



# The Trajectory Interval Forest Classifier for Trajectory Classification

Cristiano Landi  
University of Pisa  
Pisa, Italy  
cristiano.landi@phd.unipi.it

Riccardo Guidotti, Anna  
Monreale  
University of Pisa  
Pisa, Italy  
{name}.{surname}@unipi.it

Mirco Nanni  
ISTI-CNR  
Pisa, Italy  
mirco.nanni@isti.cnr.it

## ABSTRACT

GPS devices generate spatio-temporal trajectories for different types of moving objects. Scientists can exploit them to analyze migration patterns, manage city traffic, monitor the spread of diseases, etc. Many current state-of-the-art models that use this data type require a not negligible running time to be trained. To overcome this issue, we propose the Trajectory Interval Forest (TIF) classifier, an efficient model with high throughput. TIF works by calculating various mobility-related statistics over a set of randomly selected intervals. These statistics are used to create a tabular representation of the data, which can be used as input for any classical classifier. Our results show that TIF is comparable to or better than state-of-art in terms of accuracy and is orders of magnitude faster.

## CCS CONCEPTS

• **Information systems** → **Decision support systems**; **Data mining**; **Spatial-temporal systems**; **Location based services**; • **Computing methodologies** → **Artificial intelligence**; **Knowledge representation and reasoning**; **Supervised learning**.

## KEYWORDS

GPS Trajectory Classification, Mobility Data Analysis

### ACM Reference Format:

Cristiano Landi, Riccardo Guidotti, Anna Monreale, and Mirco Nanni. 2023. The Trajectory Interval Forest Classifier for Trajectory Classification. In *The 31st ACM International Conference on Advances in Geographic Information Systems (SIGSPATIAL '23)*, November 13–16, 2023, Hamburg, Germany. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/3589132.3625617>

## 1 INTRODUCTION

Smartphones, connected cars, and tracking devices with GPS capabilities produce enormous amounts of mobility data. This data pertains to the movements of various entities and is used by governments, businesses, and researchers for various purposes such as determining the transportation modes used [6, 14], ascertaining the identity of the user who generated the trajectory [18], etc. In [5], the authors compared the latest trajectory classification methods and highlighted some problems in the field. For example, they show

that the literature has an unbalanced focus on defining case-specific features rather than building general-purpose models. Moreover, most techniques heavily rely on computationally complex models such as Support Vector Machines (SVM), Multilayer Perceptrons, and Deep Convolutional Neural Networks (CNN). Unfortunately, each of these models has its own set of drawbacks: SVMs tend to be highly sensitive to noisy data, while CNNs require a substantial amount of training data to reach good performances.

In the field of time series classification, there has been a rapid advance in recent years [2]. One of the most interesting methods is the Canonical Interval Forest (CIF) [17]. CIF first transforms the input time series into a matrix of tabular-like data where every column is a feature. The set of features computed by CIF is called CATCH22 [16]. CATCH22 is a collection of 22 features extracted from the *hctsa* package maximizing the classification performance while minimizing the redundancy. Then, it uses a Random Forest (RF) on the transformed dataset to solve the classification task.

Inspired by CIF and aiming to reach similar performances in the field of GPS trajectory classification, we propose Trajectory Interval Forest (TIF), a machine learning model for general-purpose GPS trajectory classification. TIF extracts mobility-related statistics over randomly selected intervals and then uses a RF to address the classification task. The set of intervals can be defined by the number of observations, elapsed time, or traveled distance. TIF can also handle different length trajectories using three strategies. Therefore, the contribution of TIF over CIF is fourfold: (i) definition of mobility-specific statistics, (ii) simultaneous consideration of multiple signals, (iii) exploitation of spatio-temporal intervals besides observational ones, (iv) usability with objects of different sizes instead of objects with a fixed length.

## 2 BACKGROUND AND PROBLEM SETTING

In this section, we define all the concepts necessary to understand our proposal. We define a trajectory as follows:

**Definition 1** (Trajectory). A *trajectory*  $X$  is a sequence of spatio-temporal points  $X = \{(\vec{x}_1, t_1), \dots, (\vec{x}_m, t_m)\} \in \mathbb{R}^{m \times 3}$  where the spatial vectors  $\vec{x}_j = (\text{lat}_j, \text{long}_j)$  are sorted by increasing time  $t_j$ .

A trajectory classification dataset is a set of trajectories with a vector of labels attached.

**Definition 2** (Trajectory Classification Dataset). A *trajectory classification dataset*  $\mathcal{D} = (\mathcal{X}, \mathbf{y}) \in \mathbb{R}^{n \times m \times 3} \times \mathbb{N}^n$  is a set of  $n$  trajectories,  $\mathcal{X} = \{X_1, \dots, X_n\}$ , with a vector of assigned labels (or classes),  $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$  where  $y_i$  identifies a characteristic of the trajectory  $X_i$  such as the means type of transport.



This work is licensed under a Creative Commons Attribution International 4.0 License.

*SIGSPATIAL '23*, November 13–16, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0168-9/23/11.

<https://doi.org/10.1145/3589132.3625617>

For simplicity of notation, we use a single symbol  $m$  to denote the lengths of the trajectories, even if a trajectory dataset usually contains instances having a different number of observations.

**Definition 3** (Trajectory Classification). Given a trajectory classification dataset  $\mathcal{D}$ , *trajectory classification* is the task of defining a function  $f$  from the space of possible input trajectories  $\mathcal{X}$  to a probability distribution over the class values in  $\mathbf{y}$ .

Most proposals that deal with sequential data often apply a transformation before training the model. We can make this explicit by defining the trajectory classification function  $f$  as the composition of a feature extraction function  $g$ , that maps every trajectory to a fixed-sized set of input features and a function  $h$  from the space of possible inputs  $g(\mathcal{X})$  to a probability distribution over the class values in  $\mathbf{y}$ , i.e.,  $y = (h \circ g)(X)$ . Since our objective is to realize a variant of CIF [17] for trajectory classification, as in CIF, we implement  $h$  as a Random Forest classifier, while the rest of the paper focuses on defining the feature extraction function  $g$ .

To refer to the subsequence of  $X$  between the lower and upper bounds  $(l, u)$ , we use  $X_{l:u}$ .

**Definition 4** (Interval). An interval  $k$  is a tuple of bounds  $(l, u)$  indicating the beginning  $l$  and the end  $u$  of the interval.  $K$  denote a set of intervals and  $|K|$  denote its size.

### 3 TRAJECTORY INTERVAL FOREST

Inspired by CIF [17], we propose Trajectory Interval Forest (TIF), an interval-based approach to efficiently and effectively solve trajectory classification. To this aim, we defined an alternative feature set on mobility data to replace CATCH22. After that, we defined and implemented TIF extending CIF to deal with specific problems of GPS trajectories.

#### 3.1 Mobility Features Description

As the first step, we collected the largest possible number of measures presented in 20 years of literature to extract mobility features. In particular, we categorized these measures into three groups:

- *Point-based features*  $\mathcal{P}$  [3, 6, 9, 10, 19], i.e., features computed using at most two observations of the GPS trajectory.
- *Aggregated features*  $\mathcal{A}$  [9, 10, 20–22], i.e., aggregation functions applied to sequences of point-based features.
- *Interval-based features*  $\mathcal{I}$  [1, 9], i.e., non-canonical aggregation functions computed on a subsequence of the trajectory.

Details of the measures are reported in Table 1. We highlight that all the features can be calculated in linear time with respect to the number of observations in the trajectory  $X$ .

Among the point-based features  $\mathcal{P}$ , besides traditional measures such as *distance*, *speed* and *acceleration*, we underline the presence of the feature *direction*  $dir_i$ , i.e., the direction towards which an object is moving at time  $i$ . In the definition of  $dir_i$ ,  $\arctan2(a, b)$ , computes the angle between a point in a plane and the origin. The *turning angle* is then defined as the difference between two consecutive directions.

Among the aggregated features  $\mathcal{A}$ , aggregating the values obtained in group  $\mathcal{P}$ , we underline the presence of the features *rate upper* and *rate below*. These functions compute the frequency of

an event in  $\mathcal{P}$ , i.e., when the value exceeds or goes below a certain threshold  $\theta^1$ . Like in [9], we used a normalization approach in *rate upper* and *rate below* to derive frequency values in unit distance using the traveled distance.

Among the interval-based features  $\mathcal{I}$ , the *mean squared displacement* captures the variation of the movements in both latitude and longitude; the *straightness* measures the ratio between the shortest path from the origin to the destination and the actual trajectory; the *intensity use* estimates how much of the movement area, expressed as a squared tile, is used by the moving object; and the *sinuosity* captures the curving shape of the trajectory.

#### 3.2 TIF Feature Extraction Algorithm

Given the trajectory dataset  $\mathcal{X}$  and the set of intervals  $K$ , the algorithm implements the function  $g$  returning a dataset  $Z \in \mathbb{R}^{|\mathcal{X}| \times c \times |K|}$ , where  $c$  is the number of features to be extracted. For each trajectory  $X \in \mathcal{X}$ , and for each interval  $k = (l, u) \in K$ , TIF extracts the corresponding subsequence from  $X$ , i.e.,  $X_{l:u}$ . Details about how the intervals can be defined are provided in the next subsection. After that, TIF computes the point-based features  $z^{(P)}$  by applying the point-based features extraction functions  $\mathcal{P}$  on the subsequence  $X_{l:u}$ . Depending on the desired point-based features required, TIF might include in  $z^{(P)}$  also the raw latitude, longitude, and timestamp. Then, TIF computes the aggregated features  $\mathcal{A}$  on each point-based feature  $z^{(P)}$ , and stores it in  $z^{(A)}$ . Consequently, it calculates the interval-based features  $\mathcal{I}$  on  $X_{l:u}$  and stores them in  $z^{(I)}$ . The transformation of the trajectory  $X$  into features  $z$  is accomplished by concatenating  $z^{(A)}$  and  $z^{(I)}$  into  $z$ . Finally,  $z$  is added to the transformed feature dataset  $Z$ . The feature-based dataset  $Z$  can then be used as input to any classifier implementing the predictive function  $h$ , such as Random Forests [4], which is our main choice in line with CIF [17]. The time complexity of TIF is  $O(n \cdot c|K| \cdot \bar{m})$  where  $n$  is the number of trajectories in the dataset,  $|K|$  is the number of intervals, and  $\bar{m}$  is their average length.

#### 3.3 Interval Types and Filling Strategies

The transition from time series data to trajectory data introduces several challenges, mainly due to (i) the varying sampling rates, (ii) objects of different lengths, and (iii) geographic displacement. Indeed, GPS trajectory data are often subject to multiple sources of disturbance, such as measurement errors from sensors or the failure to transmit data due to the conformation of the territory.

As a consequence, to address the issue of non-constant sampling rates, three variants w.r.t. the standard CIF implementation are proposed. A first version of TIF, that we name TIF-o, identifies an interval  $k = (l, u)$  through the indexes of the spatio-temporal points in a trajectory  $X$ , i.e.,  $X_{l:u} = \{(\bar{\mathbf{x}}_i, t_i) | i \in [l, u]\}$ . Then, the other two versions, named TIF-T and TIF-s, identify an interval  $k = (l, u)$  respectively as  $X_{l:u} = \{(\bar{\mathbf{x}}_i, t_i) | l \leq t_i \leq u\}$  and  $X_{l:u} = \{(\bar{\mathbf{x}}_i, t_i) | l \leq \sum_{j=1}^i \text{dist}(\mathbf{x}_{j-1}, \mathbf{x}_j) \leq u\}$ . Finally, we name TIF-A the variant of TIF calculating features using heterogeneous types of intervals.

Furthermore, we tackle the issue of having trajectories of different lengths through four possible approaches. In the first one, the operations over intervals exceeding the maximum length of

<sup>1</sup>In our experiments, we set  $\theta$  equals to the 25% and 75% of the mean as values of  $\theta$  for upper and lower bound, respectively.

	Name	Equation	Description
$\mathcal{P}$	distance	$dist_i =   \bar{x}_i - \bar{x}_{i+1}  $	Generic distance measure.
	speed	$speed_i = \frac{dist_i}{t_{i+1} - t_i}$	The classical speed formulation.
	acceleration	$accel_i = \frac{dist_i}{(t_{i+1} - t_i)^2}$	The classical acceleration formulation.
	direction	$dir_i = \arctan2(lat_{i+1} - lat_i, long_{i+1} - long_i)$	The direction along which the object is moving.
	turning angle	$turn_i = dir_{i+1} - dir_i$	The difference in the direction.
$\mathcal{A}$	statistics	sum, max, min, mean, std, cov, var	Generic aggregation functions
	rate upper	$rate_{upper} = \frac{ \{p > \theta \mid p \in \mathcal{P}(X)\} }{sum(\{dist_i \mid i \in K\})}$	Frequency in which the $\mathcal{P}$ feature exceeds $\theta$ in unit distance.
	rate below	$rate_{below} = \frac{ \{p \leq \theta \mid p \in \mathcal{P}(X)\} }{sum(\{dist_i \mid i \in K\})}$	Frequency in which the $\mathcal{P}$ feature is below $\theta$ in unit distance.
$\mathcal{I}$	mean squared displacement	$msd = var(X_{lat}) + var(X_{long})$	Dispersion in both the latitude and longitude dimensions.
	straightness	$str = \frac{  \bar{x}_0 - \bar{x}_m  }{\sum_i dist_i}$	Compare the distance between all the points along to the minimum distance between the origin and the destination.
	intensity use	$iu = \frac{\sum_i dist_i}{\sqrt{Area\ of\ movement}}$	Ratio between total trajectory length and area of movement.
	sinuosity	$sin = 2 \left[ p \left( \frac{1-c^2-s^2}{(1-c)^2+s^2} + b^2 \right) \right]^{-0.5}$	Complexity of the trajectory where $p$ is the mean step length, $c, s$ are the mean cosine (resp. sine) of the turning angles, $b$ is the variation of step length.

Table 1: Description of features calculated on trajectories subsequences.

the trajectory under analysis are actually performed just on the nearest two available observations. We identify this variant with TIF- $\mathcal{N}$  (naive). The second and third alternatives consist in two different “filling strategies” that aim to make all the trajectories in  $\mathcal{X}$  of the same length  $m$ . The first filling strategy is named *reverse fill* (TIF-R), and it appends to the trajectory a mirrored copy of itself, simulating the object moving back along the same path, till reaching length  $m$ . The second filling strategy is named *echo fill* (TIF-E), and it is implemented by repeating the same trajectory from its beginning, i.e., by translating the moving object such that the “new” starting point coincides with the ending point of the previous iteration. Finally, as an alternative to filling strategies, we extend all the variants of TIF to manage intervals as percentages instead of as actual values. The totals to calculate such percentages are based respectively on the number of spatio-temporal points, on the total duration of the travel, and the total distance traveled by the trajectory. Thus, by using percentages instead of actual values, the requirement of having all the trajectories with the same length is not mandatory anymore, and we can also avoid any filling strategy. We name this variant TIF-P.

The geographic displacement of the trajectories poses an additional factor to consider. Including the latitude and longitude values (or some aggregated forms) in the computation of  $g$ , as input of a predictive machine learning model  $h$ , can obviously improve the performance. However, the resulting classifier  $f$  will be tailored for a specific geographic position. On the other hand, not considering directly this raw information allows the classifier  $f$  to be trained on a geographical area and applied in a different one. We indicate TIF variants also using latitude and longitude with TIF\*.

## 4 EXPERIMENTS

We experimented with TIF on five real datasets<sup>2</sup> for GPS trajectory classification with different sizes, semantics, and classification objectives. For *animals* the classification task consists of recognizing three different species. For *vehicles* the objective is to distinguish

between buses and trucks. For *seabirds* the objective is recognizing the flying trajectories of three species of seabirds. For *geolife*, we set the classification problem to recognize trajectories of public vs private means of transport. Finally, for *taxi*, we consider one month of observation. We aim to distinguish different types of taxi calls, i.e.,  $A$  if the trip was dispatched from the central,  $B$  if the trip was demanded directly to a taxi driver on a specific stand, and  $C$  otherwise. We divided each dataset into training and test sets with a ratio of 70-30. We compare TIF against various state-of-the-art baselines and competitors: ROCKET [8], CNN [7], CIF [17], GEOLET [13] and MOVELETS [11]. Hyperparameter details are provided in the implementation code. We performed a sensitivity analysis of the method as the hyperparameters changed. The results suggest setting a low value for  $|K|$ , adopting the Euclidean distance to save time, and searching for the best values for the interval lengths *min length*, *max length*. Once *min length*, *max length* are selected, it is possible to increase the number of intervals  $|K|$ . Finally, to maximize the performance, we suggest including the choice of the filling strategy in the hyper-parameter tuning phase.

### 4.1 Classification Performance

For TIF we report the accuracy and runtime for the four types of intervals considered, i.e., TIF-O, TIF-S, TIF-T, and TIF-A, as well as for the setting in which the raw latitude, longitude and time are used in the point-based features  $\mathcal{P}$ , denoted with the symbol \*.

Table 2 shows the average accuracy and runtime ( $\pm$  std. dev.) across 10 executions for the five public datasets under analysis. We report the results obtained for the best hyper-parameters setting for every method after a grid search (including filling strategies), as described in the previous subsection. If a method does not terminate within 3 hours or exceeds the available memory, we report it in the table by using the symbol “-”.

We immediately notice that one of the TIF variants is always in the first or second position for accuracy and runtime. A TIF\* variant has the highest accuracy on four datasets out of five, while TIF variants have the highest or second highest accuracy on four

<sup>2</sup>animals and vehicles: <https://t.ly/keXzn>, seabirds: <https://t.ly/fhCnR>, geolife: <https://t.ly/6VJ-E>, taxi: <https://t.ly/0GMR9>.

**Table 2: Average accuracy, runtime, and standard deviations across 10 repetitions of the experiments using the best hyperparameter configurations. Best values in bold, best values runner up in italic.**

		ROCKET	GEOLET	MOVELETS	CNN	CIF	TIF-o	TIF-T	TIF-s	TIF-A	TIF-o*	TIF-T*	TIF-s*	TIF-A*
Accuracy (→)	animals	.871±.00	.935±.00	.563±.00	.477±.06	.971±.00	.960±.02	.982±.03	.941±.02	.989±.02	<b>1.00±.00</b>	<b>1.00±.00</b>	.996±.01	.984±.03
	vehicles	.928±.00	.965±.00	.921±.00	.750±.01	.939±.00	.965±.00	<b>.987±.00</b>	.965±.01	.985±.00	.972±.01	<b>.987±.01</b>	.962±.01	.983±.01
	seabirds	.667±.00	<b>.967±.00</b>	-	.594±.05	.700±.02	.773±.02	.788±.04	.697±.04	.773±.02	.833±.02	.828±.06	.788±.06	<i>.859±.01</i>
	geolife	.733±.00	.861±.00	-	.721±.012	-	<b>.918±.00</b>	.893±.01	.903±.00	.916±.01	.916±.00	.900±.00	.904±.01	<b>.918±.00</b>
	taxi	.566±.00	.578±.00	-	.524±.01	-	.538±.01	.565±.00	.543±.01	.597±.01	.633±.01	.775±.00	.732±.00	<b>.839±.00</b>
Runtime (←)	animals	2.2s±.00	21.3s±.00	25.7s±.00	1.3s±.07	7.5s±.97	1.5s±.02	<b>0.6s±.01</b>	3.5s±.04	7.0s±.04	1.7s±.09	1.2s±.06	3.6s±.06	7.0s±.21
	vehicles	31.5s±.00	50.1s±.00	141m±.00	5.3s±.69	34.6s±.59	5.8s±.04	<b>1.6s±.08</b>	9.2s±.14	18.7±.85	5.8s±.04	17.3s±.52	9.2s±.14	15.2s±.52
	seabirds	15.7s±.00	48m±.00	-	11.8s±8.37	105.8s±.02	1.0s±3.27	3.3s±.39	23.9s±2.97	29.1±3.17	1.1s±.06	<b>0.8s±.03</b>	29.6s±3.80	32.7s±3.23
	geolife	29.1m±.00	145.2m±.00	-	27.2s±7.37	-	11.2s±.17	<b>10.5s±.02</b>	15.4s± 0.15	38.7±.03	14.8s±.04	13.0s±.00	15.9s±.05	46.0s±.01
	taxi	13.3m±.00	44m±.00	-	<b>3.2m±.41</b>	-	3.29m±.01	3.33m±.00	3.42m±.01	8.41m±.14	10.9m±.05	7.48m±.03	6.36m±.19	20.63m±.34

datasets out of four. Including the raw data, TIF\* reaches substantially higher scores than the state-of-the-art, while the accuracy of TIF is aligned with one of the competitors. Concerning runtime, with the exception of CNN on taxi, TIF variants are markedly faster than state-of-the-art approaches. Therefore, TIF provides the highest accuracies and the lowest runtimes.

However, despite the slightly higher runtime, among the variants of our proposal, we can observe that all TIF\* versions achieve the highest accuracy. Similarly, among TIF variants, TIF-A has the highest accuracy. Hence, using an ensemble of intervals defined heterogeneously w.r.t. observations, space, and time increases the predictive performance of the RF. However, for animals and vehicles, the best accuracy is achieved by TIF-o\*/TIF-o and TIF-T\*/TIF-T, probably due to the small size of these datasets.

Given the above results, we recommend relying on TIF-A\* if runtime is not an issue and if the model must not be geographically transferred on TIF-A otherwise.

## 5 CONCLUSIONS

We have presented TIF, an interval-based approach to GPS trajectory classification that transforms the spatio-temporal data into a simplified feature-based representation that can be used as input to any classifier. TIF collects, homogenizes, and implements efficiently the most relevant features defined in the field in the last 20 years. Plus, in TIF, we proposed and evaluated some strategies to handle the different lengths and non-constant sampling-rate problems. In the field of mobility data analysis, these two problems are rarely addressed. Finally, we show that TIF outperforms the competitors in terms of runtime while reaching their accuracy.

TIF can be extended to include features about the nearby moving entities, i.e., features that can capture some of the environment characteristics in which the object is moving. Inspired by HIVE-COTE [15] in the time series domain, we would like to integrate TIF and other competitors based on different trajectories representations to create a trajectory-specialized ensemble classifier. Finally, we would like to extend the capabilities of TIF by extending it to support to Multiple Aspect Trajectories [12].

## ACKNOWLEDGMENTS

This work is partially supported by the EU NextGenerationEU programme under the funding schemes PNRR-PE-AI FAIR (Future

Artificial Intelligence Research), PNRR-“SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics” - Prot. IR0000013, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, G.A. 761758 *Humane AI*, and G.A. 952215.

## REFERENCES

- [1] Paulo JAL Almeida et al. 2010. Indices of movement behaviour: conceptual background, effects of scale and location errors. *Zoologia* 27 (2010), 674–680.
- [2] Anthony J. Bagnall et al. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *DAMI* 31, 3 (2017), 606–660.
- [3] Adel Bolbol et al. 2012. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. *CEUS* 36, 6 (2012), 526–537.
- [4] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (2001), 5–32.
- [5] Camila Leite da Silva et al. 2019. A Survey and Comparison of Trajectory Classification Methods. In *BRACIS*. IEEE, 788–793.
- [6] Sina Dabiri et al. 2020. Semi-Supervised Deep Learning Approach for Transportation Mode Identification Using Trajectory. *IEEE TKDE* 32, 5 (2020), 1010.
- [7] Sina Dabiri and Kevin Heaslip. 2018. Inferring transportation modes from GPS trajectories using a convolutional neural network. *CoRR* (2018), 360–371.
- [8] Angus Dempster et al. 2020. ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *DAMI* 34, 5 (2020), 1454–1495.
- [9] Somayeh Dodge et al. 2009. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *CEUS* 33, 6 (2009), 419–434.
- [10] Mohammad Etemad et al. 2018. Predicting Transportation Modes of GPS Trajectories Using Feature Engineering and Noise Removal. In *CCAI*. Springer, 259–264.
- [11] Carlos Andres Ferrero et al. 2018. MOVELETS: exploring relevant subtrajectories for robust trajectory classification. In *SAC*. ACM, 849–856.
- [12] Carlos Andres Ferrero et al. 2020. MasterMovelets: discovering heterogeneous movelets for multiple aspect trajectory classification. *DAMI* 34, 3 (2020), 652–680.
- [13] Cristiano Landi et al. 2023. Geolet: An Interpretable Model for Trajectory Classification. In *IDA (Lecture Notes in Computer Science, Vol. 13876)*. Springer, 236–248.
- [14] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. 2008. *TraClass*: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proc. VLDB Endow.* 1, 1 (2008), 1081–1094.
- [15] Jason Lines et al. 2016. HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles for Time Series Classification. In *ICDM*. IEEE Computer Society, 1041–1046.
- [16] Carl H. Lubba et al. 2019. catch22: CAnonical Time-series CHaracteristics - Selected through comparative time-series analysis. *DAMI* 33, 6 (2019), 1821.
- [17] Matthew Middlehurst et al. 2020. The Canonical Interval Forest (CIF) Classifier for Time Series Classification. In *IEEE BigData*. IEEE, 188–195.
- [18] Farid Movahedi Naini et al. 2016. Where You Are Is Who You Are: User Identification by Matching Statistics. *IEEE Trans. Inf. Fore. Secur.* 11, 2 (2016), 358–372.
- [19] Sasank Reddy et al. 2008. Determining transportation mode on mobile phones. In *ISWC*. IEEE Computer Society, 25–28.
- [20] Zhibin Xiao et al. 2017. Identifying Different Transportation Modes from Trajectory Data Using Tree-Based Ensemble Classifiers. *IJGI* 6, 2 (2017), 57.
- [21] Yu Zheng et al. 2008. Learning transportation mode from raw gps data for geographic applications on the web. In *WWW*. ACM, 247–256.
- [22] Yu Zheng et al. 2008. Understanding mobility based on GPS data. In *UbiComp (ACM International Conference Proceeding Series, Vol. 344)*. ACM, 312–321.