

# Leveraging the Self-Transition Probability of Ordinal Patterns Transition Network for Transportation Mode Identification Based on GPS Data

Isadora Cardoso P. Silva (✉ [isadoracardoso@dcc.ufmg.br](mailto:isadoracardoso@dcc.ufmg.br))

Federal University of Minas Gerais: Universidade Federal de Minas Gerais <https://orcid.org/0000-0002-7681-7653>

Joao B. Borges

Federal University of Rio Grande do Norte: Universidade Federal do Rio Grande do Norte

Pedro H. Barros

Federal University of Minas Gerais: Universidade Federal de Minas Gerais

Antonio F. Loureiro

Federal University of Minas Gerais: Universidade Federal de Minas Gerais

Oswaldo A. Rosso

Federal University of Alagoas: Universidade Federal de Alagoas

Heitor S. Ramos

Federal University of Minas Gerais: Universidade Federal de Minas Gerais

---

## Research Article

**Keywords:** Ordinal Patterns, Ordinal Patterns Transition Network, Time Series, Information Theory, Mobility

**Posted Date:** April 6th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-348558/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# Leveraging the Self-Transition Probability of Ordinal Patterns Transition Network for Transportation Mode Identification Based on GPS Data

Isadora Cardoso-Pereira · João B. Borges · Pedro H. Barros · Antonio F. Loureiro · Osvaldo A. Rosso · Heitor S. Ramos

Received: date / Accepted: date

**Abstract** Analyzing people mobility and identifying the transportation mode used by them is essential for cities that want to reduce traffic jams and travel time between their points, thus helping to improve the quality of life of citizens. Mining this type of data, however, faces several complexities due to its unique properties. In this work, we propose the use of Information Theory quantifiers retained from the Ordinal Patterns (OP) transformation, for transportation mode identification. As an initial exploration, our results show that OP satisfactorily characterizes the trajectories. Moreover, in this scenario, the characteristics of OP transformation can be advantageous, such as its simplicity, robustness, and speed.

**Keywords** Ordinal Patterns · Ordinal Patterns Transition Network · Time Series · Information Theory · Mobility

## 1 Introduction

The analysis of GPS data is a well-studied problem that has been used to understand the mobility of different moving entities, as well as the environment they live, such as people [30], vehicles [1], animals [10], and meteorological events [23]. In particular, the analysis of people's mobility and the identification of their transportation mode are essential activities for cities that seek a

---

Isadora Cardoso-Pereira, Pedro H. Barros, Antonio F. Loureiro, Heitor S. Ramos  
Federal University of Minas Gerais, Belo Horizonte, Brazil.  
E-mail: {isadoracardoso, pedro.barros, loureiro, ramosh}@dcc.ufmg.br

João B. Borges  
Federal University of Rio Grande do Norte, Natal, Brazil.  
E-mail: joaoborges@dct.ufrn.br

Osvaldo A. Rosso  
Federal University of Alagoas, Maceió, Brazil.  
E-mail: oarosso@laccan.ufal.br

sustainable development, by providing solutions that can help improving the life quality of their citizens, such as reducing traffic jam and travel time.

Regarding its interdisciplinary and broad scope of real-world applications, it is evident the need of identifying transportation mode from trajectory data, with several studies analyzing many approaches in this field. For instance, [8] and [15] use GPS and wearable sensors data (e.g. body temperature and heart rate) to detect activities such as walking, rowing, and cycling. This kind of approach, however, obliges the user to carry several sensors in order to have their transportation mode information detected. Moreover, some works show that it is possible to combine sensor data with external information. As examples, we can cite the work of [20], in which the authors classify different forms of motorized transport such as car, bus, and subway, using GPS data, transit route information published by transit agencies, and motion detection collected from phone's accelerometer. The disadvantage of this kind of approach is the need of models capable of collecting the external information every other time, since city information can change over time. Therefore, in this work, we concentrate our efforts in classification of trajectories based on GPS data.

Mining trajectory data to extract valuable information, however, faces several complexities, being a challenging problem due to its unique properties. Besides high dimensionality, heterogeneity and noise, well-known problems in the big data era, trajectories, being a time series data, depend on the ordering, and, thus, a change in the order could change their meaning. It opposes the common assumption made by many algorithms, such as Naïve Bayes, of independent and identically distributed observations, leading standard classification methods to perform poorly [3].

In this context, the problem addressed in this work can be stated as follows:

*Given a trajectory extracted from GPS data, is it possible to obtain an useful characterization that can help the identification of the transportation modes used?*

Several characterizations has been studied in order to accomplish this goal. For instance, many works extract a large amount of features [32],[31], reaching hundreds [24]. Deep Learning was also investigated [6,11,7]. Moreover, Information Theory quantifiers were explored in this scenario as well, such as permutation entropy [29], multiscale permutation mutual information [27], and so on, showing that this kind of approach can be useful. Techniques derived from Information Theory have also been successful in the characterization of other time series data. For instance, Ordinal Patterns (OP) transformation, a model-free technique that is computationally inexpensive and have low dimensionality, and the Complexity-Entropy Plane [19] are used to characterize the behavior of vehicles through their velocities [1], electric loads [2], crude oil price [18], and financial [26].

Another research direction that has also been successful in the characterization of time series is based on the transformation of the time series into graphs. Using this strategy, networks that inherit the characteristics of the

original time series are constructed (for example, periodic series are transformed into regular graphs, and random series are transformed into random graphs). Some examples are the visibility graph [13], the horizontal visibility graph [14], and the vector visibility graph [17]. However, as each time series sample is transformed into a vertex of the graph, there is an impact on the scalability of these techniques, making them not feasible for high-dimensional time series data.

Recently, methods that combine more than one approach are emerging. In [21, 16, 28, 9], we can see techniques that obtain graphs from permutations of possible patterns in OP transformation, taking advantage of both approaches.

Such works show as Information Theory quantifiers may contribute in understanding and characterize the transportation modes, helping their identification. This work explores this hypothesis using the Ordinal Patterns (OP) [4] transformation. Additionally, we use its graph transformation, known as Ordinal Pattern Transition Network (OPTN) [21], to represent the mobility data in a new domain, and, then, classify it using features taken from such transformation. Particularly, we propose the use of a new feature, derived from OPTN, called Probability of Self-Transition ( $p_{ST}$ ) [5]. The validation of our proposal is made in real-world data, referring to transportation mode identification: we want to characterize which transportation mode (car, bus, bike, and walk) a given person carrying a GPS is traveling.

The main contribution of this work is the comprehensive study of Information Theory concepts to analyze trajectory based on GPS data, in order to identify the transportation modes used. Additionally, we propose a methodology to characterize trajectory information from real data sets that uses OP and its graph transformation in transportation mode identification. Whereas [29] studies only the use of permutation entropy, we extend this analysis to other Information Theory quantifiers, including OPTN. To the best of our knowledge, this is the first work investigating it. We hope that our study, along with others, can shed light in how the Information Theory field can help the understanding of mobility data, creating opportunities to improve the quality of life in cities.

This work is organized as follows. Section 2 presents the fundamentals of the Information Theory methods used in this work, namely Ordinal Patterns and Ordinal Patterns Transition Network, along with the features extracted from them, i.e., permutation entropy, statistical complexity, and probability of self-transition. Section 3 discusses methodology applied to the real data set used. Section 4 relates the results obtained and, finally, Section 5 concludes this work.

## 2 Ordinal Patterns Transformation

OP is a simple method of transforming time series that does not require any model assumption about the time series and can be applied to any arbitrary time series. Furthermore, such a method has an advantage of its simplicity,

speed, robustness, and invariance concerning non-linear monotonic transformations. This approach is based on the sequence that naturally arises from the time series, comparing the values that are in the same neighborhood and replacing them with a sequence of symbols [4].

Let a temporal series  $\mathbf{X}(t) = \{x_1, x_2, \dots, x_n\}$  of size  $n$  and let also an embedding dimension  $D \in \mathbb{N}$  and an embedding delay  $\tau \in \mathbb{N}$ . In each time instant  $t = \{1, \dots, n - (D - 1)\tau\}$ , we have a sliding window  $w_t \subseteq x$ , such as

$$w_t = \{x_t, x_{t+\tau}, \dots, x_{t+(D-2)\tau}, x_{t+(D-1)\tau}\}, \quad (1)$$

i.e., each element within the sliding window is obtained from the time series in the time  $t, \dots, t + (D - 1)\tau$ . This corresponds to a time series sample at evenly spaced intervals.

Moreover, the ordinal relation for each instant  $t$  consists of the permutation  $\pi = \{r_0, r_1, \dots, r_{D-1}\}$  of  $\{0, 1, \dots, D - 1\}$ , so that

$$x_{t-r_{D-1}} \leq x_{t-r_{D-2}} \leq \dots \leq x_{t-r_1} \leq x_{t-r_0}. \quad (2)$$

In other words,  $\pi$  represents the permutation of elements in the sliding window  $w_t$ , in ascending order. In order to obtain unique results, we define that, if a time series have elements such that  $x_{t-r_i} = x_{t-r_{i-1}}$ , we consider that  $r_i < r_{i-1}$ . 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  $\pi_m$  represents a permutation of the possible permutation set of  $D!$  [2].

The choosing of  $D$  depends on the time series size and must satisfy the condition  $n \gg D!$  – the higher  $D$  is, the greater the time series length is necessary to have reliably extracted data [19]. If interested, more explanations are given in [22]. For practical purposes, [4] recommend values such that  $3 \leq D \leq 7$ , which are adopted in this work.

For all  $D!$  possible permutation  $\pi$  of  $D$ , the relative frequency can be computed by the times a certain sequence appeared in the time series, divided by the number of total sequences, obtaining the histogram of the probability distribution  $P \equiv \{p(\pi)\}$ , which is defined by:

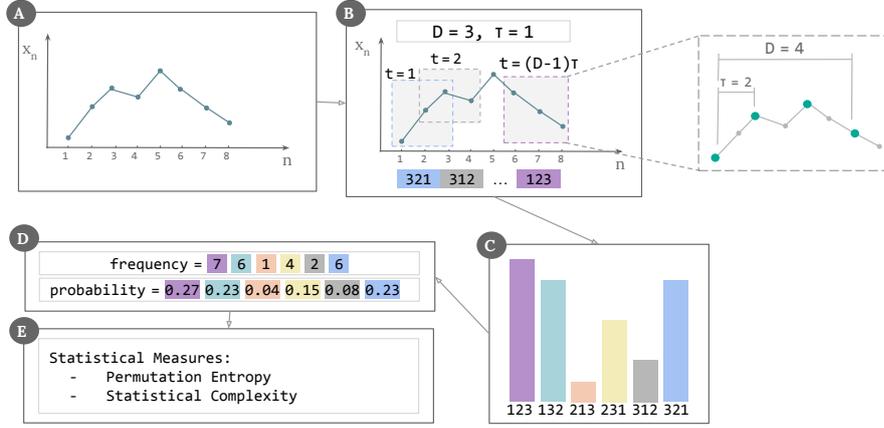
$$p(\pi) = \frac{|s_\pi|}{n - (D - 1)\tau}, \quad (3)$$

where  $|s_\pi| \in \{0, \dots, m\}$  is the number of pattern observed of type  $\pi$ .

From this new representation, it is possible to extract features, such as Information Theory quantification, which can be used to characterize the time series dynamics [19].

Figure 1 illustrates the process described above of extracting OP probability distribution from time series. **(i)** first, we have the original time series; **(ii)** we calculate sliding windows with  $D$  and  $\tau$  values. We can see in the highlighted sliding window how  $D$  and  $\tau$  behave: simplistically,  $D$  is relative to how many data points we consider to calculate the ordinal pattern, when  $D = 3$ , we use three points,  $D = 4$ , we use four points, and so on;  $\tau$  refers to the spacing between two consecutive points in the sliding window, when  $\tau = 1$ , we

use immediate neighboring points, if  $\tau = 2$ , we use points that are two points apart, and so forth. **(iii)** After computing the ordinal patterns to all the time series, we extract the histogram of relative frequency. **(iv)** We can use the probability distribution or the frequency distribution of this histogram, as we can see in the last step of the process.



**Fig. 1** The process of extracting the Ordinal Patterns probability distribution from time series

In this work, we extract two quantifiers, the Permutation Entropy, and Statistical Complexity, as discussed in the following.

## 2.1 Permutation Entropy

The Permutation Entropy is a measure of uncertainty associated with the process described by  $p_\pi$  and is defined by:

$$H[p_\pi] = - \sum p(\pi) \ln p(\pi) , \quad (4)$$

where  $0 \leq H[p_\pi] \leq \log D!$ . This measure is equivalent to the Shannon Entropy [2]. Low values of  $H[p_\pi]$  represent a sequence of increasing or decreasing values in the permutation distribution, indicating that the original time series is deterministic, while high values indicate a completely random system [4].

The maximum value for  $H[p_\pi]$  occurs when all possible permutations of  $D!$  have the same probability of occurring, which is the case for the uniform distribution  $p_u$  of permutation. Thus,  $H_{max} = H[p_u] = \ln D!$  [33]. We can define the normalized Shannon Entropy, for the case of permutation entropy, as:

$$H_S[p_\pi] = \frac{H[p_\pi]}{H_{max}} , \quad (5)$$

where  $0 \leq H_S[p_\pi] \leq 1$  [19].

## 2.2 Statistical Complexity

The Statistical Complexity is based on Jensen-Shannon divergence (JS) between the associated probability distribution  $p_\pi$  and the uniform distribution  $p_u$  (the trivial case for the minimum knowledge of the process) and is defined by:

$$C_{JS}[p_\pi] = Q_{JS}[p_\pi, p_u] \cdot H_S[p_\pi], \quad (6)$$

where  $p_\pi = \{p(\pi)\}$  is the probability distribution of ordinal patterns,  $p_u$  is the uniform distribution and  $H_S$  is the normalized Shannon Entropy, as defined in Eq. (5). The disequilibrium  $Q_{JS}[p_\pi, p_u]$  is given by:

$$Q_{JS}[p_\pi, p_u] = Q_0 \cdot JS[p_\pi, p_u] = Q_0 \cdot \left( S \left[ \frac{p_\pi + p_u}{2} \right] - \frac{S[p_\pi] + S[p_u]}{2} \right), \quad (7)$$

where  $S$  is the Shannon entropy and  $Q_0$  is defined by:

$$Q_0 = -2 \left[ \left( \frac{D! + 1}{D!} \right) \ln(D! + 1) - 2 \ln(2D!) + \ln(D!) \right]^{-1}, \quad (8)$$

which describes the normalization constant, which is equal to the inverse of the maximum value of  $JS[p_\pi, p_u]$  and  $0 \leq Q_{JS} \leq 1$  [2, 19, 25].

## 2.3 Ordinal Pattern Transition Network

Given a sequence of OP  $\Pi \equiv \{\pi_i\}$  the OPTN represents the relation between consecutive patterns and is defined as a weighted directed graph  $G_\pi = (V, E)$ , with vertices  $v_{\pi_i} \in V = \{v_{\pi_i} : i = 1, \dots, D!\}$  that correspond to a possible permutation of  $D!$  to the embedding dimension  $D$ , and edges  $E = \{(v_{\pi_i}, v_{\pi_j}) : v_{\pi_i}, v_{\pi_j} \in V\}$ .

A directed edge connects two OPs in the graph if such patterns appear sequentially in the original time series, representing a transition between the patterns. The weights  $w : E \rightarrow \mathbb{R}$  of the edges represent the probability of existence of a specific transition in  $\Pi$  and is given by:

$$w(v_{\pi_i}, v_{\pi_j}) = \frac{|II_{\pi_i, \pi_j}|}{m - 1}, \quad (9)$$

where  $|II_{\pi_i, \pi_j}| \in \{0, \dots, m - 1\}$  is the number of transitions between the permutations  $\pi_i$  e  $\pi_j$  and  $\sum_{v_{\pi_i}, v_{\pi_j}} w(v_{\pi_i}, v_{\pi_j}) = 1$ .

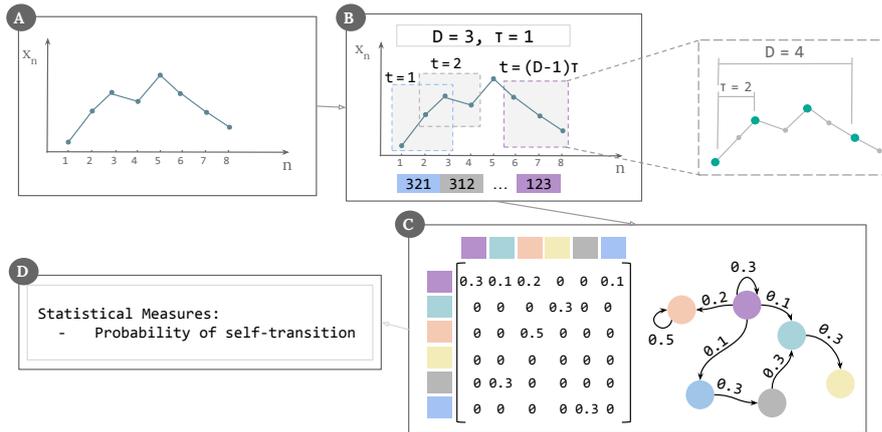
Once the graph is constructed from the OP set, some properties are inherited from this transformation. The most notable are:

- **simplicity and speed:** the graph construction only depends on the number  $m$  of OPs, needing to count the number of transitions in  $m - 1$  steps. In turn, the time series transformation into OP depends on the size  $n$  of the time series and the embedding dimension  $D$ . The time complexity to perform such transformation is bounded by  $O(nD \ln D)$ , assuming that

the permutations are obtained by sorting the data points of each sliding window by a common sort algorithm, such as merge sort (of complexity  $O(D \ln D)$  in the worst case). Using worse sort algorithms, such as selection sort, of complexity  $O(D^2)$  in the worst case, the time complexity increases to  $O(nD^2)$ . However, as  $D$  is recommended to be at most 7, the sorting will take no more than 7 elements, so the complexity of such strategy is more dependent on time series size  $n$  [5].

- **scalability**: the approaches that use a visibility graph [13], for instance, transform each time series sample into a vertex within the graph – an impracticable approach to high-dimensional time series due to the space required for storage. On the other hand, the number of vertices of the OPTN is given by the embedding dimension  $D$ , not depending on the size of the series and being limited by  $D!$ .
- **robustness**: OPs are robust to the presence of noise and invariants with respect to non-linear monotonic transformations [2, 19].

Figure 2 illustrates the process described above: (a) Given a time series; (b) we calculate the sliding windows with values for  $D$  and  $\tau$  ( $D = 3$  and  $\tau = 2$ , in the figure); (c) we obtain the OP; and (d) we build the OPTN with a vertex to each OP found in the time series and with edges that describe the temporal succession of patterns.



**Fig. 2** The process of extracting the Ordinal Patterns Transition Network from time series

## 2.4 Probability of Self-Transition

The self-transitions of the transition graph are the edges from a vertex to itself, also known as loop. Its presence in a graph represents the occurrence of the same OP consecutively.

In [28], the authors proposed an analysis of the entropy computed through the weights of the edges of the transition graph after the removal of the self-transition edges. However, these transitions are directly related to the temporal correlation of the original time series and are a valuable indication of the hidden dynamics and, therefore, should not be discarded. The way these edges are placed is an essential element for the subsequent analysis of the graph.

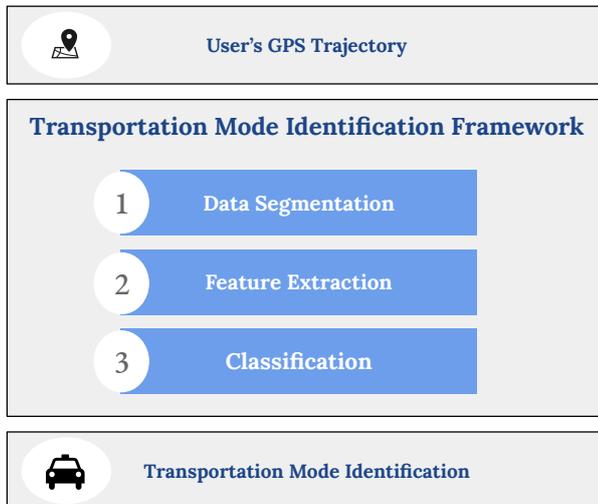
The probability of self transition is defined as the probability of occurrence of a sequence of equal patterns within the OP set and can be expressed as:

$$p_{st} = p(\pi_i, \pi_i) = \sum_{i \in 1, \dots, D!} w(v_{\pi_i}, v_{\pi_i}) . \quad (10)$$

The weight normalization of the graph adopted in this work is similar to that adopted by [28], where the authors normalize the weights such that all weights sum 1. However, in our case, we accept the presence of self-transition [5].

### 3 Transportation Mode Identification Framework

To achieve our goal of acquiring transportation mode information using Information Theory methods, we develop a framework composed of three steps, called Data Segmentation, Feature Extraction, and Classification, as shown in Figure 3. These steps will be better explained throughout this section.



**Fig. 3** Transportation Mode Identification Framework

### 3.1 Data set description

In this work, we use the GeoLife<sup>1</sup> data, collected by [30]. This data set presents GPS trajectories of 182 users over five years (from April 2007 to August 2012), containing latitude, longitude, and altitude information.

Among these users, 73 have transportation mode information, which will be classified in this study. Note that only transportation mode with a duration higher than 1000 hours were considered, since we understand that the smaller the time series, the more difficult to extract relevant information, which leads to the generation of low-quality models. Table 1 describes the transportation mode used in this work. We have four kinds of transportation: walking, bike, bus, and personal car (car/taxi).

**Table 1** Distance and duration of transportation mode

<b>Transport</b>	<b>Distance (km)</b>	<b>Duration (h)</b>
walking	10 123	5460
bike	6495	2410
car/taxi	32 866	2384
bus	20 281	1507

Here, we define trajectory as an uninterrupted sequence of GPS points (latitude and longitude) that belong to the same transportation mode. We consider that every user is at the same altitude, thus discarding this measure.

### 3.2 Data Segmentation

As said previously, we use the portion of GeoLife data set which contains the transportation mode information, that includes 73 users. We segment the raw trajectory provided by separating it by user, day, and transportation mode. This type of segmentation supposes that the transportation mode used in each raw trajectory is known. It is a strong assumption, however, since we are focusing in the use of Information Theory methods, we use such segmentation technique. Several works in literature do the same step, such as [32,6], to cite a few. Moreover, for the same reason, although presented in some previous works, such as [24,6], we do not use noise removal in our framework; it is helpful to evaluate the robustness of our approach as well. Also, we discard trajectories with less than 10 points so we may avoid the creation of low-quality trajectories, which may affect the generated model.

Table 2 presents some statistics about the obtained trajectories: how many trajectories each transportation contains (with the proportion in the data set);

<sup>1</sup> <https://www.microsoft.com/en-us/download/details.aspx?id=52367>

and the average (with the standard variation), maximum, and minimum points of the trajectories by transportation. As in [32], we join car and taxi as driving, since they have fewer trajectories. Particularly, car encompasses 860 trajectories (9.8% of the data set), an average of 627.88 points ( $\pm 866.97$ ), and with the maximum, and minimum trajectory containing 8394 and 12 points, respectively. Taxi consists of 545 trajectories (6.3%), with an average of 470.92 ( $\pm 758.70$ ), and maximum and minimum trajectories size of 10841 and 11 points, respectively.

We note that the data set is imbalanced, with walking containing about the half of the trajectories. However, the average size of the trajectories is smaller, showing that people tend to commute longer when using some kind of transportation.

**Table 2** Obtained trajectories from data set after Data Segmentation step

Transportation	Trajectories	Trajectory Size		
		Average (STD)	Maximum	Minimum
walking	3910 (45.0%)	412.86 ( $\pm 1038.21$ )	19836	11
bus	1856 (21.3%)	691.88 ( $\pm 911.13$ )	19604	11
bike	1593 (18.3%)	602.23 ( $\pm 972.73$ )	22080	11
car/taxi	1340 (15.4%)	567.44 ( $\pm 830.48$ )	10841	11
<b>total</b>	<b>8699 (100%)</b>	<b>530.88 (<math>\pm 976.75</math>)</b>	<b>22080</b>	<b>11</b>

### 3.3 Feature Extraction

This step comprehends the transformation of segmented trajectories to features extracted from OP and OPTN transformation. Trajectory data is usually provided in latitude, longitude, and altitude. These features, however, are difficult to interpret. Hence, their transformation in another feature improve the transportation mode identification. We transform the trajectories (latitude and longitude information) the geographical distance between two succeeding GPS points. We use the geodesic distance between two trajectory points, which can be seen as a generalization of a straight distance to a curved surface. In other words, the geodesic distance is the shortest path between two points on the Earth, using the model of an ellipsoid of revolution. There are several ellipsoid models, being World Geodetic System (WGS) 84 the standard in cartography and satellite navigation, including GPS. We use it in this paper, as explained in [12].

Using distance, latitude, and longitude as features, we transform them into the OP representation, extracting permutation entropy and statistical complexity from them. We also transform their OP representation into OPTN, extracting the probability of self-transition.

Several other steps could be performed, such as extracting more features, but we do not intend to exhaust such possibilities since our goal here is to

analyze how Information Theory can help the transportation mode field more than to achieve the best classification results.

### 3.4 Classification

This step receives the extracted features and classifies them using simple classification methods. To evaluate if our proposal is capable of generalizing, and also to validate our results, we use cross-validation, with 10-folds. It is important to note that this cross-validation is performed in the extracted features, immediately prior to classification. Such features, after going through the transformations described in this work, can be interpreted as independent and identically distributed, allowing the use of this validation method. Moreover, since our data are imbalanced, we stratify each class in order to force each fold to have the same class distribution, i.e., we preserve the percentage of sample for each class in each fold. It ensures that neither of the classes are over-represented, leading to increased unrealistically results.

Since the idea of this work is to highlight the particularity of each data transformation, little effort was devoted to the adjustment of the classification algorithms. It may be possible to obtain better results of the evaluation metrics by adjusting the parameters of the classifier. However, our objective is not only to present good results of such metrics but to know if our proposal is suitable for characterization and classification of trajectories. Our classification was made using simple algorithms, which are: k-Nearest Neighbors (k-NN), with  $k = 2$ ; Support Vector Machines (SVM), with linear (SVM-L) and radial (SVM-R) kernels; Decision Tree (DT), Random Forest (RF) (50 trees), and Gradient Boosting Decision Tree (XGBoost) (50 trees as well).

### 3.5 Algorithm Complexity Analysis

We have the start and end point of each transportation mode used by each user separated from the raw trajectory data. Thus, the Data Segmentation step finds the trajectory related to the transportation mode label and split it. The cost for this operation is  $O(|\ell| rn)$ , where  $|\ell|$  is the size of transportation mode label data,  $r$  represents how many raw trajectories a user has, and  $n$  the size of user's trajectory.

The cost to extract the motion feature, distance, is  $O(n)$ , which is the cost for the Feature Extraction step.

Algorithm 1 shows the pseudo-code to the OP transformation. The lines from 3 to 10 go through all the trajectories, doing the following operations: in line 4, we select the indices of points that will be transformed to OP – that is, we select every  $D$  points, spaced by  $\tau$  value. This costs  $O(1)$ . Line 5 indeed obtains such points values from the trajectory data, based on the *indices* values, with the cost  $O(1)$  as well. The ARGSORT function returns the indices that would sort an array, hence, in line 6 we obtain the ordinal

pattern  $\pi$  that represent the current trajectory slice. It uses a simple algorithm to sort, such as Merge sort, having a cost of  $O(D \ln D)$ . Hence, the complexity cost for the OP transformation is  $O(nD \ln D)$ .

---

**Algorithm 1** Ordinal Patterns Transformation
 

---

```

1: procedure OP_TRANSFORMATION(trajectory,  $D$ ,  $\tau$ )
2:   symbols = list()
3:   for  $p$  in range(1, SIZE(trajectory) - ( $D - 1$ ) $\tau$ ) do
4:     indices = range( $p$ ,  $p + (D - 1)\tau$ ,  $\tau$ )
5:     values = loc(indices, trajectory)
6:      $\pi$  = argsort(values)
7:     symbols.append( $\pi$ )
8:   end for
9: return symbols
10: end procedure

```

---

To obtain the OP distribution, we first obtain the list of patterns, that costs  $O(D!)$ . After, we count each pattern obtained in the trajectory. It costs  $O(n)$ . This is the same cost to obtain OPTN transformation, since we just need to count the transitions between each pattern. To extract each feature the cost is  $O(n)$  as well. Therefore, the Data Transformation step, that includes the OP and OPTN transformation and feature extraction, costs  $O((D!) + (nD \ln D) + O(n)) = O(nD \ln D)$ , since  $D!$  is bounded by at most 7 elements, as explained in Section 2.

The complexity of the Classification step is related to the classification algorithm used. For instance, a RF classifier has cost  $O(N^2 \times f \times t)$ , where  $N$  is the total of trajectories,  $f$  is the total of features, and  $t$  the total of trees.

### 3.6 Implementation

All the implementation procedure in this work was made in Python (version 3.7.3), using the Anaconda distribution (version 4.7.11)<sup>2</sup>. The machine in which they were executed had the following configuration: Ubuntu 18.04.3 OS, 20 × Intel(R) Core(TM) i9-9900X CPU @ 3.50GHz, and 125 GB RAM. The implementation of our framework can be found online at: [https://github.com/icps/tmc\\_ordinal\\_patterns](https://github.com/icps/tmc_ordinal_patterns).

## 4 Results and Discussion

In this section we will present the results for transportation mode identification using features extracted from OP and OPTN transformations, as described in Section 3. First, we will evaluate the influence of the OP parameters, the embedding dimension  $D$  and the embedding delay  $\tau$ , respectively. After, we will examine the classification results.

<sup>2</sup> <https://www.anaconda.com>

#### 4.1 Influence of the embedding dimension $D$

Figure 3 presents the obtained accuracy in the classification of latitude, longitude, and distance, using different values of  $D$ , from 3 to 6. Although  $D = 7$  is also recommended in [4], this value can be unfeasible in this scenario, since, for long trajectories, it can take a long time to transform data. Therefore, we do not use this value.

This figure shows the classification of five sets of features, since we want to evaluate their strength to characterize the transportation features and hence contribute to their identification. The first three figures 4a-4c display the accuracy results for each feature classified independently. The figures 3d and 3e exhibit some combination of features: the former shows the results for the features extracted from OP transformation only and the latter presents the results of the OP combined with the OPTN transformation. All figures have the same values in the y-axis, making it easier to compare the results.

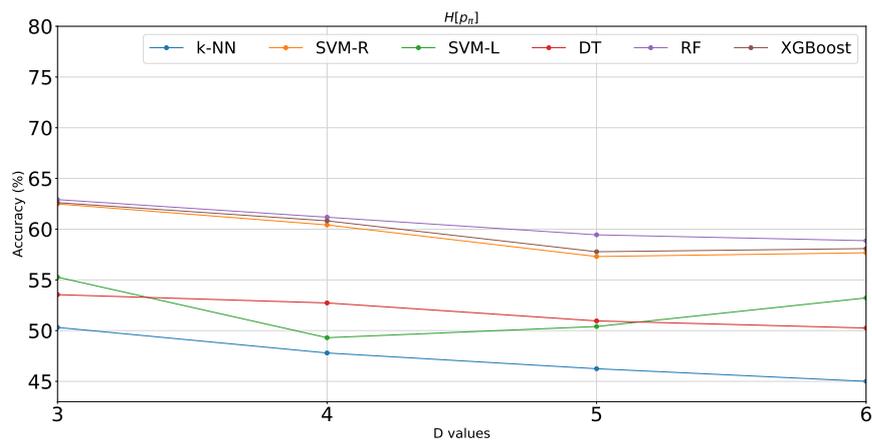
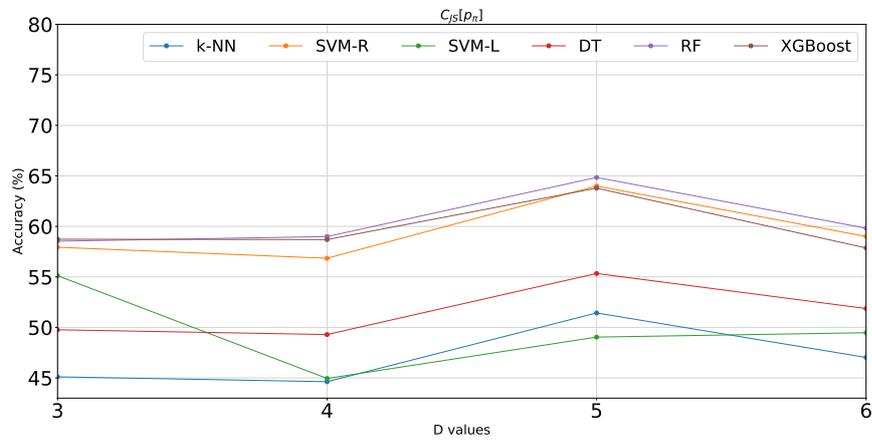
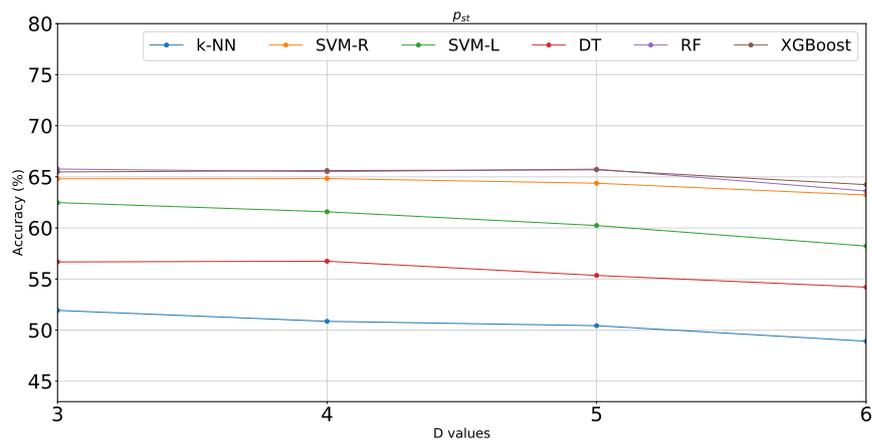
When classifying each feature in isolation, we see that  $C_{JS}[p_\pi]$  and  $p_{st}$  achieves the best accuracy results, about 65%, while the best result for  $H[p_\pi]$  is about 63%. However, we can see that statistical complexity achieves this value only with  $D = 5$ , whereas probability of self-transition demonstrate more robustness, achieving similar results with  $D = 3$ ,  $D = 4$ , and  $D = 5$ ; for  $D = 6$  there is a slight decrease in result, indicating that the relationship in this time series is present in smaller windows of time (the same can be observed with the other features). Additionally, this result implies that the sequence in time series can be important to identify the transportation modes, since this information is lost in OP transformation, while OPTN captures it. Further study must be performed to understand the importance of the OPs sequence information; we do not dive in such direction in this work.

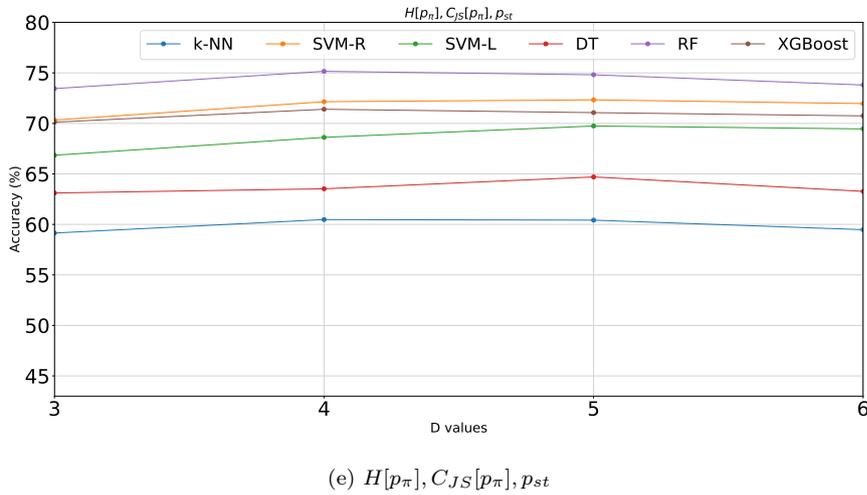
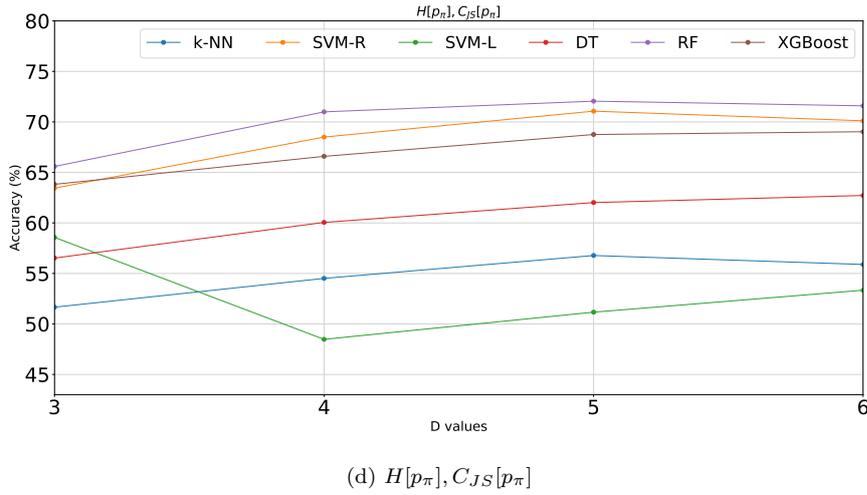
Regarding the classifier, we can see that RF, SVM-R, and XGBoost perform better in the three cases, with similar results. It indicates that the space of these trajectories are not linear, needing a classifier that can explore non-linear spaces to achieve better results.

Moreover, combining the OP features,  $C_{JS}[p_\pi]$  and  $H[p_\pi]$ , give us better accuracy results than the  $p_{st}$  alone, about 72%, a gain of about 7%. Naturally, since they extract different information about the trajectories, the aggregation of both features enhance the results.

Joining the three features for classification give us an improvement of about 3% when compared to the OP features set, reaching about 75%, and about 10% when compared to the features alone. It is one more step to corroborate our early hypothesis of the sequence information being somewhat important in this context. But, clearly, we see that the aggregation of both OP representations brings the best results.

Additionally, we can see that  $D = 4$  and  $D = 5$  present similar results for the three features – in this case, we prefer to use the smallest value, since the cost of the algorithm will be lower. Again, RF performs better, with about 2% of gain against the second better classifier, SVM-R.

(a)  $H[p_\pi]$ (b)  $C_{JS}[p_\pi]$ (c)  $p_{st}$



**Fig. 3** Accuracy results for variation in  $D$  parameter to OP transformation with different set of features

#### 4.2 Influence of the embedding delay $\tau$

Now, we will evaluate the  $\tau$  influence in the transportation mode recognition. We will fix the  $D$  value as four, as shown to be the best result in the previous experiment. The maximum  $\tau$  value depends on the time series size  $n$  and the dimension  $D$ , being limited by:

$$\tau < \frac{n}{D-1}. \quad (11)$$

Hence, the greater the  $\tau$  value, the greater the number of time series samples needed. For example, for  $D = 4$  and  $\tau = 1$ , the time series has to be, at least, greater than 3;  $D = 4$  and  $\tau = 3$ ,  $n > 9$ ;  $D = 4$  and  $\tau = 5$ ,  $n > 15$ ;  $D = 4$  and  $\tau = 10$ ,  $n > 30$ ;  $D = 4$  and  $\tau = 15$ ,  $n > 45$ , and so on. Note that greater  $D$  values require greater trajectories to perform the transformations, which can be not possible to obtain in some cases. Therefore, the preference for smaller values, along with the lower algorithm complexity. In Table 3 we present our data set size  $n$  when  $\tau$  increases. When  $\tau = 15$ , for instance, more than 1000 trajectories are discarded, when compared to  $\tau = 1$ ; i.e., we lose information when increasing  $\tau$ .

**Table 3** Values of  $n$  as  $\tau$  increases

$\tau$	1	5	10	15
walking	3910	3789	3515	3222
bus	1856	1836	1807	1774
bike	1593	1578	1526	1470
car/taxi	1340	1316	1251	1210
<b>total</b>	<b>8699</b>	<b>8519</b>	<b>8099</b>	<b>7676</b>

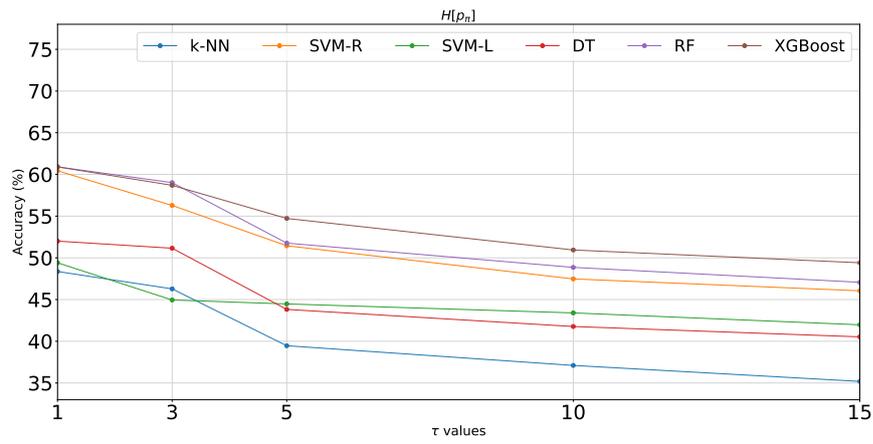
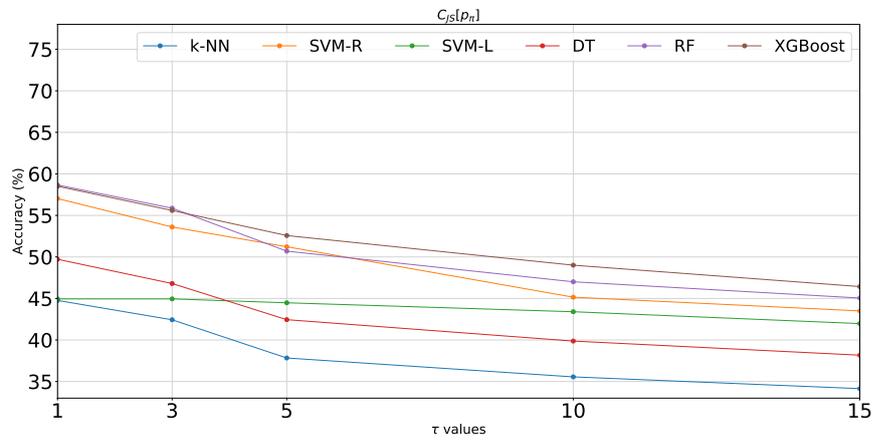
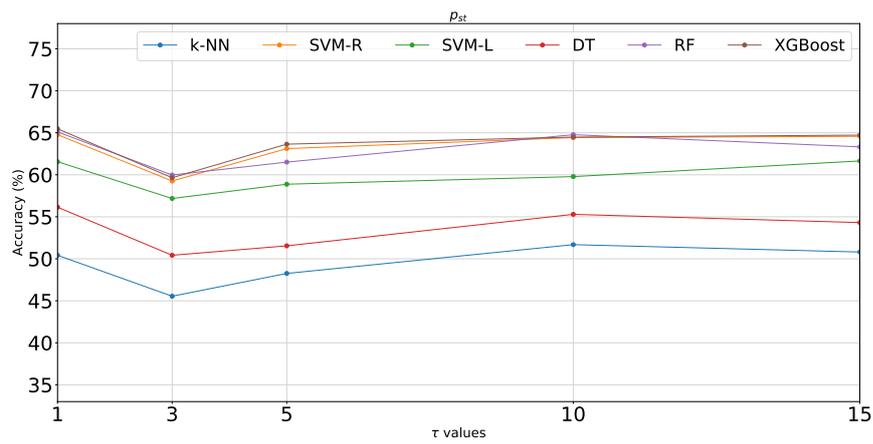
In Figure 3 we see the values obtained when using  $D = 4$  and  $\tau = \{1, 3, 5, 10, 15\}$ . The presentation scheme is similar to the above experiment: the first three figures 4a-4c show the features classified alone, and 3d-3e present the OP set and a set containing the three explored features, respectively. The y-axis has the same value in all figures.

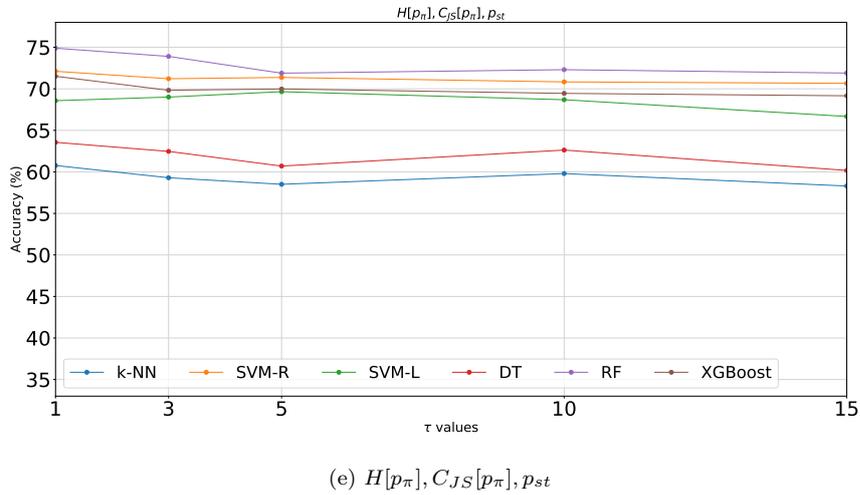
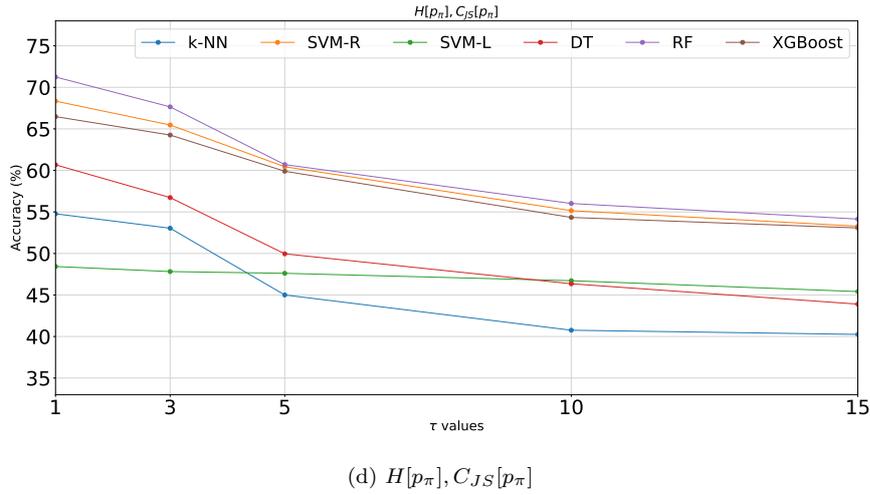
We see that, for the features classified alone, the accuracy value obtained by the OP features,  $C_{JS}[p_\pi]$  and  $H[p_\pi]$ , declines as the  $\tau$  value increases, and both present similar results, about 60%. On the other hand,  $p_{st}$  presents the best results, about 65%, almost 5% of gain. Moreover, differently, in  $p_{st}$ , as the  $\tau$  value increases, the results fall a little but soon return to the previous value obtained when  $\tau = 1$ . It suggests that there is a long relationship between the trajectory points that can be captured when the sequence between the patterns is preserved and that cannot be found in the probability distribution.

As before, when combining the OP features, we may gain more information and hence enhance the results, however, as  $\tau$  increases, the accuracy deteriorates as well. Joining the three features brings the best of both worlds, improving the results and bringing some robustness. Anew, the best classifier is RF, with SVM-R in second.

### 4.3 Classification results

After that, we will adopt the values  $D = 4$  and  $\tau = 1$  and the RF classifier to analyze in more details the transportation modes of the used data set. The

(a)  $H[p_\pi]$ (b)  $C_{JS}[p_\pi]$ (c)  $p_{st}$



**Fig. 3** Accuracy results for variation in  $\tau$  parameter to OP transformation with different set of features

classification was done using the three features,  $p_{st}$ ,  $H[p_\pi]$ , and  $C_{JS}[p_\pi]$ , since, as we saw earlier, this set obtains a better accuracy value.

Table 4 shows the classification for the set of classes presented in the data set, along with the confidence interval (with 95 % confidence). Besides accuracy, we used sensitivity (sen), precision (pre), and F1-score as evaluation metrics, defined as:

- The sensibility (sen) explains how effectively the classifier identifies positive predictions. That is, the ability of our model to identify which individuals pertain to a class;

- The precision (pre) express the proportion of points in data which the model says that they are relevant and they are;
- F1-score is the harmonic mean between precision and sensibility.

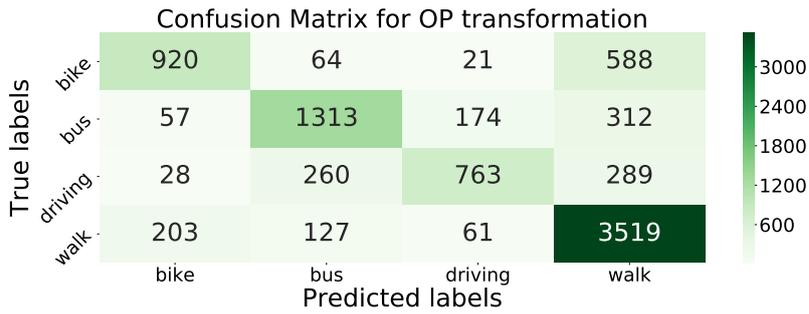
It is possible to see that there are more challenging to distinguish between transportation that, intuitively, travels at a similar pace, as walking and bike, and car/taxi and bus. For more distinct transport, such as walk and bus, best results are achieved.

**Table 4** Evaluation metrics of classification in the used data set

Classes	Pre	Sen	F1	Accuracy
bike	82.00 % ( $\pm$ 2.30)	59.32 % ( $\pm$ 3.00)	68.80 % ( $\pm$ 2.70)	84.46 % ( $\pm$ 1.20)
walk	85.12 % ( $\pm$ 0.98)	94.71 % ( $\pm$ 0.68)	89.65 % ( $\pm$ 0.75)	
bus	90.67 % ( $\pm$ 1.70)	79.90 % ( $\pm$ 1.40)	84.94 % ( $\pm$ 1.40)	90.88 % ( $\pm$ 0.87)
walk	90.97 % ( $\pm$ 0.63)	96.09 % ( $\pm$ 0.72)	93.46 % ( $\pm$ 0.62)	
car/taxi	90.69 % ( $\pm$ 2.10)	74.03 % ( $\pm$ 2.00)	81.47 % ( $\pm$ 1.40)	91.41 % ( $\pm$ 0.62)
walk	91.63 % ( $\pm$ 0.57)	97.37 % ( $\pm$ 0.64)	94.41 % ( $\pm$ 0.40)	
car/taxi	89.08 % ( $\pm$ 2.00)	86.04 % ( $\pm$ 2.60)	87.46 % ( $\pm$ 1.40)	88.75 % ( $\pm$ 1.20)
bike	88.66 % ( $\pm$ 1.70)	91.02 % ( $\pm$ 1.90)	89.78 % ( $\pm$ 1.10)	
car/taxi	77.94 % ( $\pm$ 2.60)	71.72 % ( $\pm$ 3.10)	74.64 % ( $\pm$ 2.50)	79.60 % ( $\pm$ 1.90)
bus	80.73 % ( $\pm$ 1.80)	85.29 % ( $\pm$ 2.00)	82.92 % ( $\pm$ 1.60)	
bike	85.54 % ( $\pm$ 1.20)	90.65 % ( $\pm$ 1.50)	87.99 % ( $\pm$ 0.80)	88.58 % ( $\pm$ 0.75)
bus	91.58 % ( $\pm$ 1.20)	86.80 % ( $\pm$ 1.40)	89.10 % ( $\pm$ 0.73)	
bike	76.38 % ( $\pm$ 3.10)	57.75 % ( $\pm$ 3.10)	65.64 % ( $\pm$ 2.40)	74.89 % ( $\pm$ 1.10)
bus	74.54 % ( $\pm$ 2.50)	70.74 % ( $\pm$ 2.70)	72.53 % ( $\pm$ 2.10)	
car/taxi	74.93 % ( $\pm$ 2.30)	56.94 % ( $\pm$ 2.20)	64.67 % ( $\pm$ 2.00)	
walk	74.78 % ( $\pm$ 1.20)	90.00 % ( $\pm$ 1.20)	81.67 % ( $\pm$ 0.87)	

In figure 4 we see the confusion matrix to the complete set of the investigated transportation modes. As stated before, similar transportation modes are more difficult to distinguish, such as walk and bike; especially, driving is heavily misclassified as bus and walk. Regarding bus, it is a plausible condition, since all of these transportation modes are road-based, and may have similar features values. Moreover, there is not an easy explanation to the misclassification between driving and walk. One can suppose that an extraordinary fact were occurring that forced the car or taxi to have similar features to walk or vice-versa. In fact, it would be necessary to analyze the events that were influencing the mobility captured by the GPS data to understand this situation.

Comparing to previous works, although our results are not the state-of-the-art, with 74.89% of accuracy, we present some clearly gain using the OP transformation, presenting similar results to [31] (74.30% of accuracy) and [32] (76.20%), that use more complex and hardware intensively methods than ours.



**Fig. 4** Confusion matrix using the RF classifier, for  $D = 4$ ,  $\tau = 1$ , and  $\{C_{JS}[p_\pi], H[p_\pi], p_{st}\}$

## 5 Conclusion

In this work, we used the Ordinal Patterns transformations to classify transportation modes recorded as GPS trajectories. We transformed the GPS trajectory data, which is a time series, into Ordinal Patterns, and afterwards, we transformed such patterns into the Transition Network and the Probability Distribution of the pattern frequency. From the latter, we extracted two well-known Information Theory quantifiers, which are the Permutation Entropy and the Statistical Complexity; and from the former, we extracted the probability of self-transition, which is directly related to the temporal correlation of the original time series.

Such feature presents better accuracy results than Permutation Entropy and Statistical Complexity. Hence we can affirm that the probability of self-transition satisfactorily characterizes the trajectories. Besides that, this feature has less dependence from the embedding dimension  $D$  and embedding delay  $\tau$ , the needed parameters of Ordinal Patterns transformation. Note that, although our proposal is validated here to the transportation mode identification, we believe that the findings presented here may be applied in practical situations as well as in technological applications related to time series classification problems. In this scenario, the characteristics of OP transformation can be advantageous, such as its simplicity, robustness, and speed.

Furthermore, to the best of our knowledge, this is the first work that uses Information Theory quantifiers derived from Ordinal Patterns Transition Network to transportation mode identification, showing that it is a feasible approach to this kind of problem.

For future work, we intend to extensively study more features, in order to achieve better time series characterization, and, consequently, better classification. Moreover, we intend to test our approach in different data sets to evaluate its robustness facing different problems.

## 6 Declarations

**Funding:** This work was financially supported by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ), and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

**Code availability:** The implementation of our framework can be found online at: [https://github.com/icps/tmc\\_ordinal\\_patterns](https://github.com/icps/tmc_ordinal_patterns).

**Conflict of interest:** The authors declare that they have no conflict of interest.

**Ethics approval:** Not applicable.

**Consent to participate:** Not applicable.

**Consent for publication:** Not applicable.

## References

1. Aquino, A., Cavalcante, T., Almeida, E., Frery, A., Rosso, O.: Characterization of vehicle behavior with information theory. *The European Physical Journal B* **88**(10), 257 (2015)
2. Aquino, A., Ramos, H., Frery, A., Viana, L., Cavalcante, T., Rosso, O.: Characterization of electric load with information theory quantifiers. *Physica A: Statistical Mechanics and its Applications* **465**, 277 – 284 (2017). DOI <https://doi.org/10.1016/j.physa.2016.08.017>
3. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery* **31**(3), 606–660 (2017)
4. Bandt, C., Pompe, B.: Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.* **88**, 174102 (2002). DOI [10.1103/PhysRevLett.88.174102](https://doi.org/10.1103/PhysRevLett.88.174102)
5. Borges, J.B., Ramos, H.S., Mini, R.A., Rosso, O.A., Frery, A.C., Loureiro, A.A.: Learning and distinguishing time series dynamics via ordinal patterns transition graphs. *Applied Mathematics and Computation* **362**, 124554 (2019)
6. Dabiri, S., Heaslip, K.: Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation research part C: emerging technologies* **86**, 360–371 (2018)
7. Endo, Y., Toda, H., Nishida, K., Kawanobe, A.: Deep feature extraction from trajectories for transportation mode estimation. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 54–66. Springer (2016)
8. Ermes, M., Pärkkä, J., Mäntyjärvi, J., Korhonen, I.: Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *IEEE transactions on information technology in biomedicine* **12**(1), 20–26 (2008)
9. Guo, H., Zhang, J.Y., Zou, Y., Guan, S.G.: Cross and joint ordinal partition transition networks for multivariate time series analysis. *Frontiers of Physics* **13**(5), 130508 (2018)
10. Handcock, R.N., Swain, D.L., Bishop-Hurley, G.J., Patison, K.P., Wark, T., Valencia, P., Corke, P., O’Neill, C.J.: Monitoring animal behaviour and environmental interactions using wireless sensor networks, GPS collars and satellite remote sensing. *Sensors* **9**(5), 3586–3603 (2009)

11. Jiang, X., de Souza, E.N., Pesaranghader, A., Hu, B., Silver, D.L., Matwin, S.: Trajectory-rynet: An embedded GPS trajectory representation for point-based classification using recurrent neural networks. In: Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering, pp. 192–200. IBM Corp. (2017)
12. Karney, C.F.F.: Algorithms for geodesics. *Journal of Geodesy* **87**(1), 43–55 (2013). DOI 10.1007/s00190-012-0578-z
13. Lacasa, L., Luque, B., Ballesteros, F., Luque, J., Nuño, J.C.: From time series to complex networks: The visibility graph. *Proceedings of the National Academy of Sciences* **105**(13), 4972–4975 (2008). DOI 10.1073/pnas.0709247105
14. Luque, B., Lacasa, L., Ballesteros, F., Luque, J.: Horizontal visibility graphs: Exact results for random time series. *Phys. Rev. E* **80**, 046103 (2009). DOI 10.1103/PhysRevE.80.046103
15. Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J., Korhonen, I.: Activity classification using realistic data from wearable sensors. *IEEE Transactions on information technology in biomedicine* **10**(1), 119–128 (2006)
16. Ravetti, M.G., Carpi, L.C., Gonçalves, B.A., Frery, A., Rosso, O.A.: Distinguishing noise from chaos: Objective versus subjective criteria using horizontal visibility graph. *PLOS ONE* **9**(9), 1–15 (2014). DOI 10.1371/journal.pone.0108004
17. Ren, W., Jin, N.: Vector visibility graph from multivariate time series: a new method for characterizing nonlinear dynamic behavior in two-phase flow. *Nonlinear Dynamics* **97**(4), 2547–2556 (2019)
18. Ribeiro, H.V., Jauregui, M., Zunino, L., Lenzi, E.K.: Characterizing time series via complexity-entropy curves. *Phys. Rev. E* **95**, 062106 (2017). DOI 10.1103/PhysRevE.95.062106
19. Rosso, O.A., Larrondo, H.A., Martin, M.T., Plastino, A., Fuentes, M.A.: Distinguishing noise from chaos. *Phys. Rev. Lett.* **99**, 154102 (2007). DOI 10.1103/PhysRevLett.99.154102
20. Shah, R.C., Wan, C.y., Lu, H., Nachman, L.: Classifying the mode of transportation on mobile phones using GIS information. In: Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing, pp. 225–229. ACM (2014)
21. Small, M.: Complex networks from time series: Capturing dynamics. In: 2013 IEEE International Symposium on Circuits and Systems, pp. 2509–2512. IEEE (2013)
22. Staniek, M., Lehnertz, K.: Parameter selection for permutation entropy measurements. *International Journal of Bifurcation and Chaos* **17**(10), 3729–3733 (2007)
23. Wang, J., Young, K., Hock, T., Lauritsen, D., Behringer, D., Black, M., Black, P.G., Franklin, J., Halverson, J., Molinari, J., et al.: A long-term, high-quality, high-vertical-resolution GPS dropsonde dataset for hurricane and other studies. *Bulletin of the American Meteorological Society* **96**(6), 961–973 (2015)
24. Xiao, Z., Wang, Y., Fu, K., Wu, F.: Identifying different transportation modes from trajectory data using tree-based ensemble classifiers. *ISPRS International Journal of Geo-Information* **6**(2), 57 (2017)
25. Xiong, H., Shang, P., He, J., Zhang, Y.: Complexity and information measures in planar characterization of chaos and noise. *Nonlinear Dynamics* pp. 1–15 (2020)
26. Yin, Y., Shang, P.: Weighted multiscale permutation entropy of financial time series. *Nonlinear Dynamics* **78**(4), 2921–2939 (2014)
27. Yin, Y., Wang, X., Li, Q., Shang, P., Gao, H., Ma, Y.: Multiscale permutation mutual information quantify the information interaction for traffic time series. *Nonlinear Dynamics* **102**(3), 1909–1923 (2020)
28. Zhang, J., Zhou, J., Tang, M., Guo, H., Small, M., Zou, Y.: Constructing ordinal partition transition networks from multivariate time series. *Scientific reports* **7**(1), 7795 (2017)
29. Zhang, L., Liu, L., Bao, S., Qiang, M., Zou, X.: Transportation mode detection based on permutation entropy and extreme learning machine. *Mathematical Problems in Engineering* **2015** (2015)
30. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding mobility based on GPS data. In: Proceedings of the 10th international conference on Ubiquitous computing, pp. 312–321. ACM (2008)

31. Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding mobility based on GPS data. In: Proceedings of the 10th international conference on Ubiquitous computing, pp. 312–321. ACM (2008)
32. Zheng, Y., Liu, L., Wang, L., Xie, X.: Learning transportation mode from raw GPS data for geographic applications on the web. In: Proceedings of the 17th international conference on World Wide Web, pp. 247–256. ACM (2008)
33. Zunino, L., Soriano, M.C., Rosso, O.A.: Distinguishing chaotic and stochastic dynamics from time series by using a multiscale symbolic approach. *Phys. Rev. E* **86**, 046210 (2012). DOI 10.1103/PhysRevE.86.046210

# Figures

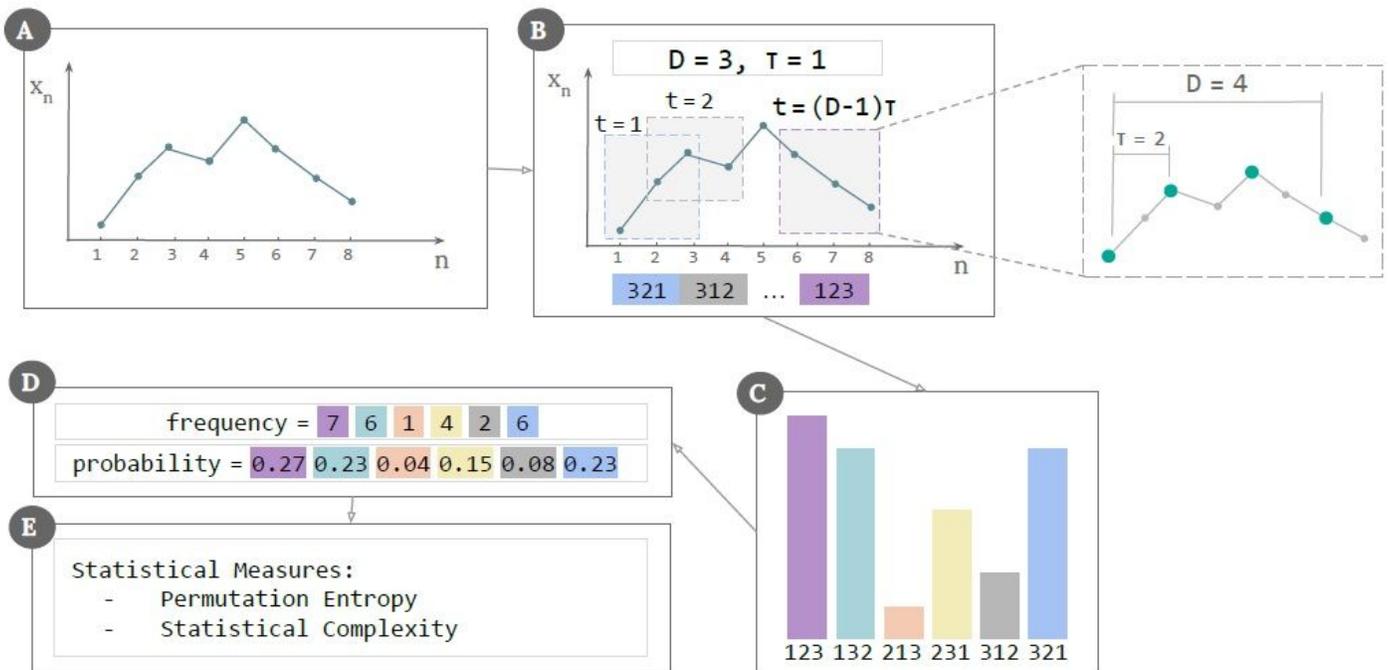


Figure 1

The process of extracting the Ordinal Patterns probability distribution from time series

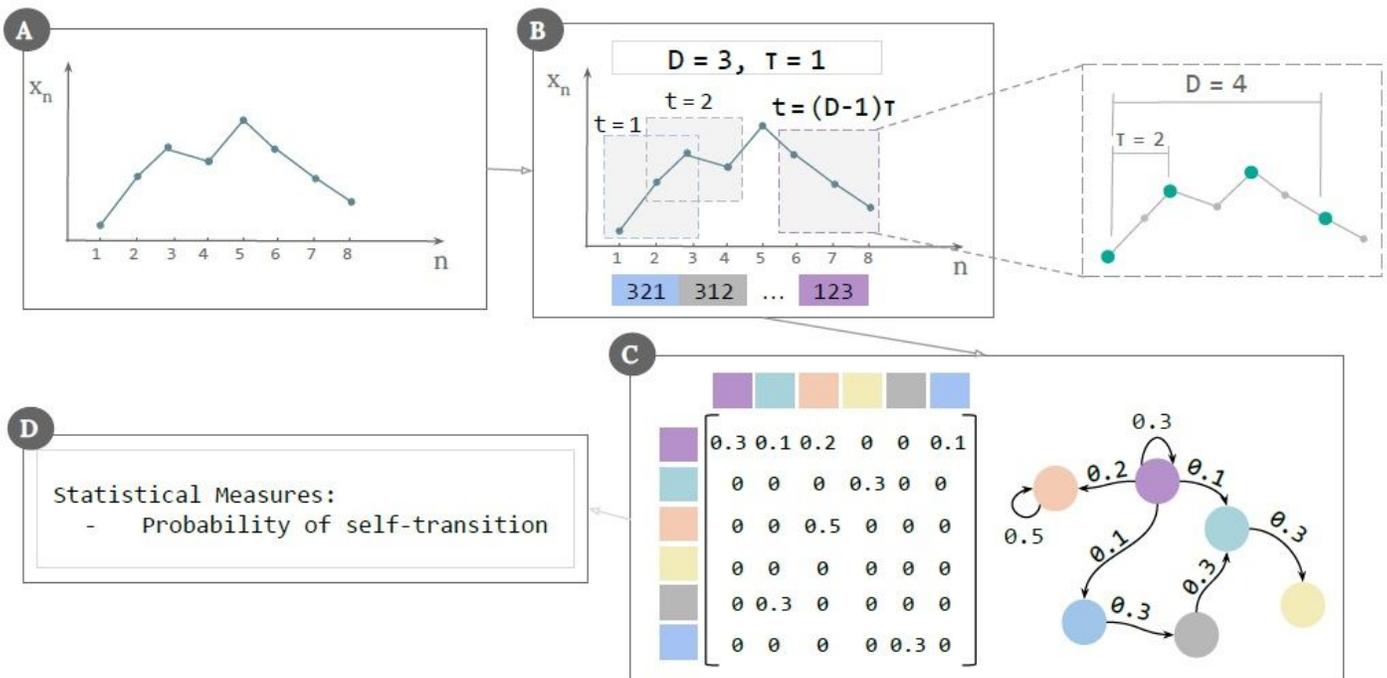


Figure 2

The process of extracting the Ordinal Patterns Transition Network from time series

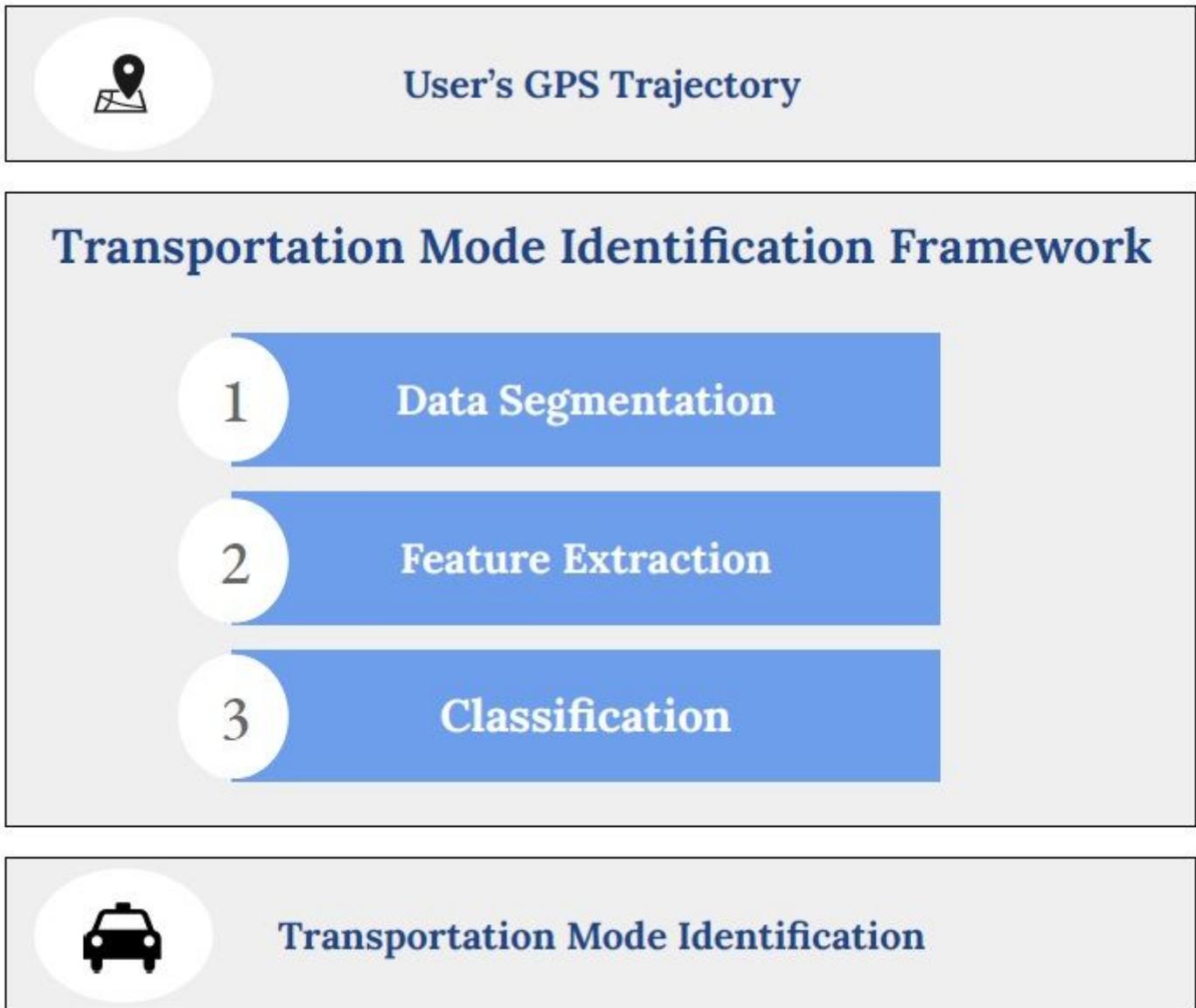
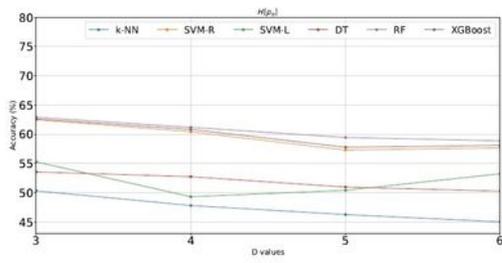
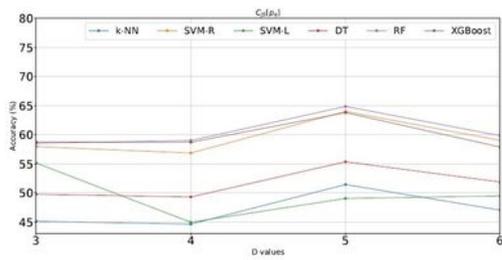


Figure 3

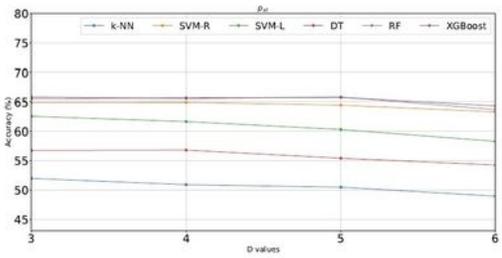
Transportation Mode Identification Framework



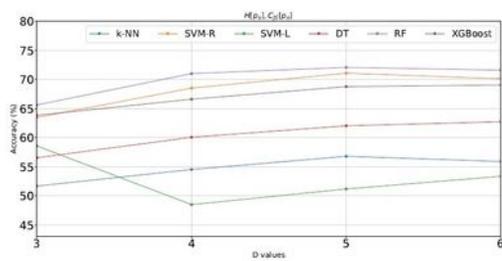
(a)  $H[p_n]$



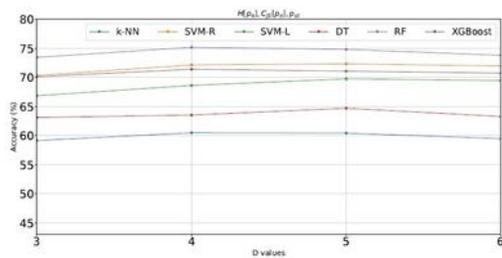
(b)  $C_{JS}[p_n]$



(c)  $p_n$



(d)  $H[p_n], C_{JS}[p_n]$



(e)  $H[p_n], C_{JS}[p_n], p_n$

**Figure 4**

Accuracy results for variation in r parameter to OP transformation with different set of features

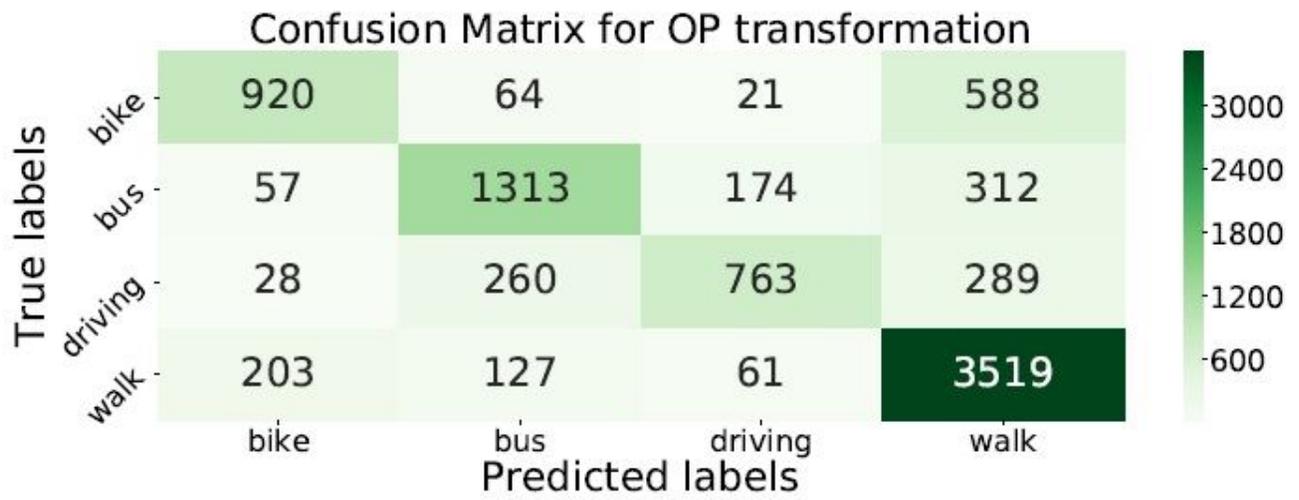


Figure 5

Confusion matrix using the RF classifier, for  $D = 4$ ,  $r = 1$ , and  $\{CJS[pr]; H[pr]; pst\}$