best hyperparameter values to their specific goals. In the first experiment (Sec. V-C), values were determined based on cluster visualization, while in Section V-D, the average F1 score of the training set was considered (as further detailed in Sec. V-D5). This section discusses the explored values.

1) **Amplitude threshold (q):** Setting too high or too low q can hinder displacement differentiation (i.e., vehicles moving similarly but with different speeds), as it results in a single amplitude level, $a_m = 1$ or $a_m = 0$, respectively. Consequently, as *POPAyI* probability distribution expects two distinct amplitude levels and half of the possible permutations are excluded due to an imbalanced amplitude threshold, it could adversely impact the accuracy of our results. In addition, the q unit is the same used in the dataset. In Section V-C, q is measured in meters, while in Section V-D, q is measured in kilometers. Thus, we evaluated different values for each experiment: in the first, we used $q = \{0.5, 1, 2, 3\}$, and in the second, we tested $q = \{0.0005, 0.005, 0.05, 0.1, 0.5, 1, 2, 3\}$.

2) **Embedding dimension (D):** For balance performance and computational efficiency, we consider values of D ranging from 3 to 5, since $D = 6$ requires a larger number of parameters (i.e., 10⁶ and 10⁵ in *POPAyI* and POP, respectively), disrupting our objective of a solution that balances performance and computational efficiency. If presenting similar results, we

selected the smallest D value that yielded the best results, satisfying the condition $n \gg D!$.

3) **Embedding delay (τ):** The number of minimum patterns extractable from a trajectory is given by $m = n - (D - 1)\tau$, where $m > 0$. Consequently, the maximum τ value is limited to $\tau < \frac{n}{D-1}$. Therefore, larger τ values necessitate larger trajectory sizes. For example, with $D = 4$ and $\tau = 5$, $n > 15$. Hence, our experiments explored different values: Section V-C considered $\tau = \{1, 2, 5, 10, 15\}$ since $n > 500$; and in Section V-D, we evaluated $\tau = \{1, 2\}$, considering $n > 10$ (i.e., size before padding, to ensure the extraction of at least one pattern from the original trajectory).

C. Amplitude benefits from POPAyI

Here we aim to examine how amplitude relates to speed gradient and its ability to differentiate vehicles under varying speed-like conditions. The purpose is to showcase the usefulness of amplitude rather than evaluate *POPAyI* performance. Hence, to conduct this experiment in a controlled environment, we use the Cologne dataset (cf. Section V-A), which consists of data exclusively related to one type of transportation.

The dataset encompasses 700,000 trips from about 100,000 distinct vehicles. For this specific experiment, we sought vehicles with diverse traffic characteristics, categorizing them into two groups based on their average speeds: (i) less than five m/s, ranging from 2 m/s to 5 m/s (referred to as the slow group); and (ii) equal to or exceeding five m/s, spanning from 6 m/s to 30 m/s (termed as the fast group). We chose these speed limits based on their travel time: the slow group travels during rush hour (6:25 am to 9 am), while the fast group commutes in the early morning (1 am to 5:30 am). To ensure representative feature extraction that adheres to the condition $n \gg D!$, we specifically opted for trajectories from unique vehicles exceeding 500 points. This criterion led us to randomly select 3000 vehicles (1500 per group), the maximum count meeting our standards in the slow group.

1) **Evaluation Results:** Figure 5 compares results without (Figure 5a) and with (Figure 5b) amplitude. It shows the results of the PE and FI features extracted from the *POPAyI*’s probability distribution. For a better cluster visualization and exclusively for this evaluation, the best parameters configuration is $D = 3$, $\tau = 10$, and $q = 2$, selected as depicted in Section V-B. The τ value indicates the discretization of the dataset in time intervals of 10 seconds, which gives the time distance between points in the OPs, i.e., inside the sliding window. It allows obtaining more representative patterns. Indeed, there are not enough changes in vehicle position for a temporal granularity lower than 10s.

Figure 5a shows that without amplitude is hard to distinguish between vehicles having different mobility behaviors, though presenting evident variances in average speeds. As discussed in Section IV-C, mobility behaviors are reflected in their speed dynamism over time and their spatiotemporal mobility dynamic during a timeframe. This dynamic is set aside if no amplitude information is considered in the ordinal pattern transformation. Hence, as shown in Figure 5b, amplitude information leverages the capture of displacement behaviors, revealing two distinct groups of vehicle behaviors.

²<https://www.microsoft.com/en-us/download/details.aspx?id=52367>

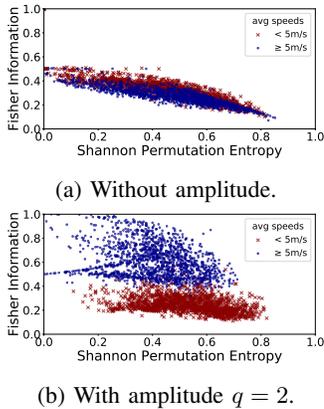


Fig. 5: PE and FI results for $D = 3$, $\tau = 10$, (a) without and (b) with amplitude ($q = 2$). Cologne dataset.

Note that both figures show that the two groups contain vehicles with speed dynamics ranging from random to deterministic occurrences, i.e., from high to low PE. Nevertheless, without the amplitude information (cf. Figure 5a), it is hard to quantify how dispersed are the values of the related probability distribution, usually captured in the FI feature measurement. On the other hand, thanks to the amplitude information (cf. Figure 5b), more information is extracted from vehicle patterns, revealing distinct spatiotemporal differences in mobility behavior, which is reflected in how probability distribution's values of pattern occurrences are scattered. This locality property is captured in the FI results when the amplitude is added to the ordinal pattern transformation.

Those observations are reinforced by Rosso et al. [40], who showed that, in regular processes, normalized PE is close to zero and FI is close to one. On the other hand, stochastic systems have PE close to one and FI close to zero. In this context, FI and PE results in Figure 5 show that adding the amplitude information in OP transformation (i) makes pattern distribution more deterministic (i.e., few concentrated patterns) in the fast group, which culminates in a FI increase; and (ii) increases the randomness of pattern distribution in the slow group, resulting in a stochastic behavior that is harder to predict (high PE) and in a FI decrease. Indeed, 87.27% of vehicles in the fast group present a single pattern responsible for more than half of the time series symbols. Among those 1309 vehicles set, 672 have the dominant pattern as $\pi = (210, 1)$, and 637 have the pattern $\pi = (012, 1)$.

In fact, Zheng et al. [1] discussed in their work that trajectories suffering from congestion or heavy traffic are harder to infer since their temporal dynamics are uncertain, as verified in our current evaluation. In our case, 78.27% of vehicles in the slow group contain nine patterns in their distribution without any concentration. Among these 1174, 1094 vehicles present all possible patterns except for $\pi = (021, 0)$ and $\pi = (102, 0)$, endorsing their increase in random behavior.

2) **POPAyI Transition Network:** Furthermore, the number of patterns in the probability distribution affects the number of edges and nodes from the *POPAyI* transition network. We observe a density of 0.1447 edges (standard deviation of 0.036)

and 0.6461 (0.121) nodes in the fast group. It means that they contain, on average, approximately eight nodes (out of $2D! = 12$ possibilities) and ten edges (out of 66 possibilities). On the contrary, the slow group presents a density of 0.2010 (0.054) edges and 0.7843 (0.135) nodes: about ten nodes and 13 edges on average. Without amplitude, the slow group has a density of edges and nodes of 0.528 (0.121) and 0.957 (0.115), respectively, whereas the fast group contains 0.497 (0.102) and 0.973 (0.101) for density of edges and nodes, respectively. It means five nodes (from six possible) and nine edges on average for both groups. These similar values increase the challenge of identifying the different behaviors needed to precisely detect modes of transportation, which explains the indistinguishability of the two groups in Figure 5a.

Finally, results show the amplitude information's benefits in distinguishing various spatiotemporal mobility behaviors in the challenging context of a unique type of vehicle. Therefore, we claim that amplitude-leveraging of *POPAyI* brings influential detection capability in scenarios counting on heterogeneous transports. The following section demonstrates this claim by comparing *POPAyI* with seven related literature proposals.

D. Transportation mode detection

Different transportation modes exhibit distinct traffic behaviors. Taxis, for example, display more randomness influenced by passenger destinations, while buses follow predictable routes with regular stops. *OP's probability distribution captures and reflects both random and regular behaviors: deterministic behaviors concentrate on specific patterns, while random behaviors encompass a wider range of possible patterns.* Hence, this section demonstrates the effectiveness of features extracted from *POPAyI* transformations (i.e., probability distribution and transition network) in distinguishing traffic behaviors. *These features achieve state-of-the-art results in TMD with fewer parameters than DL approaches.* Evaluation is performed on the GeoLife dataset (Section V-A), which offers diverse transports for a comprehensive assessment of *POPAyI's* performance compared to existing methods.

1) **Evaluation Metrics:** We compared *POPAyI* with the state-of-the-art methods in four different transportation sets using the reported results from each study for the lack of experiment descriptions for replication, hence, some metrics are missing since they did not reported it. We reported accuracy, F1 score, recall, and precision. Accuracy is the fraction of correct predictions. F1-score is a weighted harmonic average between precision (*pre*) and recall (*rec*), defined as $F1 = 2 \times \frac{pre \times rec}{pre + rec}$, where *pre* expresses the proportion of positive predictions that was true positives and *rec* explains the proportion of true positives that were correctly identified.

In addition, we included other interesting metrics, e.g., the trace size needed for training, the number of features, and the number of parameters for each model. These metrics provide insights into the practicality of each method, especially in resource-constrained environments such as IoT and Federated Learning (FL). The ideal method in such scenarios should achieve high scores while using minimal features, parameters, and data size to perform well on limited devices with few

computational resources. Therefore, these metrics are crucial for assessing methods' feasibility in real-world applications.

Tables I, II, and III provide the confidence intervals (at a 95% confidence level), indicated in parentheses. These intervals were computed from the 1000 bootstrap rounds, with stratified classes and sample replacement from the test set.

2) **Data Preprocessing:** We apply specific data cleaning and training selection in this experiment to ensure consistency and comparability with the literature. For fixed-length data segments smaller than a threshold, we apply wrapping padding as Moreau et al. [2], duplicating the segment until the desired length is reached. Data cleaning differ across transportation mode sets. For the **first set** (walk, bike, car&taxi, bus), we remove segments with fewer than 10 data points or merge them if consecutive and belonging to the same transport, as [1]. In the **second set** (walk, bike, car, bus&taxi, subway, train), we remove abnormal fixed-length segments that exceed the maximum values in the average speed distribution: 10 m/s for walking, 25 m/s for bike, 35 m/s for bus and taxi, 35 m/s for car, 55 m/s for train, and 25 m/s for subway. The **third set** (walk, bike, car&taxi, bus, train) and **fourth set** (walk, bike, car&taxi, bus, subway, train) employ thresholds based on speed and acceleration to remove GPS points that exceed these values, as Dabiri and Heaslip [5], respectively: 7 m/s and 3 m/s^2 for walk, 12 m/s and 3 m/s^2 for bike, 34 m/s and 2 m/s^2 for bus, 50 m/s and 10 m/s^2 for car&taxi, and 34 m/s and 3 m/s^2 for train. Moreover, in third set, train refers to all railways-based transports, i.e., train, subway, and railways [5].

3) **Number of Features:** For DL approaches, the number of features corresponds to the input size. Regarding traditional ML methods, Zheng et al. [1] used 13 features related to vehicle motion, e.g., speed change rate and stop rate. Xiao et al. [3] extracted 111 features, including global and local ones. Global features are descriptive statistics for the entire trajectory, such as average and skewness. The local features generated by profile decomposition focus on movement behavior (e.g., percentage of each decomposition class).

POPAYI uses the features described in Section IV-A. We extract 11 features from the probability distribution and transition network and incorporate eight statistical features related to motion: average, standard deviation, minimum, and maximum values of distance and speed time series. Therefore, we use 19 features in *POPAYI* for all transportation modes sets.

4) **Number of Parameters:** For DL approaches, the number of parameters is the number of weights. In *POPAYI*, it is the number of bins in probability distribution and the number of edges in the transition network assuming a complete graph, totaling $2D! + \frac{2D!(2D!-1)}{2}$. Traditional ML methods have too few parameters; thus, we do not show it in Table III.

5) **ML method and hyperparameter selection:** While our primary focus is on extracting features from *POPAYI*, selecting the most suitable ML method is crucial to prevent biases that could impact classification results [41]. Hence, to determine the best *POPAYI* hyperparameters and select the top-performing ML method for our experiment, we employed a simplified version of Successive Halving [42]. First, we randomly divided data into training (70%) and test sets (30%). We evaluated all *POPAYI* hyperparameters in each ML method

using varying percentages of the training set (20%, 40%, and 80%), excluding the least effective classifier (i.e., lower average F1 score) at each step until the optimal ML method was identified. The chosen ML method thoroughly evaluated the entire training set, considering all possible combinations of selected values for D , τ , and q to pinpoint the best hyperparameter combination. All rounds used cross-validation with 5 folds and stratified classes, ensuring that each fold had the same class distribution and, thus, neither of the classes was over-represented, which may lead to increased unrealistic results. This comprehensive approach ensured the robustness of our methodology and optimized experimentation time, especially given the substantial number of hyperparameter combinations involved. For ML methods, we employed Support Vector Machines (SVM) with Radial Basis Function (RBF) kernel, Decision Trees (DT), XGBoost with 300 trees, and Random Forest (RF) with 500 trees. Both tree-based ensembles allow unlimited depth until all nodes/leaves contain fewer than two samples, each subtree using a maximum of four features. The *POPAYI* hyperparameter values are detailed in Section V-B. We used the fourth set since it contains the largest number of transports.

In this process, SVM was the first excluded, with an average F1 score approximately 15% lower than the other methods in all hyperparameter combinations. The second round banned DT, which achieved about 8% less F1 score. In the last round, XGBoost outperformed RF by about 2%.

The figure 6 illustrates the systematic search using XGBoost on the entire training set, revealing that two configurations achieved similar results: $D = 3$, $\tau = 2$, and $q = 0.0005$, and $D = 4$, $\tau = 2$, and $q = 0.05$. Considering the lower number of parameters, we opted for the former.

The unit of q depends on the dataset, with the current experiment using kilometers. Therefore, $q = 0.0005$ represents a threshold of 0.5 meters, providing a low but meaningful value that may capture nuances in traffic dynamics, such as stationary or accelerating/decelerating transports or even traffic congestion for road-based transportation. Additionally, the consistent use of $\tau = 2$ in achieving the best results suggests that trajectories exhibit more representative transport dynamics when considering patterns between non-consecutive points. Yet, the Figure also indicates the difficulty of establishing a straightforward correlation between hyperparameter values.

Table I highlights the importance of choosing an appropriate ML method, as different algorithms yield distinct performance levels. SVM performs poorly, with about 50% lower scores than the best methods. DT improved the scores significantly, suggesting a better fit for the task. RF further enhances performance, achieving higher accuracy and F1 score than SVM and DT. Notably, XGBoost emerges as the top-performing ML method, surpassing all other algorithms in all metrics, especially RF, by about 2% in F1 score, precision, and recall.

6) ***POPAYI's contribution to classification:*** Table II provides valuable insights into the individual contributions of POP and amplitude in TMD for the fourth set, that contains the largest number of transports. We prepared this experiment as described in Section V-D1. Additionally, we evaluated the performance of 1D OP applied separately to latitude

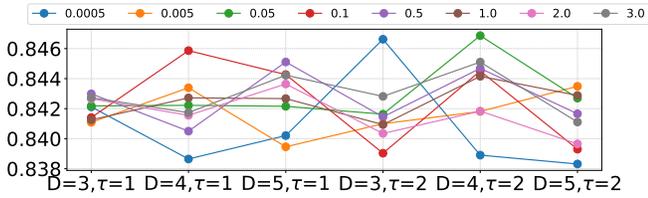


Fig. 6: Systematic search of POPAyI hyperparameters.

TABLE I: Comparison for different ML methods. $D = 3$, $\tau = 2$, and $q = 0.0005$.

	Accuracy	F1 score	Precision	Recall
SVM	37.36% (± 0.02)	15.62% (± 0.03)	20.35% (± 0.02)	33.82% (± 0.32)
DT	78.96% (± 0.03)	74.67% (± 0.05)	74.77% (± 0.05)	74.65% (± 0.05)
RF	86.35% (± 0.03)	82.53% (± 0.04)	81.64% (± 0.04)	83.72% (± 0.04)
XGBoost	87.12% (± 0.03)	84.02% (± 0.04)	83.28% (± 0.04)	85.90% (± 1.71)

and longitude. For this, we extract features of Section IV-A from lat and long separately and applied XGBoost feature importance method to select the top most relevant 19 features, the same number used in *POPAyI*.

The results presented in the table demonstrate significant performance improvements achieved by both POP and *POPAyI* compared to 1D OP. Notably, POP surpasses 1D OP in all metrics by about 3%. It suggests that adopting the polar form effectively captures mobility behavior more accurately than standard OP while preserving the exact parameter count. Additionally, using POP eliminates the need for time-consuming feature selection, expediting the TMD process.

Furthermore, *POPAyI* achieves superior performance compared to POP, about a 2% improvement across all evaluated metrics. Therefore, amplitude information further enhances the accuracy of identifying and distinguishing different patterns of mobility behavior, contributing to a lower false alarm rate and enhancing the detection of positive instances. However, it comes at a higher computational cost since it requires one order of magnitude more in terms of the number of parameters. In resource-constrained scenarios where computational efficiency is a priority, using only POP can still yield satisfactory results, albeit with slightly less power than *POPAyI*.

7) **Comparison Results:** The advantages of traditional ML methods are handling data of many sizes and a small number of parameters, but they need more features to achieve better metrics results. In the **first set**, *POPAyI* surpasses the results reported by Zheng et al. [1] by a significant margin: with only

TABLE II: Results for different D values for *POPAyI* with and without amplitude (i.e., *POP*). Trajectory size of 500 points.

	D	F1 score	Recall	Precision	No. of Params.
<i>POPAyI</i> $q = 0.0005$	3	84.02% (± 0.04)	83.28% (± 0.04)	85.90% (± 1.71)	7.8×10^1
	4	84.04% (± 0.04)	83.40% (± 0.04)	84.81% (± 0.04)	1.2×10^3
	5	83.49% (± 0.04)	82.80% (± 0.04)	84.32% (± 0.04)	2.8×10^4
<i>POP</i>	3	82.24% (± 0.04)	81.23% (± 0.04)	83.47% (± 0.04)	2.1×10^1
	4	82.31% (± 0.03)	81.45% (± 0.04)	83.32% (± 0.04)	3.0×10^2
	5	81.13% (± 0.04)	79.39% (± 0.04)	81.99% (± 0.04)	7.3×10^3
1D OP lat. and long. separately	3	78.96% (± 0.05)	78.49% (± 0.05)	78.04% (± 0.05)	2.1×10^1
	4	78.40% (± 0.05)	77.81% (± 0.05)	79.31% (± 0.05)	3.0×10^2
	5	80.81% (± 0.03)	80.16% (± 0.05)	81.63% (± 0.04)	7.3×10^3

6 more features, *POPAyI* achieves a remarkable improvement of over 10% in all evaluated metrics. In the **second set**, *POPAyI* demonstrates consistency in delivering solid metric results using significantly fewer features than Xiao et al. [3], which extracted 111 features. Despite having five times fewer features, *POPAyI* achieves comparable F1-score and accuracy results, with a difference of around 4%. Moreover, these sets highlight *POPAyI*'s capability to extract meaningful insights from mobility with a compact feature set, outperforming even DL approaches.

In the **third set**, *POPAyI* outperforms both a lightweight approach and an ensemble of CNNs regarding parameter efficiency, making it more suitable for resource-constrained scenarios. Despite using a smaller trace size for training, *POPAyI* achieves comparable classification results with significantly fewer parameters. In addition, compared to only the best CNN in the ensemble, *POPAyI* leverages F1 score and accuracy by about 10%. Furthermore, increasing the trace size enhances the performance of *POPAyI* allowing it to achieve similar metric results to Lu and Xia [6] and James [21] without increasing the number of parameters, which is impossible with DL methods. The ability of *POPAyI* to utilize longer segments provides richer features, enhancing its effectiveness in TMD.

Finally, *POPAyI* has a significant advantage over DL approaches regarding parameter efficiency. It consistently achieves competitive results with 10^3 times fewer parameters compared to the Lightweight CNN method [2]. In the first set, *POPAyI* achieves similar results to the Lightweight CNN with the same trace size for training (500 points). However, as the trained trace size increases, *POPAyI* outperforms the Lightweight CNN in metrics results, still using 10^3 times fewer parameters. This trend continues in the **fourth set** with six different transportation modes, where *POPAyI* surpasses the F1-score of the Lightweight CNN, even when using (i) the same trace size, (ii) 10^3 times fewer parameters, and (iii) reducing the number of features by 96%.

8) **Confusion Matrix:** Figure 7 illustrates the confusion matrix for the fourth set, containing the largest number of transports. The main diagonal represents true positives, showcasing *POPAyI*'s accuracy in correctly identifying transports and capturing the expected differences in temporal dynamics.

The confusion matrix unveils intriguing misclassification patterns, which could be attributed to various hypotheses. Imbalanced classes might significantly impact this scenario, where having more examples could enhance the classifier's ability to identify traffic dynamics and distinguish between different transports. Generating synthetic trajectories could be a potential solution, although it poses challenges.

Moreover, particular transportation modes are more prone to inaccurate predictions when compared to others, potentially due to shared traffic characteristics leading to similar temporal dynamics. Instances include bus and car&taxi, as well as subway and bus. Conversely, the confusion matrix indicates that distinctions between motor and non-motor transports are more evident in their classifications (e.g., walking and biking compared to car&taxi and train) owing to their distinct behaviors in speed and traveled distances.

Additionally, some misclassifications may derive from data

TABLE III: Quantitative comparison. *POPAYI* with $D = 3$, $\tau = 2$, and $q = 0.0005$.

Transportation modes sets	Methods	Reported F1 score	Reported Accuracy	Reported Recall	Reported Precision	Data size training	No. of feats.	No. of params.
walk, bike, car&taxi, bus	DT [1]	74.77%	76.20%	76.37%	76.92%	10-8000	13	-
	<i>POPAYI</i>	86.34% (± 0.04)	90.21% (± 0.05)	86.56% (± 0.05)	86.16% (± 0.06)	10-8000	19	7.8×10^1
	Light. CNN [2]	87.10% (± 1.1)	-	-	-	500	500	1.1×10^4
	<i>POPAYI</i>	86.86% (± 0.04)	89.73% (± 0.03)	86.74% (± 0.04)	87.02% (± 0.04)	500	19	7.8×10^1
walk, bike, car, train bus&taxi, subway	XGBoost [3]	87.40%	90.77%	90.84%	86.46%	10-39120	111	-
	<i>POPAYI</i>	82.40% (± 0.07)	86.55% (± 0.04)	80.67% (± 0.09)	84.57% (± 0.07)	10-39120	19	7.8×10^1
walk, bike, car&taxi, bus, train	Best CNN [5]	74.80%	79.80%	-	-	200	200	2.6×10^6
	7 CNNs [5]	83.90%	84.80%	82.42%	86.30%	200	200	1.8×10^7
	LSTM [21]	91.90%	92.70%	91.84%	92.00%	200	200	8.1×10^6
	AE + CNN [6]	88.28%	89.47%	86.99%	89.85%	200	200	4.1×10^7
	Light. CNN [2]	83.90% (± 1.10)	-	-	-	500	500	1.1×10^4
	<i>POPAYI</i>	85.68% (± 0.03)	87.42% (± 0.03)	85.02% (± 0.03)	86.50% (± 0.03)	500	19	7.8×10^1
	<i>POPAYI</i>	85.77% (± 0.03)	89.48% (± 0.05)	84.77% (± 0.05)	86.99% (± 0.04)	1000	19	7.8×10^1
walk, bike, car&taxi, bus, subway, train	<i>POPAYI</i>	86.71% (± 0.05)	89.76% (± 0.03)	85.92% (± 0.05)	87.61% (± 0.05)	10-17000	19	7.8×10^1
	Light. CNN [2]	81.80% (± 1.10)	-	-	-	500	500	1.1×10^4
	<i>POPAYI</i>	84.02% (± 0.04)	87.12% (± 0.04)	83.28% (± 0.04)	85.90% (± 1.71)	500	19	7.8×10^1
	<i>POPAYI</i>	85.77% (± 0.03)	89.48% (± 0.04)	84.77% (± 0.05)	86.99% (± 0.05)	1000	19	7.8×10^1

limitations, such as subway and walking, which might occur due to a stronger GPS signal at subway stations when the subway stops, creating the impression of walking dynamics (i.e., lower speed).

Addressing these challenges through preprocessing measures such as cleaning, gap filling, and transfer learning could improve classification results. However, such approaches may demand more extensive efforts and could conflict with our goal of maintaining a lightweight approach.

Finally, the impact of false negatives and positives varies across Intelligent Transportation Systems (ITS) applications. For instance, in urban planning studies, combining walking and biking classifications might be acceptable, as both modes share similarities in not following traffic directions strictly. Mixing buses and car&taxi might also be adequate in this scenario. However, safety-critical applications demand minimal false negatives. Failing to detect pedestrians and bicycles among autonomous vehicles could lead to accidents, compromising safety. In such cases, minimizing false negatives between motorized and non-motorized vehicles is essential.

For *POPAYI*, we observe that false negatives between motorized and non-motorized are less frequent, which is advantageous for safety-critical applications. Moreover, it contains fewer false positives, providing precise and reliable detection. This is essential in applications seeking to avoid unnecessary interventions, improve efficiency, and enhance user experiences, such as traffic management and planning.

9) **Latency:** We used a machine with the following configuration: Ubuntu 20.04 OS, 12 \times Intel® Core™ i7-10750H CPU @ 2.60GHz, and 15 GB RAM. In this context, *POPAYI* takes about 40 seconds to train and 0.01 seconds to classify segments containing 1000 points, and respectively, 20 seconds and 0.005 seconds for the ones with 500 points. Because of the lack of publicly available reproducible implementation of the other methods, we could not reassess their time complexity. However, as shown in the last column in Table III, for different scenarios, *POPAYI* needs fewer parameters than DL methods (from 3 to 6 orders of magnitude) but still achieves competitive results. Traditional ML strategies have the disadvantage of

Ground Truth	walk	98.7	1.1	0.0	0.0	0.2	0.0
	bike	6.9	90.5	2.3	0.3	0.0	0.0
	bus	2.3	3.4	80.2	10.8	2.0	1.4
	taxi&car	1.3	1.1	19.5	73.1	2.9	2.0
	subway	7.4	1.6	12.6	5.9	68.5	4.0
	train	0.3	0.7	3.0	4.3	3.0	88.6
			walk	bike	bus	taxi&car	subway
		Predicted					

Fig. 7: Confusion matrix using XGBoost with 300 trees, for $D = 3$, $\tau = 2$, $q = 0.0005$. Trajectory size of 500 points.

needing more features than *POPAYI* to classify, which is a burdensome step that requires more effort to extract them.

VI. CONCLUSION AND FUTURE DIRECTIONS

We introduced *POPAYI*, a novel TMD approach based on the Ordinal Pattern (OP) transformation applied to mobility time series. *POPAYI* uses polar coordinates in the OP extraction, enabling multivariate analysis while preserving the natural non-linear mobility aspects. It also the first work to incorporate amplitude information in a multivariate OP transformation.

Experimental results on two popular mobility datasets showed that *POPAYI* achieves excellent performance while balancing accuracy, complexity, and the number of features used. Compared to traditional ML methods, *POPAYI* achieved similar results with significantly fewer features, reducing the feature count by approximately 90 while maintaining comparable performance. Similarly, *POPAYI* outperformed DL approaches while using 1000 to 10000 times fewer parameters. ***POPAYI* applicability:** With its ability to handle diverse trip sizes and its robustness against observational and dynamic noise, *POPAYI* is a highly suitable choice for real-world transportation scenarios characterized by the complex mobility behavior of individuals, such urban management and planning. Its lightweight and efficient design makes it particularly well-suited for resource-constrained environments

such as ITS and Smart City applications. By processing data locally on edge devices, *POPAYI* minimizes data transmission and communication costs, making it ideal for energy-efficient IoT transportation systems by optimizing resource usage. The integration with edge computing enables adaptive decision-making, dynamic resource allocation, and low-latency local processing, ensuring scalability and responsiveness for ITS solutions. Furthermore, when combined with Federated Learning (FL) frameworks, *POPAYI* empowers the development of personalized and privacy-preserving services, making it valuable in various B2C (business-to-consumer) FL applications, including personalized healthcare and virtual personal assistants.

Transfer *POPAYI*'s learning: We recognize the need to enhance *POPAYI*'s generalizability by validating it with mobility data from various cities. We also plan to explore transfer learning techniques to adapt models to cities with distinct transportation characteristics. Additionally, we aim to investigate strategies that can reduce the need for manual labeling of trajectories, such as semi-supervised and weak-supervised learning, as well as FL, which preserve privacy while aggregating information from multiple devices and can achieve state-of-the-art performance with limited labeled data.

OP more amplitude levels: We intend to incorporate more amplitude levels into *POPAYI* to further enhance the distinction between transports by capturing a broader range of speed dynamics. Hence, we aim to investigate the tradeoff between amplitude levels and the algorithm's complexity to find an optimal balance that maximizes the discriminative power of the OP transformation while maintaining computational efficiency.

Data sampling resolution: In our future work, we aim to enhance *POPAYI* by adapting the parameter τ to appropriate data sampling. This approach will address several challenges associated with determining a representative temporal sampling, including the curse of dimensionality, the risk of overlooking important information, increased preprocessing time, and the occurrence of rare patterns in OP transformations. Therefore, this strategy can help enhance the robustness and effectiveness of the OP transformation, leading to more accurate feature extraction and improved classification results.

Leveraging laws in human mobility behavior: For instance, while the literature often treats private cars and taxis as the same mode, their behaviors differ significantly. Taxis follow routes determined by passenger pick-ups, being mainly random origin-destination patterns. In contrast, private cars follow human mobility laws influenced by drivers' daily circadian habits. Thus, we plan to incorporate new features based on human mobility laws (e.g., few important places, and itineraries adapted to traffic conditions) as well as scheduled transportation properties (e.g., fixed and unchangeable itineraries and highly regular stay points) [43]. We anticipate these features will improve accuracy and robustness in TMD.

ACKNOWLEDGMENT

This work was partially funded by grant APQ-00426-22, Fundação de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG), grant 2023/00721-1, São Paulo Research Foun-

dation (FAPESP), and grant 312682/2021-2, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), INRIA and STIC AmSud LINT (code 22-STIC-07).

REFERENCES

- [1] Y. Zheng, Q. Li, Y. Chen, X. Xie, and W.-Y. Ma, "Understanding mobility based on GPS data," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 312–321.
- [2] H. Moreau, A. Vassilev, and L. Chen, "The devil is in the details: An efficient convolutional neural network for transport mode detection," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [3] Z. Xiao, Y. Wang, K. Fu, and F. Wu, "Identifying different transportation modes from trajectory data using tree-based ensemble classifiers," *ISPRS International Journal of Geo-Information*, vol. 6, no. 2, p. 57, 2017.
- [4] M. Guo, S. Liang, L. Zhao, and P. Wang, "Transportation mode recognition with deep forest based on GPS data," *IEEE Access*, vol. 8, pp. 150 891–150 901, 2020.
- [5] S. Dabiri and K. Heaslip, "Inferring transportation modes from GPS trajectories using a convolutional neural network," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 360–371, 2018.
- [6] S. Lu and Y. Xia, "Dual supervised autoencoder based trajectory classification using enhanced spatio-temporal information," *IEEE Access*, vol. 8, pp. 173 918–173 932, 2020.
- [7] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.
- [8] M. Mohr, F. Wilhelm, M. Hartwig, R. Möller, and K. Keller, "New approaches in ordinal pattern representations for multivariate time series," in *The Thirty-Third International Flairs Conference*, 2020.
- [9] Y. Rayan, Y. Mohammad, and S. A. Ali, "Multidimensional permutation entropy for constrained motif discovery," in *Asian Conference on Intelligent Information and Database Systems*. Springer, 2019, pp. 231–243.
- [10] J. Zhang, J. Zhou, M. Tang, H. Guo, M. Small, and Y. Zou, "Constructing ordinal partition transition networks from multivariate time series," *Scientific reports*, vol. 7, no. 1, pp. 1–13, 2017.
- [11] I. Cardoso-Pereira, J. B. Borges, P. H. Barros, A. F. Loureiro, O. A. Rosso, and H. S. Ramos, "Leveraging the self-transition probability of ordinal patterns transition network for transportation mode identification based on GPS data," *Nonlinear Dynamics*, vol. 107, no. 1, pp. 889–908, 2022.
- [12] B. Fadlallah, B. Chen, A. Keil, and J. Príncipe, "Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information," *Physical Review E*, vol. 87, no. 2, p. 022911, 2013.
- [13] H. Azami and J. Escudero, "Amplitude-aware permutation entropy: Illustration in spike detection and signal segmentation," *Computer methods and programs in biomedicine*, vol. 128, pp. 40–51, 2016.
- [14] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical review letters*, vol. 88, no. 17, p. 174102, 2002.
- [15] A. L. Aquino, T. S. Cavalcante, E. S. Almeida, A. C. Frery, and O. A. Rosso, "Characterization of vehicle behavior with information theory," *The European Physical Journal B*, vol. 88, no. 10, pp. 1–12, 2015.
- [16] O. A. Rosso, H. A. Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes, "Distinguishing noise from chaos," *Physical review letters*, vol. 99, no. 15, p. 154102, 2007.
- [17] J. B. Borges, H. S. Ramos, R. A. Mini, O. A. Rosso, A. C. Frery, and A. A. Loureiro, "Learning and distinguishing time series dynamics via ordinal patterns transition graphs," *Applied Mathematics and Computation*, vol. 362, p. 124554, 2019.
- [18] H. Huang, Y. Cheng, and R. Weibel, "Transport mode detection based on mobile phone network data: A systematic review," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 297–312, 2019.
- [19] P. Sadeghian, J. Håkansson, and X. Zhao, "Review and evaluation of methods in transport mode detection based on GPS tracking data," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 8, no. 4, pp. 467–482, 2021.
- [20] L. Li, J. Zhu, H. Zhang, H. Tan, B. Du, and B. Ran, "Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using GPS data," *Transportation Research Part A: Policy and Practice*, vol. 136, pp. 282–292, 2020.
- [21] J. James, "Travel mode identification with GPS trajectories using wavelet transform and deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1093–1103, 2020.
- [22] J. Kim, J. H. Kim, and G. Lee, "GPS data-based mobility mode inference model using long-term recurrent convolutional networks,"

Transportation Research Part C: Emerging Technologies, vol. 135, p. 103523, 2022.

- [23] R. Li, Z. Yang, X. Pei, Y. Yue, S. Jia, C. Han, and Z. He, "A novel one-stage approach for pointwise transportation mode identification inspired by point cloud processing," *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104127, 2023.
- [24] J. James, "Semi-supervised deep ensemble learning for travel mode identification," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 120–135, 2020.
- [25] P. Vinayaraj and K. Mede, "Multi-branch deep learning based transport mode detection using weakly supervised labels," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, pp. 525–530, 2022.
- [26] Z. Li, G. Xiong, Z. Wei, Y. Lv, N. Anwar, and F.-Y. Wang, "A semisupervised end-to-end framework for transportation mode detection by using gps-enabled sensing devices," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7842–7852, 2021.
- [27] Y. Endo, H. Toda, K. Nishida, and A. Kawanobe, "Deep feature extraction from trajectories for transportation mode estimation," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Auckland, New Zealand: Springer, 2016, pp. 54–66.
- [28] R. Zhang, P. Xie, C. Wang, G. Liu, and S. Wan, "Classifying transportation mode and speed from trajectory data via deep multi-scale learning," *Computer Networks*, vol. 162, p. 106861, 2019.
- [29] G. Jiang, S.-K. Lam, P. He, C. Ou, and D. Ai, "A multi-scale attributes attention model for transport mode identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 152–164, 2020.
- [30] J. Zeng, Y. Yu, Y. Chen, D. Yang, L. Zhang, and D. Wang, "Trajectory-as-a-sequence: A novel travel mode identification framework," *Transportation Research Part C: Emerging Technologies*, vol. 146, p. 103957, 2023.
- [31] Y. Ma, X. Guan, J. Cao, and H. Wu, "A multi-stage fusion network for transportation mode identification with varied scale representation of GPS trajectories," *Transportation Research Part C: Emerging Technologies*, vol. 150, p. 104088, 2023.
- [32] L. Zhang, L. Liu, S. Bao, M. Qiang, and X. Zou, "Transportation mode detection based on permutation entropy and extreme learning machine," *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [33] K. Keller and H. Lauffer, "Symbolic analysis of high-dimensional time series," *Intl. Journal of Bifurcation and Chaos*, vol. 13, no. 09, pp. 2657–2668, 2003.
- [34] D. Cuesta Frau, "Permutation entropy: Influence of amplitude information on time series classification performance," *Mathematical Biosciences and Engineering*, vol. 16, no. 6, pp. 6842–6857, 2019.
- [35] J. Xia, P. Shang, J. Wang, and W. Shi, "Permutation and weighted-permutation entropy analysis for the complexity of nonlinear time series," *Communications in Nonlinear Science and Numerical Simulation*, vol. 31, no. 1-3, pp. 60–68, 2016.
- [36] X. Sun, M. Small, Y. Zhao, and X. Xue, "Characterizing system dynamics with a weighted and directed network constructed from time series data," *Chaos*, vol. 24, no. 2, p. 024402, 2014.
- [37] J. B. Borges, H. S. Ramos, and A. F. Loureiro, "A classification strategy for Internet of Things data based on the class separability analysis of time series dynamics," *ACM Trans. on IoT*, vol. 37, no. 4, 2022.
- [38] X. Kong, H. Everett, and G. Toussaint, "The graham scan triangulates simple polygons," *Pattern Recognition Letters*, vol. 11, no. 11, pp. 713–716, 1990.
- [39] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, 2013.
- [40] O. A. Rosso, F. Olivares, and A. Plastino, "Noise versus chaos in a causal fisher-shannon plane," *Papers in physics*, vol. 7, no. 1, pp. 0–0, 2015.
- [41] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [42] K. Jamieson and A. Talwalkar, "Non-stochastic best arm identification and hyperparameter optimization," in *Artificial intelligence and statistics*. PMLR, 2016, pp. 240–248.
- [43] E. Mucelli Rezende Oliveira, A. Carneiro Viana, C. Sarraute, J. Brea, and I. Alvarez-Hamelin, "On the regularity of human mobility," *Pervasive and Mobile Computing*, vol. 33, pp. 73–90, 2016.



Isadora Cardoso-Pereira holds a BSc in Computer Engineering from Universidade Federal de Alagoas (UFAL), Brazil, and an MSc in Computer Science from Universidade Federal de Minas Gerais (UFMG), Brazil, in 2018 and 2020, respectively. She is currently a Ph.D. candidate in Computer Science at UFMG, with research interests including Data Science, Artificial Intelligence, Internet of Things, and Urban Computing.



João B. Borges obtained his BSc in Computer Science from Universidade Estadual do Rio Grande do Norte, Brazil, in 2006, his MSc in teleinformatics from Universidade Federal do Ceará, Brazil, in 2009, and his Ph.D. in Computer Science from UFMG in 2021. He is presently a professor in the Department of Computing and Technology at Universidade Federal do Rio Grande do Norte, Brazil. His research interests include computer networks, distributed systems, mobile and ubiquitous computing, Internet of Things, time series, and data analysis.



Aline C. Viana received her BSc in Computer Science from the Universidade Federal de Goiás (UFG), Brazil, in 1998, her MSc in Electrical Engineering from UFG and Universidade Federal do Rio de Janeiro, Brazil, in 2001, and her Ph.D. in Computer Science from Université Pierre et Marie Curie, France, in 2005. She is currently a Research Director at INRIA and heads the TRiBE research team. Her research interests include computer networks focusing on mobile network, human behavior modeling, Internet of Everything, and smart cities.



Technical Achievement Award of the Brazilian Computer Society.

Antonio A. F. Loureiro (Member, IEEE) holds a BSc and MSc in Computer Science from UFMG and a Ph.D. in Computer Science from the University of British Columbia, Canada. He is a full professor at UFMG, leading the research group on mobile ad hoc networks. He has regularly published in international conferences and journals and presented keynotes and tutorials at international conferences. He was awarded the 2015 IEEE Ad Hoc and Sensor (AHSN) Technical Achievement Award and the Computer Networks and Distributed Systems Interest Group



Heitor S. Ramos (Senior Member, IEEE) received his BSc in Electrical Engineering from the Universidade Federal da Paraíba, Brazil, in 1992, his MSc in Computing Modeling from UFAL in 2004, and his Ph.D. in Computer Science from UFMG in 2012. He is currently an Associate Professor at the Department of Computer Science, UFMG, DCC/UFMG, Brazil. His research interests include data analysis for the Internet of Things, sensor networks, social networks, and urban computing applications.