Trajectory Outlier Detection: Algorithms, Taxonomies, Evaluation, and Open Challenges

ASMA BELHADI, Kristiania University College
YOUCEF DJENOURI, SINTEF Digital
JERRY CHUN-WEI LIN, Western Norway University of Applied Sciences
ALBERTO CANO, Virginia Commonwealth University

Detecting abnormal trajectories is an important task in research and industrial applications, which has attracted considerable attention in recent decades. This work studies the existing trajectory outlier detection algorithms in different industrial domains and applications, including maritime, smart urban transportation, video surveillance, and climate change domains. First, we review several algorithms for trajectory outlier detection. Second, different taxonomies are proposed regarding application-, output-, and algorithm-based levels. Third, evaluation of 10 trajectory outlier detection algorithms is performed on small, large, and big trajectory databases. Finally, future challenges and open issues with regard to trajectory outliers are derived and discussed. This survey offers a general overview of existing trajectory outlier detection algorithms in industrial informatics applications. As a result, mature solutions may be further developed by data mining and machine learning communities.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Anomaly detection**; • **Information systems** → **Data mining**;

Additional Key Words and Phrases: Trajectory outlier detection, industrial informatics applications, data mining, machine learning

ACM Reference format:
Asma Belhadi, Youcef Djenouri, Jerry Chun-Wei Lin, and Alberto Cano. 2020. Trajectory Outlier Detection: Algorithms, Taxonomies, Evaluation, and Open Challenges. *ACM Trans. Manage. Inf. Syst.* 11, 3, Article 16 (June 2020), 29 pages.
https://doi.org/10.1145/3399631

1 INTRODUCTION

Machine learning offers emergent solutions to many industrial systems, such as transportation [93], manufacturing [118], video surveillance [78], climate change [49], and networking [107]. Anomaly detection is one of the main machine learning techniques intensively applied in

Authors' addresses: A. Belhadi, Department of Technology, Kristiania University Colleage, Kirkegata 24, 0153 Oslo, Norway; email: asma.belhadi.17@gmail.com; Y. Djenouri, Department of Mathematics and Cybernetics, SINTEF Digital, Forskningsveien 1, 0373 Oslo, Norway; email: youcef.djenouri@sintef.no; J. C.-W. Lin, Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Inndalsveien 28, 5063 Bergen, Norway; email: jerrylin@ieee.org; A. Cano, Department of Computer Science, Virginia Commonwealth University, 401 W. Main St, Richmond, VA, 23284, USA; email: acano@vcu.edu.

2158-656X/2020/06-ART16 $15.00
https://doi.org/10.1145/3399631

industrial informatics [28, 126, 139]. An anomaly is defined as an abnormal observation that appears to be inconsistent with the rest of the data [105]. Anomaly detection techniques have been dealt with different data representation, including points [56], trajectories [142], time series [12], and spatio-temporal information [38, 67].

There is a considerable number of existing published studies and surveys on trajectory outlier detection, especially in different industrial applications including maritime, smart urban transportation, video surveillance, and networking domains. After several decades of theoretical development, a significant number of new technologies and applications have appeared that analyze the trajectories in video surveillance domains and applications. This article presents a comprehensive and systematical survey of trajectory analysis in the video surveillance field. Here we attempt to find a clearer way to present the concepts and practical aspects of video surveillance for the trajectory analysis research community. The main contributions of this work are as follows:

- This survey gives a general overview of existing trajectory outlier detection algorithms in industrial informatics applications.
- We provide a comprehensive review and discussion of several algorithms for trajectory outlier detection.
- We introduce different taxonomies regarding to application-, output-, and algorithm-based levels.
- We discuss arising challenges and open issues in trajectory outlier detection.
- We explore future research lines that may be further developed by the data mining and machine learning community.

1.1 Review Works

This section reviews the relevant survey papers in the area and defines the new contributions of this work. Schubert et al. [111] offered a unified notion of locality in the existing outlier detection techniques based on neighborhood computation. They helped design more sophisticated advanced methods handled with specified industrial applications. Chalapathy and Chawla [25] reviewed the existing deep learning algorithms for anomaly detection such as the one-class neural network, deep hybrid models, supervised deep anomaly detection, semi-supervised anomaly detection, and unsupervised deep anomaly detection. Fernandes et al. [48] reviewed the existing anomaly detection algorithms under four dimensions: network traffic anomalies, network data types, intrusion detection systems categories, and detection methods and systems. Zheng [142] provided different representation of trajectories including sequences, matrices, graphs, and tensors, and different preprocessing tasks such as noise filtering, map matching, and compression. It divided solutions to trajectory outlier detection on anomalous trajectories and sub-trajectories, finding noise points in the whole set of trajectories, and the identification of anomalous events in trajectories. Feng and Zhu [47] proposed a general framework of trajectory data mining embedded on several layers including preprocessing, data management, query processing, trajectory data mining tasks, and privacy protection. Their framework with different levels of abstraction helped in better understanding of the existing solutions regarding trajectory mining. Gupta et al. [54] provided an interesting survey that discussed the techniques for the detection of temporal outliers. Their survey organized a discussion about different types of data, presented various outlier definitions, and discussed various applications for which temporal outlier techniques have been successfully employed (environmental sensor networks, trajectory, biological, astronomy, and web data).

Zheng et al. [143] presented a general urban computing framework, which was composed of four steps: urban sensing, urban data management, data analytics, and service providing. In the first step, urban sensing aims to capture people's mobility using GPS sensors or their mobile phone signals. In the second step, urban data management employs powerful indexing structures to store the spatio-temporal information obtained in the first step. In the third step, data analytics are able to identify and extract useful patterns, such as clusters and outliers, benefiting from the indexing structures. In the fourth step, the service providing goal is to interpret the obtained information and send it to the city planner authority for dispersing and diagnosing anomalies. Bhowmick and Narvekar [18] classified the existing trajectory outlier detection approaches according to the method used in the processing step: distance-, density-, and motifs-based outliers. Djenouri and Zimek [43] sketched some existing urban traffic flow outlier detection algorithms by analyzing locality notion proposed in the work of Schubert et al. [111], whereas other works [14, 39] review existing urban traffic outlier detection methods, including different representations such as flow values, segment flow values, and trajectory and sub-trajectory outliers. The presented solutions were limited to the intelligent transportation community. Another work providing intensive comprehensive study of existing data mining and machine learning solutions for intelligent transportation analysis was that of Alsrehin et al. [6]. Meng et al. [91] analyzed the existing trajectory outlier detection regarding two dimensions. The first dimension is the key features used to retrieve the trajectory outliers, such as speed, direction, position, and time. The second dimension is the distance used to measure the divergence among trajectories. Habeeb et al. [55] addressed the issue of detecting anomalies in real time and reviewed the existing real-time big data processing technologies related to anomaly detection, such as Spark, Hadoop, Storm, and Hafka. Xiao et al. [130] reviewed existing pattern mining techniques for trajectory anomaly detection and forecasting on maritime traffic data. The study affirms the importance of proposing advanced data mining techniques to deal with maritime traffic research challenges. Atluri et al. [9] discussed different types of pattern and motifs to solve trajectory outliers on both spatial and temporal dimensions. Chandola et al. [27] provided a comprehensive and structured overview of the existing solutions to solve the sequence anomaly detection problem and consider trajectories as a case study of their study. Ahmed et al. [4] presented an in-depth survey of various pattern mining–based anomaly detection techniques and compared them using synthetic financial data. Gogoi et al. [53] presented a survey of well-known pattern mining–based outlier detection and compared them in both supervised and unsupervised environments.

Unsupervised learning is a methodology that can be utilized for outlier detection. It can thus be easily organized into three categories [26] as the research issues in recent decades: the reconstruction-based model [22, 146], clustering-based analysis model [11, 63, 151], and one-class classification-based model [32, 116, 127]. In reconstruction-based models, they conduct anomaly analysis from a single aspect; the performance is very limited, and it always produces the reconstruction error. Furthermore, the limitations of traditional clustering-based models include it being hard to handle the multi- or high- dimensional data for outlier detection. Dimensionality reduction was presented to solve this limitation [26], but this method does not require the guidance from the subsequent clustering analysis; the major information for clustering may be lost during the reduction progress. Furthermore, in the one-class classification approaches, a discriminative boundary surrounding the normal instances is learned for outlier detection. However, when the size of the dimensions highly increases, this one-class classification model always suffers the limitation of suboptimal performance.

With the rapid growth of computer vision and multimedia databases (i.e., surveillance data in video scenes [58, 96]), it is also an important issue to analyze the trajectories of the video surveillance domain. In this surveillance domain application, trajectory can be used in automatic visual surveillance [2], traffic management [98], suspicious activity detection [129], sports video analysis [106], video summarization [140], synopsis generation [101], and video-to-text descriptors [102], among others. To analyze the trajectory patterns, clustering, abnormal detection, and

Table 1. Related Review Papers on Trajectory Outlier Detection

Reference	Trajectory Data	Applications	Methods
Chalapathy & Chawla [25]	No	No	Deep learning
Schubert et al. [111]	No	No	Local anomaly detection
Fernandes et al. [48]	No	Network traffic data	Statistical, distance, and evolutionary
Zheng [142]	Yes	Transportation	Data mining
Gupta et al. [54]	Yes	No	Temporal data mining
Zheng et al. [143]	Yes	Urban traffic data	Cluster based
Bhowmick & Narvekar [18]	Yes	No	Distance, density, and motifs
Djenouri & Zimek [43]	No	Urban traffic data	Statistical, distance, and pattern mining
Djenouri et al. [39]	Yes	Urban traffic data	Online and offline processing
Alsrehin et al. [6]	No	Intelligent transportation	Data mining and machine learning
Meng et al. [91]	Yes	No	Distance based
Habeeb et al. [55]	No	No	Big data technologies
Xiao et al. [130]	Yes	Maritime traffic data	Pattern mining
Atluri et al. [9]	Yes	Spatio-temporal data	Pattern mining
Chandola et al. [27]	Yes	No	Pattern mining
Ahmed et al. [4]	No	Financial data	Pattern mining
Gogoi et al. [53]	Yes	Network-based data	Pattern mining
Our work	Yes	Several applications	Distance, density, pattern mining, and other machine learning solutions

summarization have become major research topics in recent decades. It is not a trivial task for trajectory clustering [21, 61, 137] due to several limitations: noise data may influence the tracking results, large variation in movement patterns may occur, and the size of the surveillance video is often very large, and thus many trajectories should be analyzed. Detecting suspicious or abnormal activity has also been an interesting topic in recent years [121]. The challenge of this topic is to define abnormality or anomaly since it is very subjective and context dependent. Furthermore, most parts of surveillance videos are not of much use other than to summarize or shorten the meaningful parts in the videos [90], which is another challenging task in analyzing the trajectories in the surveillance videos.

Table 1 summarizes some features of the existing survey papers. Compared to the existing surveys, this article investigates comprehensive works with different domains and applications of trajectory outlier detection. All of the other works were limited to only some applications, such as urban traffic data, some output such as trajectory outliers, or some algorithms such as distance- or density-based solutions. The main motivation of this research work is to provide a comprehensive review and comparative study of trajectory outlier detection proposed in several real-world applications and with regard to the outputs. This survey organizes the existing solutions into different taxonomies: applications based (intelligent transportation, video analysis, climate change), output based (trajectory and sub-trajectory), and algorithm based (distance, density, pattern mining, etc.).

1.2 Basic Concepts

This section introduces fundamental definitions for trajectory outlier detection. A trajectory is an ordered sequence of location points in space. We will denote by pt a single spatial location point, where each pt is a tuple of two values—the latitude and the longitude of this location.

Definition 1.1 (Trajectory Database). We define a trajectory database $T = \{T_1, T_2, \ldots, T_m\}$, where each raw trajectory T_i is a sequence of spatial location points $\{pt_{i1}, pt_{i2}, \ldots, pt_{in}\}$.

As is common in the literature [39], the location points that are similar enough are aggregated into regions. Let us denote by R a location *region* in space.

Definition 1.2 (Mapped Trajectory Database). We define a mapped trajectory database $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\}$, where each mapped trajectory Λ_i is a sequence of spatial location regions $\{R_{i1}, R_{i2}, \ldots, R_{in}\}$, obtained by replacing each point pt_{ik} in T_i with its region R_{ik}.

We define the dissimilarity between any two trajectories as the distance between them.

Definition 1.3 (Trajectory Dissimilarity). The similarity between two trajectories Λ_i and Λ_j, denoted $d(\Lambda_i, \Lambda_j)$, is defined by the symmetric difference—that is, the number of all regions in the two trajectories minus the number of shared regions in the two trajectories. Formally,

$$d(\Lambda_i, \Lambda_j) = n - \{|(R_{il}, R_{jl})| \ \forall l \in [1 \ldots n]\}. \tag{1}$$

Definition 1.4 (kNN). We define kNN of a trajectory Λ_i, denoted $k\text{NN}(\Lambda_i)$, as

$$k\text{NN}(\Lambda_i) = \{\Lambda_j \in \Lambda \setminus \{\Lambda_i\} \mid d(\Lambda_i, \Lambda_j) \leq k_{dist}(\Lambda_i)\}, \tag{2}$$

where $k_{dist}(\Lambda_i) = d(\Lambda_i, \Lambda_l)$ *is the k-distance of the trajectory* Λ_i *defined such as it exists* k trajectories $\Lambda' \in \Lambda$, *it holds that* $d(\Lambda_i, \Lambda_l) \geq d(\Lambda_i, \Lambda')$.

Now we are ready to formally define the trajectory outlier detection problem as follows.

Definition 1.5 (Trajectory Outlier Detection Problem). We define the set of the first p trajectory outliers using kNN, *denoted* $\mathcal{G}^+ = \{\Lambda_1^+, \Lambda_2^+, \ldots, \Lambda_p^+\}$, by

$$\mathcal{G}^+ = \{\Lambda_i^+ | \forall j \in \Lambda \setminus \mathcal{G}^+, k\text{NN}(\Lambda_i) \geq k\text{NN}(\Lambda_j)\}. \tag{3}$$

Moreover, most solutions presented throughout the article are composed of four steps. The first step is trajectory collection, where trajectories are harvested from different sources, including environmental sources such as sensors, archive sources such as events log databases, or other data sources. The trajectories collected are stored in a single database. The second step is preprocessing the trajectories using machine learning techniques. This step includes (1) trajectory enrichment, such as adding statistical values like the mean of the trajectory samples and standard deviation; (2) trajectory cleaning, such as that of textual data (Natural language processing (NLP) techniques [86] are a typical example of cleaning methods that could be used in this stage); (3) selection of the appropriate features from all of the trajectories' data, which depends on the application used in the mining process (Principal component analysis (PCA) [22] is one of the most used methods for dimensionality reduction); and (4) normalization of trajectory data, which is needed for some outlier detection operators, such as similarity and density computation. The third step is anomaly detection, where outlier detection techniques are used to derive trajectory outliers. The fourth step is interpretation of the trajectory outliers extracted from the previous step, which largely depends upon the application used.

Outline. The rest of the article is organized as follows. Section 2 presents trajectory outlier detection algorithms ordered chronologically. In Section 3, we provide our insight on the reviewed papers and propose three taxonomies of the existing trajectory outlier detection algorithms. Section 4 shows the performance evaluation of 10 existing trajectory outlier detections on small, large, and big databases. Section 5 provides open research issues in the trajectory outlier detection field. Section 6 presents our conclusions.

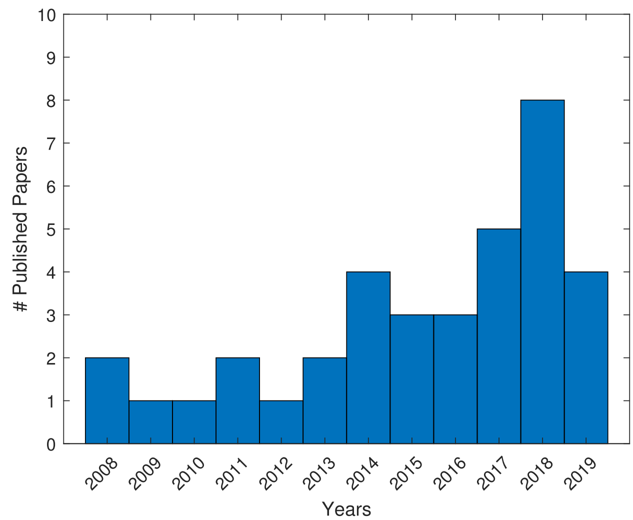


Fig. 1. Number of published papers on trajectory outlier detection from 2008 to 2019.

2 ALGORITHMS

Figure 1 presents the number of published papers on trajectory outlier detection from 2008 to 2019. We observe from this figure that there has been increasing research interest in this field over the past 3 years. Therefore, this section presents an overview of trajectory outlier detection algorithms ordered chronologically.

2008–2013. Lee et al. [71] introduced an angular model for retrieving sub-trajectory outliers. The authors developed the TRAOD (TRAjectory Outlier Detection) algorithm by investigating the *partition and detect* strategy in the mining of sub-trajectory outliers. In the partition step, each trajectory is partitioned into the smallest meaningful units, named *t-partitions*. In the detection step, the score of each *t-partition* is determined, which is based on the sum of densities of the *t-partitions* of all trajectories observed during the same time window as this *t-partition*. The *t-partitions* whose scores exceed 1 are considered as outliers. Piciarelli et al. [100] introduced a support vector machine (SVM) to detect trajectory anomalies. The main principle of this approach is to group the set of trajectories sharing many features into similar clusters using an SVM approach. A new trajectory is tested for anomaly by comparing it with the cluster model instead of exploring the entire trajectories. Ying et al. [133] proposed clustering trajectory outlier detection (CTOD). A new similarity measure between the *t-partitions* using minimal bounding boxes (MBBs) is introduced by incorporating both spatial and temporal dimensions. The DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [46] is then applied on the *t-partitions* of the whole trajectories. The *t-partitions* of the cluster of the lower density are considered as outliers. Similar to TRAOD, this approach allows to identify both trajectory and sub-trajectory outliers; however, it is not straightforward to determine the *t-partitions*. Ge et al. [51] developed a TOP-EYE algorithm for identifying direction-based trajectory outliers. The monitoring area is partitioned into grids, each of which is further divided into eight directions with $\frac{\pi}{4}$ of an angle range for each direction. Each grid is represented by a vector of eight density values; the i^{th} density value is defined by the number of trajectories passed in this grid and have the direction along the i^{th} direction. The outlying score of each trajectory is calculated by summing the density values of each direction in each grid passed by this trajectory. Only trajectories with a score less than a given threshold are considered as outliers. The set of grids is the key feature of this approach; however, it is not straightforward to determine the grids in an accurate way. Masciari [89] exploits the lifting strategy to compress the whole trajectories on trajectories on a few representative points. Based on these lifted versions of trajectories, a Fourier transform is defined, and kNN is performed on the lifted trajectories to identify the trajectory outliers. This approach only takes a spatial dimension and ignores the temporal information of the entire trajectories. Zhang et al. [138] proposed the *iBAT* (isolation-Based Anomalous Trajectory) algorithm by exploring the "few and different" properties of anomalous trajectories. It adopts the *iForest* (isolation Forest) algorithm [77] by generating a random tree of trajectories. The set of trajectories is recursively divided until almost all of them are isolated. A shorter path of trajectories is derived. These trajectories are considered as outliers, since they are isolated faster than normal trajectories isolated in a longer path. Liu et al. [80] developed the RTOD (Relative distance-based Trajectory Outlier Detection) algorithm. It introduced the problem of absolute distance between trajectories, where only points participate in the trajectory similarity. In this issue, some trajectories have the same points but deviate in terms of segments. Thus, the trajectories are partitioned into several segments of consecutive points. A similarity between two different trajectories is based on the *Hausdorff* distance [30], estimated between their segments. To reduce the number of comparisons in Hausdorff, only neighbor segments are selected based on a local correlation matrix of the segments of all trajectories. This approach is fast, yet its accuracy is reduced by investigating an approximate Hausdorff function. Saleem et al. [108] introduced the RPAT (Road segment Partitioning towards Anomalous Trajectory detection) algorithm. The trajectories are partitioned into sub-trajectories on the basis of the road segments. The score of each sub-trajectory is independently processed using some features, including the speed, the flow rate, and the visited time. The score of each trajectory t is the sum of scores of all sub-trajectories belong to t. Trajectories with a score greater than a user-specified threshold are anomalous. An updating procedure is finally established to maintain the average speed, the average flow rate, and the average visited time of each road segment. This approach allows to identify both trajectory and sub-trajectory outliers, but it is difficult to determine the key attributes of representing the trajectories. Chen et al. [29] proposed the *iBOAT* (isolation-Based Online Anomalous Trajectory) algorithm to find anomalous taxi sub-trajectories. The points on the mapped new sub-trajectories that cause the outlierness are identified by using the adaptive working window strategy. The score of the sub-trajectory t is computed by the number of anomalous grids visited by t. g_i is an anomalous grid for the trajectory t, if the number of cell points of other trajectories in g_i is less than a given threshold. A general framework of *iBOAT* with several optimizations, such as an inverted mechanism file [152], is presented in the work of Sun et al. [117] to deal with a large number of trajectories in real time.

2014. Laxhammar and Falkman [70] presented the SHNN-CAD (Sequential Hausdorff Nearest Neighbours Conformal Anomaly Detector) algorithm. It is designed to identify trajectory outliers in an online environment. The training trajectories are first determined for each new trajectory t by selecting its kNN using the Hausdorff distance. Afterward, a conformal anomaly detector [69] is applied on the selected kNN trajectories to calculate the statistical confidence value of t. A probabilistic threshold ϵ is employed to discriminate normal or abnormal trajectories, whereas trajectories with probability lower than ϵ are anomalies. This approach only works for supervised environments. Yu et al. [136] developed PN-Outlier (Point-Neighbor-based trajectory Outlier), a neighborhood-based method for deriving sub-trajectory outliers. The point-based neighboring is determined for each point in each trajectory. The sub-trajectory outliers are ranked according to the number of neighbor points exceeding a specified value. Lan et al. [68] proposed an algorithm to identify taxi drive outliers. The taxi trajectories are mapped to the segments of the road network, where each road segment is associated with two numbers. The first number denotes the expected number of taxis passing this road segment at a specified time period, and the second number denotes the standard deviation of the number of taxis passing this road segment at the same time period. For a new traffic scenario, sub-trajectories passed with road segments that highly deviate from the expected value using the deviation value are identified as anomalous. Liu et al. [81] addressed the problem of incomplete data and proposed a new algorithm, named *BT-miner*, for retrieving anomalous trajectories from mobile phone data. Correlation-based clustering and an adaptive parameter-free detection method called *R-scan* are first applied to find anomalies on mobile phone data. Both spatial and temporal information have been exploited to generate trajectory outliers from the anomalous mobile phone time series data.

2015. Zhu et al. [147] proposed TPRO (Time-dependent Popular Routes-based trajectory Outlier detection). The temporal anomalies are retrieved based on the most frequent routes on each timestamp δt. A partitioning strategy is developed to split the trajectories into similar groups. The edit distance is then calculated during the interval time δt between the representative trajectory of each group and the top l popular roads of the given city. A threshold θ is used to distinguish the anomalous groups from the normal groups. This approach considers the temporal anomalous; however, it requires time-intensive computing in the preprocessing step. Ando et al. [7] developed the ACE (Anomaly Clustering Ensemble) algorithm. It is an ensemble for anomaly detection from multi-resolution trajectory features, where a clustering ensemble learning model is involved in different settings of scale and resolution parameters. A meta-feature represented by clusters of two trajectory (target and auxiliary) databases is extracted. The anomaly score is determined by first calculating the dissimilarity between the summarized features of both databases, then propagating the scores of the centroids to each trajectory database separately. This framework has been tested using two databases, but it could be generalized in a multi-trajectory database environment. Lin et al. [76] developed the iBDD (isolation-Based Disorientation Detection) algorithm that exploits the *few and different* strategy to detect disorientation of an ongoing trajectory from the historical trajectories. A symbolization procedure is established to filter noise points from all GPS trajectories. The anomalous degree of the ongoing trajectory t is determined by counting the number of all historical trajectories that support t; if this degree is greater than a user threshold, t is considered as normal, and otherwise it is viewed as a *few and different* trajectory and is counted as an outlier. These approaches suffer from the parameter settings, which isolate the normal trajectories from the abnormal trajectories, and it is not evident to set this user threshold.

2016. Banerjee et al. [10] introduced the problem of finding temporally anomalous sub-trajectories and proposed two algorithms: Naive solution and MANTRA (Maximal ANomalous sub-TRAjectories). Naive solution starts by enumerating all possible sub-trajectory candidates. For each sub-trajectory t, a gap of time between t and the other sub-trajectories regarding the same spatial information is determined. Maximal anomalous sub-trajectories are those having a time gap greater a given threshold. This naive solution is time consuming, particularly for a large number of points per trajectory. To address this issue, the MANTRA algorithm was proposed to recursively prune the search space into a set of *islands*. The main features of islands include that all maximal anomalous sub-trajectories reside only inside them, and thus the rest of the parts of the trajectory can be efficiently pruned. Lei [72] introduced the MT-MAD (Maritime Trajectory Modeling and Anomaly Detection) framework for the detection of anomalies in maritime trajectories. It presents three features that cause maritime anomalies, such as spatial, sequential, and behavioral features. The spatial feature is defined as a density-based feature of anomalies, where trajectories with a low-density region are considered as anomalous. The sequential feature is captured when a sequence of a snapshot of trajectories is highly changed, regardless of their historical spatial dimensions. The behavioral feature anomalies are defined if the behavior vector of the given trajectory represented by a speed and direction is dissimilar to those of neighboring trajectory streams. An aggregation of these three features is combined into a suspicious degree, where trajectories with a suspicious degree equal to or greater than the user-defined threshold are considered as anomalous. This approach only considers the spatial features of the trajectories; however, in real maritime scenario, a temporal dimension is also a key feature of the processed trajectories. Maiorano and Petrosino [85] developed an adapted ROSE (Rough Outlier Set Extraction) algorithm [5], which uses a rough set approach for finding the top outliers in unlabeled spatio-temporal data. The trajectories are first divided according to different time frames into several sub-trajectories. The ROSE algorithm is launched at each point of each sub-trajectory t to decide whether this point is anomalous or not. The outlier score of t is evaluated by the ratio between the number of anomalous points of t over the number of all points in t. If the score is greater than a threshold, t is labeled as an outlier.

2017. Yu et al. [134] developed two approaches—TN-Outlier (Trajectory-Neighbor based trajectory Outlier) and PN-Outlier (Point-Neighbor-based trajectory Outlier)—which are neighborhood-based methods for deriving sub-trajectory outliers. In PN-Outlier, the point-based neighboring is determined for each point in each trajectory. The sub-trajectory outliers are ranked according to the number of neighbor points exceeding a specified value. In TN-Outlier, the trajectory-based neighboring is determined for each sub-trajectory, where the score of sub-trajectories is determined by summing the neighborhood values of its points. These approaches allow to retrieve both trajectory and sub-trajectory outliers; however, both approaches only work for sparse trajectories with a large number of points shared among them. Luan et al. [83] introduced the LDTRAOD (Local Density TRAjectory Outlier Detection) algorithm, an improved version of the TRAOD algorithm. The trajectories are partitioned into several *t-partitions*, and the local outlier factor (LOF) algorithm is used on each *t-partition* of each tumbling window. Mao et al. [88] proposed the use of fragments and developed TF-Outlier (Trajectory Fragment Outlier) for finding sub-trajectory outliers, by considering each line segment of two consecutive points as a fragment. The LOF algorithm is used to determine the fragment outliers, where the local difference density of each segment is used rather than the local reachability density as the ratio between the number of neighbors of a current segment over the sum of the number of neighbors of the remaining segments. The local anomaly factor is then computed as an LOF value by replacing the local reachability density values with the local difference density values. Wu et al. [128] presented the DB-TOD (Driving Behavior-based Trajectory Outlier Detection) approach. The set of historical trajectories is first matched to the road network of the city according to the source and destination points. The probabilistic learning model described by the maximum entropy inverse reinforcement [149] is then used to transform the mapped trajectories into historical action trajectories. Thus, each road segment is regarded as a state, the different road decisions (e.g., turning left, turning right, or moving forward) are regarded as actions, and the drivers are considered as agents. Afterward, the learning model is launched to estimate the cost of historical trajectories. For a new sub-trajectory t, the score value is computed by the ratio of the cost of t and the sum of costs of all historical trajectories. If the obtained value is greater than a user threshold, then t is labeled as an outlier, and otherwise it is a normal sub-trajectory.

2018. Wang et al. [123] proposed the use of hierarchical clustering for deriving anomalous trajectories from taxi GPS data. First, all of the trajectories crossing the same source destination are extracted. Second, an edit distance algorithm is adopted to calculate the similarity between trajectories. An adaptive hierarchical clustering algorithm is finally performed to distinguish the anomalous trajectories from the regular one. Employing hierarchical clustering allows to identify different levels of trajectory outliers; however, the overall process needs a considerable amount of memory, particularly for very large number of trajectories. Pulshashi et al. [103] developed the kAA (k-ahead Artificial Arcs) strategy to improve the accuracy of the TRAOD algorithm [71] by smoothing the trajectories in identifying outliers. The trajectories are first transformed to the graph by only considering a single path from the starting point to the ending point. Artificial arcs are then created to the k nearest points from the current point. A Dijkstra algorithm [92] is finally applied to calculate the shortest path from the starting and the ending points of each trajectory, where a smoothed trajectory is a sequence of points belonging to the shortest path. The results on ship trajectory data reveal the improvement of kAA against TRAOD. This is because kAA recognizes more points compared to the *t-partitions* of TRAOD. However, the shortest path calculation needs a lot of time and is memory consuming. Kong et al. [64] proposed the LoTAD (Long-term Traffic Anomaly Detection) approach. This method consists of the following steps. The aim of the TS-segments creationstep is to create the TS-segments database from both the bus trajectory and the bus station line databases. A matrix of bus lines is created, where each element is a couple denoting the average velocity and average stop time at a given road segment during the specified time slot. The average velocity of each TS-segment is computed based on the velocity of all trajectories that belong to the same TS-segment and the weight coefficient assigned to each road segment. The average stop time is calculated based on the number of points in the trajectory and the stop time in each point. The anomaly index is determined using the Manhattan distance [95] between each two road segments, where the LOF is applied to find the anomalous road segments. Afterward, the k-means algorithm [84] is employed to find similar regions, and the score of each region is obtained by the sum of all LOF values of all road segments belonging on it, where only the top anomalous regions are returned. These approaches give good results in terms of accuracy by employing the LOF in the entire process. However, it is not straightforward to determine the *t-partitions* and the TS-segments. Zhang et al. [141] presented a stream computing framework for detecting ship trajectory outliers in real time and reminding other ships to take avoidance measures. First, the F-DBSCAN algorithm is performed on the historical trajectory database to extract the relevant features of each data ship point and partition the trajectories by the DBSCAN approach. Second, the outlier detection problem is transformed to the binary classification problem, where the batch computing model represented by SVMs, decision trees, and random forest is applied to find anomalies in a streaming way. Zhu et al. [148] presented STN-Outlier (Sub-trajectory and Trajectory Neighbor-based Outlier), in which a set of relevant features is extracted from the trajectory database. Thus, two feature extraction strategies have been investigated: (1) the intra-trajectory feature, which is characterized by the differences between points of two successive trajectories, and (2) the inter-trajectory feature, which indicates the spatial and directional differences of two trajectories. These two features are entered in the distance computation for finding sub-trajectory outliers based on neighborhood computation in a streaming way. This approach suffers from time-intensive calculation in determining the dissimilarity among the trajectories. Yu et al. [135] presented TODCSS (Trajectory Outlier Detection based on Common Slices Sub-sequences). It generates a set of slices by connecting consecutive line segments in each trajectory having the same direction. A slice is considered as noise if the number of its neighbors is less than the given threshold. A slice neighborhood is derived based on similarity, which is featured by the number of common segments between slices. This approach finds sub-trajectory outliers; however, it is difficult to generate the set of slices from the line segments. Ying et al. [132] proposed the use of the ANPR (Automatic Number-Plate Recognition) system instead of GPS devices to derive trajectory outliers. The approach follows three stages. The first stage is abstraction, which aims to abstract the regions of interest from a large number of ANPR records. The second stage is detection, where the goal is to determine the similarity between trajectories using the adjusting longest common weighted subsequence. The third stage is classification, which is designed to classify the trajectories into anomalous and normal trajectories according to the similarity measure calculated in the previous stage. The use of ANPR rather than GPS devices allows high accuracy in trajectory classification and low costs of system maintenance. Qin et al. [104] considered driving behavior and road network constraints and used Dempster-Shafer evidence theory by developing the DS-Traj (Dempster-Shafer for Trajectory outliers) algorithm. Features related to trajectories are first extracted, such as route selection, intersection rate, heading change rate, slow point rate, and velocity change rate. The probability value is assigned to each feature using anomalous evidence. The anomalous trajectories are extracted using the Dempster-Shafer theory [36], where the anomalous decision rules are found. The use of Dempster-Shafer theory allows to provide fuse evidence without prior knowledge.

2019. Román et al. [109] developed the CaD (Context-aware Distance)-based algorithm to identify anomalous trajectories from human trajectories. The algorithm starts by selecting the best sub-trajectories representing each trajectory using the TSA (Trend Segmentation Algorithm) [113]. The selected sub-trajectories are clustered using PAM (Partition Around Medoids) [97] and context aware similarity. The outlier detection is finally performed by computing the *k*NN score (with $k = 1$) by selecting the reference set, the sub-trajectories belonging to the same cluster. This approach suffers from its accuracy, in which only the 1-NN is applied in finding the trajectory outliers. Huang and Zhang [60] developed a hybrid *k*NN and LOF for finding anomalous ship trajectories. The ship's candidates are first extracted by using the *k*NN model. The local reachability distance is then calculated for each ship candidate to find out anomalies. This algorithm is able to deal with large ships' maritime data by reducing the search space to only the *k* nearest neighbors. These approaches suffer from the parameter settings by incorporating the DBSCAN, the LOF, the *k*NN, and the supervised classification algorithms in finding the trajectory outliers. Oehling and Barry [99] presented an adapted LOF for airline flights called *LoOP-AF* (Local Outlier Probability for Airline Flights). First, the airline flight trajectory data are collected and cleaned by removing flight data errors. Typical flight data errors are synchronization errors and cycling. A synchronization error occurs when some parts of the data are not recorded due to temporary recorder failures, electrical transients, or other issues. Second, LoOP [65] is applied on the extracted trajectories. Third, an expert validates the abnormal trajectories. It is hard to distinguish between the flight data errors that were considered as noises and the data trajectory abnormalities. Bouindour et al. [20] developed a new method for detecting abnormal events in videos data streams, represented by the trajectory of human behaviors in an open urban space. A modified convolutional neural network [66] is integrated with the feature selection to select the relevant feature maps related to shapes and motions. This allows detection of abnormal events in non-crowded and crowded scenes. This approach is able to localize all redundant and rare events during the learning process. Moreover, it is able to derive new abnormal events during the testing process. Chu et al. [33] introduced the use of an unsupervised deep feature learning model for abnormal event detection from video stream trajectory data by developing a new approach called *SCG-SFL* (Sparse Coding Guided Spatiotemporal Feature Learning). Spatio-temporal features are selected using a deep three-dimensional convolutional network. The machine learning approaches are accurate when the ground truth is provided. These approaches could not be applied with unlabeled trajectories, where unsupervised learning should be applied for identifying the abnormal trajectories.

3 TAXONOMIES AND DISCUSSIONS

From this literature review, different ways have been proposed for finding trajectory outliers. In the following, we provide our insights of the reviewed papers.

First, some trajectory outlier detection approaches are based on neighborhood computation, and these approaches use efficient structures for computing the similarity between trajectories. The accuracy of such approaches is low compared to the other approaches, where the judgment of anomalies is provided locally based on the neighborhood trajectories. Another issue with such methods is the way of determining the similarity between trajectories: some methods use adapted Euclidean metrics, whereas other methods use sophisticated metrics (e.g., angular based). Consequently, a normalization step is needed to fairly compare these methods.

Second, some approaches consider clustering techniques as a preprocessing step to find trajectory outliers. One possible way to do this is to establish the density-based paradigm [110], where normal trajectories need not belong to any noise clusters. Each trajectory in these noise clusters could be interpreted as anomalous. Some of existing trajectory outlier detection approaches define clusters and do not consider trajectories not belonging to any cluster as outliers. These approaches perform well on trajectories with a low number of data points; however, with high-dimensional data, they require high computational time processing.

Third, some approaches explore pattern mining techniques to find trajectory outliers. These approaches first transform the trajectory database into a transactional database, apply the pattern mining process, and then derive outliers using some interestingness metrics. These approaches perform well in terms of accuracy, but they are time consuming because they adopted traditional models [3] for finding relevant patterns in trajectory data. Adopting emerging pattern mining models [13, 16, 40, 41] may improve the performance of such methods. Proposing specified pattern mining algorithms with specified interestingness metrics for the trajectory data is another way to address this issue.

Fourth, approaches based on machine learning techniques such as SVM, ensemble learning, and granular computing are efficient for some applications, where training trajectory data with ground truth is provided. However, in almost all cases, it is not possible to have the historical trajectory data for learning these models. One way to solve this issue is to build labeled trajectory data manually, which is a computationally costly task. In some cases, such as ship data, an expert is needed to judge if the labeled trajectory data are well created or not. In addition, the way of building labeled trajectory data influences the final accuracy of such methods. Thinking about standard training trajectory data is a critical issue with regard to these methods.

Fifth, an evaluation score of the existing trajectory outlier detection algorithms allows to decide whether a trajectory is anomalous or not. For many trajectory outlier detection approaches, particularly when using trajectories with a high number of data points, the outlier score is not really explainable. The scores of the trajectory outlier detection methods widely vary in terms of their scale. Thus, considering two approaches A_1 and A_2 and a given trajectory Λ_i, we can have this situation: score(A_1, Λ_i) >>> score(A_2, Λ_i). Regarding this scenario, it could be difficult to understand the meaning of the outlier score of some approaches, and also it will be not straightforward to compare the existing trajectory outlier detection models. An open research issue of the trajectory outlier detection is to define standard outlier score metrics for a comparison and evaluation of the existing outlier models in trajectory data [91]. Proposals in the state of the art include trajectory distance metrics: Euclidean distance, Hausdorff distance, longest common sub-sequence (LCCS), dynamic time warping (DTW), and others based on structural distance including comparison of direction, angle, speed, and density.

We provide three different taxonomies to categorize the existing solutions:

- *Algorithm based*: This classifies the existing trajectory outlier detection algorithms according to the techniques employed in finding outliers. Our extensive research study reveals that almost all of the solutions for trajectory outlier detection are distance based, density based, and pattern mining based. In addition, there are few works based on machine learning solutions. Distance-based approaches are based on neighborhood computation. Density-based approaches aim to compute the density of each trajectory and consider trajectories with low density values as outliers, A pattern mining–based approaches explore the different correlations among trajectories to find the outliers. Machine learning–based approaches learn the outlier detection process from the training trajectories to identify anomalies in the newly inserted trajectories.

- *Application based*: This allows different users to understand which algorithm is suitable for their application. This category includes solutions regarding the application resolved, such as(1) maritime transportation, to determine outliers from ships trajectory; (2) urban and flight transportation, to detect outliers from taxi, buses, and flight trajectories; (3) video surveillance and communication, which derive anomalies from video streams and mobile networking, such as movement of people and routing communications; and (4) climate change, which derives outliers from hurricane data.
- *Output based*: This allows the user to understand the different outputs of the trajectory outlier detection algorithms and decide which algorithm could be used for their case. It includes solutions regarding the output of the trajectory outliers (full trajectory or sub-trajectory). As for the trajectory, it aims to detect anomalies after processing the whole trajectory database. These algorithms are fast, but they only detect the whole trajectory and not the specific chunks causing anomalies. As for the sub-trajectory, these algorithms are capable of finding chunks in trajectories causing anomalies. However, they require a huge amount of memory and computing resources to find and analyze the sub-trajectories. Table 2 summarizes the taxonomies proposed in this work.

4 EVALUATION

4.1 Metrics and Datasets

The evaluation is performed using the F-measure and the area under the curve of the receiver operating characteristic (AUC), which are common measures for the evaluation of outlier detection methods [44]. The common problem of outlier detection approaches is the missing of the ground truth in the standard databases. To solve this issue, virtual outlier trajectories are generated from the test database using probability $\mathcal{P}$. A virtual outlier trajectory is obtained by injecting synthetic anomalies such as

$$\forall t \in T, \forall p_i \in t \begin{cases} p_i = p_i + p^j_{noise} & \mathcal{P} \geq \gamma \\ p_i = p_i & Otherwise, \end{cases} \tag{4}$$

where T is the trajectory database and t is a single trajectory. In addition, p_{noise} is a Gaussian noise with 0 as mean and σ^2 as variance: $p_{noise} \sim \mathcal{N}(0, \sigma^2)$. p^j_{noise} is the j^{th} value returned by the Gaussian noise p_{noise}. p_i is the i^{th} point of the trajectory t. $\mathcal{P}$ is the probability that the noise observation is injected. γ is the noise injection threshold.

The process starts by varying the noise injection threshold from 0.1 to 0.9, and then we iteratively generate noise trajectories for each noise injection value. This strategy allows to create noise trajectories in different levels of anomalies. This procedure allows us to divide the data into outlier trajectories and inlier trajectories. The F-measure and AUC are given as follows:

$$F - measure = \frac{2 \times Recall \times Precision}{Recall + Precision}, \tag{5}$$

$$Recall = \frac{|O_A \bigcap O|}{|O|}, \tag{6}$$

Table 2. Taxonomies of Existing Trajectory Outlier Detection Algorithms

Algorithms	Application Based				Output Based		Algorithm Based			
Problems	Maritime Transportation	Urban & Flight Transportation	Video Surveillance & Communication	Climate Change	Trajectory	Sub-Trajectory	Distance	Density	Pattern Mining	Others
TRAOD [71]		√				√	√			
Piciarelli et al. [100]			√		√					√
CTOD [133]		√				√	√			
TOP-EYE [51]			√			√		√		
iBAT [138]		√			√				√	
Fourier [89]		√		√	√		√			
RTOD [80]				√		√	√			
iBOAT [29]		√				√			√	
RPAT [108]			√			√	√			
PN-Outlier [136]		√	√		√			√		
Detect [68]		√			√					√
BT-miner [81]			√		√					√
SHNN-CAD [70]			√		√		√			
TPRO [147]		√			√		√			
ACE [7]			√			√				√
iBDD [76]		√			√				√	
MANTRA [10]		√				√			√	
ROSE [85]			√			√				√
MT-MAD [72]	√					√			√	
DB-TOD [128]		√				√				√
TN-Outlier [134]		√	√		√			√		
LDTRAOD [83]				√		√		√		
CaD [109]			√			√	√			
TF-Outlier [88]		√				√		√		
LoTAD [64]		√				√		√		
TODCSS [135]			√			√			√	
F-DBSCAN [141]	√				√			√		
*k*AA [103]	√					√	√			
STN-Outlier [148]		√				√		√		
Wang et al. [123]		√			√		√			
Ying et al. [132]		√				√				√
DS-Traj [104]		√			√					√
*k*NN-LOF [60]	√				√			√		
ConvNet [20]			√		√					√
SCG-SFL [33]			√		√					√
LoOP-AF [99]		√			√			√		

$$Precision = \frac{|O_A \cap O|}{|O_A|}, \tag{7}$$

$$AUC = mean_{o \in O_A, i \in I_A} \begin{cases} 1 & if\ score(o) > score(i) \\ 0.5 & if\ score(o) = Score(i) \\ 0 & if\ score(o) < score(i) \end{cases}, \tag{8}$$

where O is the set of all outliers in the dataset, O_A is the set of outliers returned by the algorithm, and I_A is the set of inliers returned by the algorithm.

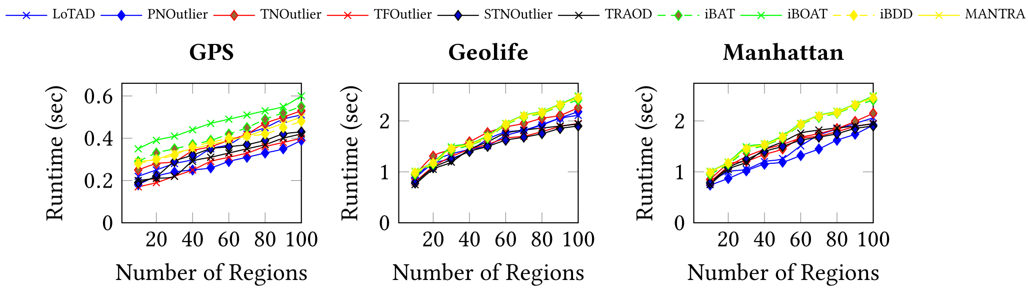


Fig. 2. Runtime performance with different numbers of regions.

The datasets used in this comparative analysis are well-known sparse trajectory databases:

(1) The *GPS* database retrieved from the UCI repository[1] contains GPS points of the trajectories people are taking while driving theirs cars [34]. This database contains 603 trajectories with 5,317 different points. This database is considered to be sparse, and the number of points per trajectory exceeds 2,000.
(2) *Geolife*[2] was collected by Microsoft Research Asia. The Geolife trajectory database contains 17,621 trajectories with 152,241 different points [144]. This database is considered to be sparse, and the number of points per trajectory exceeds 5,000.
(3) *Manhattan*[3] consists of 1,000 taxi trajectories collected over a 1-year period. This database is also considered to be sparse, and the number of points per trajectory exceeds 1,000 points.
(4) In *Taxi Trajectories*,[4] three trajectory databases are evaluated: (1) taxi 13-1 containing 1.89 million trajectories, (2) taxi 13-2 containing 3.69 million trajectories, and (3) taxi 15 containing 57,000 trajectories. These trajectory databases are considered to be sparse, and each trajectory contains more than 1,500 different points.

Several experiments have been conducted on existing outlier trajectory detection algorithms using the preceding databases. The first experiment aims to compare distance-, density-, and pattern mining–based approaches in terms of quality and runtime performance by varying the number of neighbors and the minimum support threshold. The second experiment aims to study the ability of existing outlier trajectory approaches in terms of the runtime by varying both the number of trajectories and the number of points per trajectory. We finish the experiment by studying the statistical analysis and limitations of the existing trajectory outlier detection algorithms. In the following, 10 algorithms are compared, including iBAT, iBOAT, iBDD, MANTRA, LoTAD, PN-Outlier, TN-Outlier, TF-Outlier, STN-Outlier, and TRAOD.

4.2 Runtime

Figures 2 and 3 present the runtime performance of existing trajectory outlier detection by studying data sparsity. Therefore, intensive experiments have been carried out by varying both the number of regions and the number of trajectories as input. By increasing the number of regions from 10 to 100, in the first set of results we can observe that the pattern mining–based solutions are very time consuming compared to the distance- and density-based solutions. In fact, the

[1] https://archive.ics.uci.edu/ml/datasets/.
[2] https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/.
[3] https://www.cs.cornell.edu/~arb/data/Manhattan-taxi-trajectories.
[4] https://www.geomesa.org/.

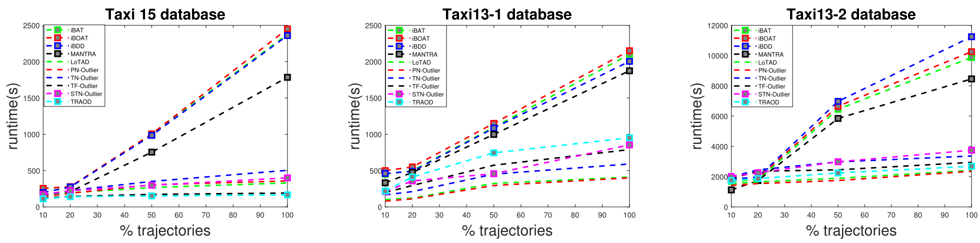


Fig. 3. Runtime performance with different numbers of trajectories.

pattern mining–based solutions are very sensitive to the number of items (regions in this case); a high number of regions results in a huge search space, and consequently complex enumeration trees are explored. In addition, iBOAT is significantly higher compared to the other pattern mining–based approaches, whose aim is not only to identify trajectory outliers but also the sub-trajectory outliers. This process requires high computational and memory resources. The density- and distance-based solutions are fast compared to the pattern mining–based solutions because they only consider computing similarities and density functions to identify outliers. TN-Outlier is slower because it uses the sub-trajectory for density computation. However, PN-Outlier is faster because it is based on point-based density computation. By increasing the percentage of trajectories used in the experiment from 10 to 100, in the first set of results we can observe that the pattern mining–based solutions are faster than the distance- and density-based solutions. In fact, the pattern mining–based solutions are not sensitive to the number of transactions (trajectories in this case). The complexity of the pattern mining–based solutions are highly dependent on the number of regions, and it is polynomial to the number of trajectories. However, density- and distance-based solutions need to compute and determine similarity between all trajectories. As the number of trajectories increase, the computational time of these approaches increases as well. From this study, we can conclude that if we deal with a large number of regions in real-time processing, the distance- and density-based solutions are more suitable than the pattern mining-based solutions. In case of dealing with a large number of trajectories in real-time processing, the pattern mining–based solutions are more suitable than the distance- and density-based solutions.

4.3 Accuracy

Figure 4 presents the quality performance determined by the F-measure and AUC of the existing trajectory outlier detection algorithms using the standard trajectory databases (GPS, Manhattan, and Geolife). We varied different parameters including the number of neighbors for distance, density-based solutions, and the minimum support threshold for the pattern mining–based solutions, where intensive experiments have been conducted to study the effect of such parameters on the existing trajectory outlier detection algorithms. By increasing the number of neighbors from 1 to 15, the quality of the distance- and density-based approaches increased in every database used in the experiment. On 15 neighbors, the quality converges for almost all solutions, except TN-Outlier and LoTAD, which converge on 30 neighbors. Moreover, TN-Outlier and LoTAD give good results in terms of quality of outliers (F-measure and AUC). This is due to the outlier detection algorithm used that is based on LOF, where the density computation is done for all points of each trajectory to determine its score. In addition, TN-Outlier uses the sub-trajectory for computing density, which is very accurate, compared to PN-Outlier, which is based on point-based density computation. By decreasing the minimum support value from 90% to 20%, the quality of the pattern mining–based approaches increased. This is explained by the fact that by considering low values of mining support, more relevant patterns are discovered and therefore enhance the detection of

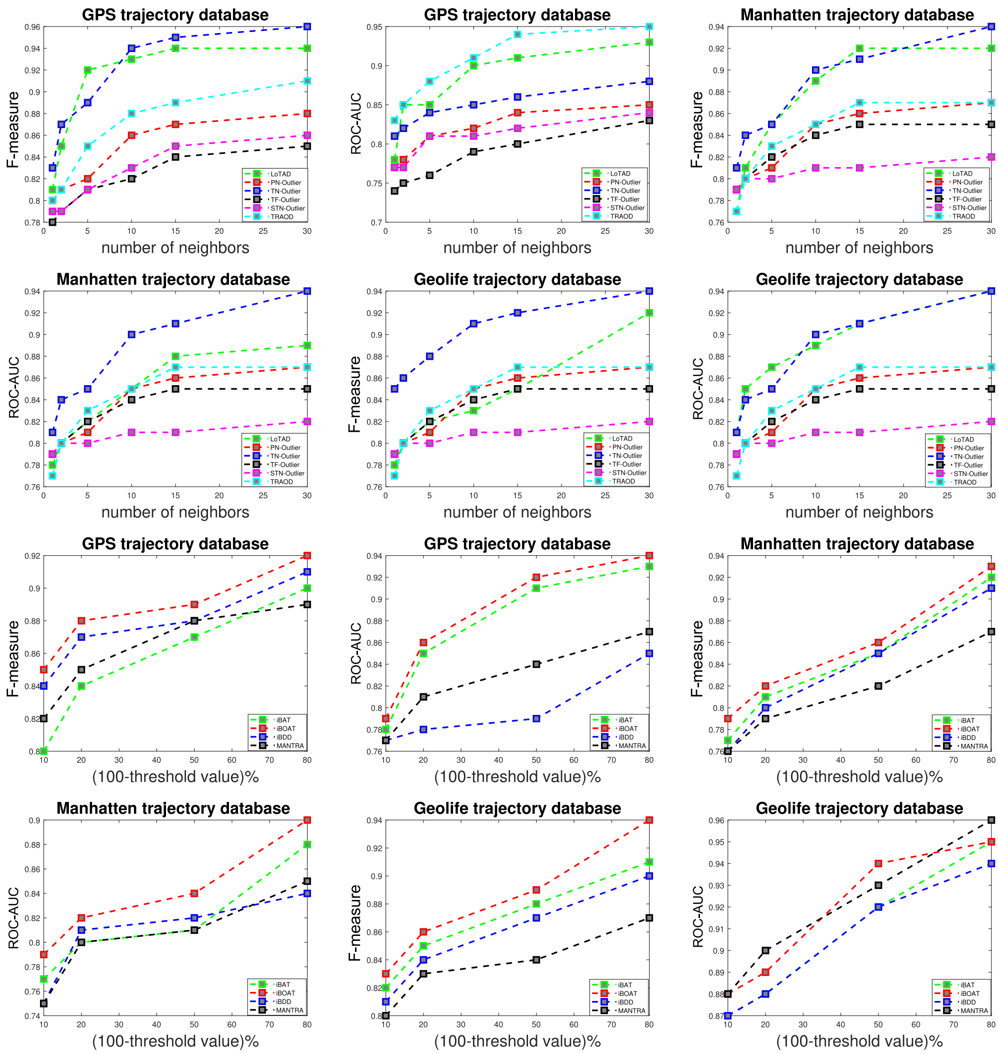


Fig. 4. Quality performance on standard trajectory databases (GPS, Manhattan, and Geolife).

anomalous patterns in the whole trajectory database. Moreover, these solutions provide relevant functions and measures to determine the outlier score of each trajectory. In addition, iBOAT gives the best results for almost all cases, which is explained by the fact that iBOAT aims to identify sub-trajectory outliers, the exact sub-sequences in the trajectories, which cause anomalies. However, the other pattern mining–based solutions only identify whether the given trajectory is an outlier or not. The next experiment aims to compare the outlier trajectory detection in both categories for dealing with taxi trajectory databases in terms of quality. The results reported in Table 3 show that by increasing the database size, the quality of all approaches lessens. Moreover, the pattern mining–based approaches outperform the distance- and density-based approaches in terms of both F-measure and AUC. This is because pattern mining–based solutions study the different correlations between the trajectories to find the outliers, whereas the distance- and density-based solutions only compute the similarity of the single trajectory with all remaining trajectories.

Table 3. Quality Performance on Taxi Trajectory Databases

Dataset	Taxi15		Taxi13-1		Taxi13-2	
Algorithm	F-measure	AUC	F-measure	AUC	F-measure	AUC
iBAT	0.82	0.80	0.78	0.78	0.71	0.68
iBOAT	0.83	0.79	0.80	0.78	0.75	0.72
iBDD	0.81	0.80	0.78	0.77	0.72	0.71
MANTRA	0.80	0.80	0.76	0.75	0.68	0.66
LotAD	0.79	0.78	0.77	0.75	0.69	0.66
PNOutlier	0.81	0.79	0.77	0.76	0.67	0.66
TNOutlier	0.81	0.79	0.79	0.77	0.71	0.70
TFOutlier	0.80	0.78	0.78	0.76	0.70	0.69
STNOutlier	0.80	0.78	0.77	0.75	0.69	0.68
TRAOD	0.80	0.78	0.79	0.77	0.68	0.67

4.4 Statistical Analysis and Discussion

This section aims to analyze and discuss the results obtained in the previous section by performing the Z-test for the 10 trajectory outlier detection algorithms used in this study, including iBAT, iBOAT, iBDD, MANTRA, LoTAD, PN-Outlier, TN-Outlier, TF-Outlier, STN-Outlier, and TRAOD, using the six trajectory databases used in the experiment. This can be modeled as follows:

(1) Each algorithm is viewed as a normal variable.
(2) Each trajectory database is divided into 10 partitions, and each partition contains 10% of the entire trajectories. Each partition represents an observation, and as a result, 60 different observations are generated.
(3) The result of each partition is considered as a sample.

Twenty-seven estimators (from E_1 to E_{27}) are used in the analysis. The first nine estimators are designated for the runtime performance, the second nine estimators are designated for F-measure performance, and the last nine estimators are designated for AUC performance. A detailed description of these estimators is given as follows:

Performance Estimators
E_1 = CPU(iBAT)-CPU(iBOAT)
E_2 = E_1-CPU(iBDD)
E_3 = E_2-CPU(MANTRA)
E_4 = E_3-CPU(LoTAD)
E_5 = E_4-CPU(PN-Outlier)
E_6 = E_5-CPU(TN-Outlier)
E_7 = E_6-CPU(TF-Outlier)
E_8 = E_7-CPU(STN-Outlier)
E_9 = E_8-CPU(TRAOD)

F-measure Estimators
E_{10} = F(iBAT)-F(iBOAT)
E_{11} = E_{10}-F(iBDD)
E_{12} = E_{11}-F(MANTRA)
E_{13} = E_{12}-F(LoTAD)
E_{14} = E_{13}-F(PN-Outlier)
E_{15} = E_{14}-F(TN-Outlier)
E_{16} = E_{15}-F(TF-Outlier)
E_{17} = E_{16}-F(STN-Outlier)
E_{18} = E_{17}-F(TRAOD)

AUC Estimators
E_{19} = AUC(iBAT)-AUC(iBOAT)
E_{20} = E_{19}-AUC(iBDD)
E_{21} = E_{20}-AUC(MANTRA)
E_{22} = E_{21}-AUC(LoTAD)
E_{23} = E_{22}-AUC(PN-Outlier)
E_{24} = E_{23}-AUC(TN-Outlier)
E_{25} = E_{24}-AUC(TF-Outlier)
E_{26} = E_{25}-AUC(STN-Outlier)
E_{27} = E_{26}-AUC(TRAOD)

where CPU(A) is the average of runtime values of the given algorithm A in the 60 observations. F(A) is the average of F-measure values of the given algorithm A in the 60 observations. AUC(A) is the average of AUC values of the given algorithm A in the 60 observations. A is the algorithm that belongs to the set {iBAT, iBOAT, iBDD, MANTRA, LoTAD, PN-Outlier, TN-Outlier, TF-Outlier, STN-Outlier, TRAOD}.

First, the normality of the 10 algorithms is checked using the Shapiro-Wilk test, which is available on the XLSTAT tool. Therefore, the first hypothesis, H_0, and the alternative hypothesis, H_α, are defined as follows:

- H_0: The algorithms follow a normal distribution.
- H_α: The algorithms do not follow a normal distribution.

The used significance level (α) was set to 1%. The results of the Shapiro-Wilk test indicate that H_0 cannot be rejected. Hence, the algorithms follow the normal distribution. In other words, the non-normality is not significant. Afterward, the Z-test is used with $\alpha = 5\%$ to compare the algorithms. XLSTAT shows that E_6, E_{11}, and E_{20} give higher values than the other estimators, which means that PN-Outlier is statistically better than the other algorithms in terms of runtime, and iBOAT is statistically better than the other algorithms in terms of F-measure and AUC.

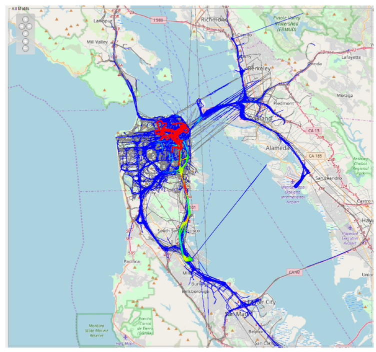
From our analysis, we can conclude that the pattern mining–based solutions are better in terms of accuracy than the distance- and density-based solutions. Thus, the pattern mining–based solutions study the different correlations among trajectories and find the relevant patterns useful to identify anomalies from trajectories. iBOAT is the best algorithm in this category in terms of accuracy, of which the aim is to identify not only the trajectories outliers but the sub-sequences that cause anomalies. However, these algorithms are highly time and memory consuming, and they are very sensitive to the number of items (regions) and the minimum support threshold. As the number of items become large and the minimum support threshold is small, these algorithms generate a huge enumeration tree. Big trees are hard to store and explore, making it difficult to identify the relevant patterns. In contrast, the distance- and density-based solutions are fast compared to the pattern mining–based solutions and do not need a lot of memory capacity to find the trajectory outliers. PN-Outlier is the best algorithm in terms of accuracy, as it is a point-based density computation algorithm. The complexity of these algorithms is polynomial, depending on the number of trajectories, unlike the pattern mining–based solutions, which are exponential, depending on the number of regions. However, these solutions do not consider the different dependencies among the trajectories, as they compute the similarity and the density of a single trajectory separately. To summarize, if the application needs real-time processing, the similarity- and density-based solutions are more suitable. The researchers and users would need to optimize and work on the PN-Outlier variants. However, if the given application needs high quality and precision of the returned trajectory outliers, the pattern mining–based solutions are preferred. The researchers and users would need to optimize and work on the different variants and adaptation of the iBOAT algorithm.

5 OPEN RESEARCH ISSUES

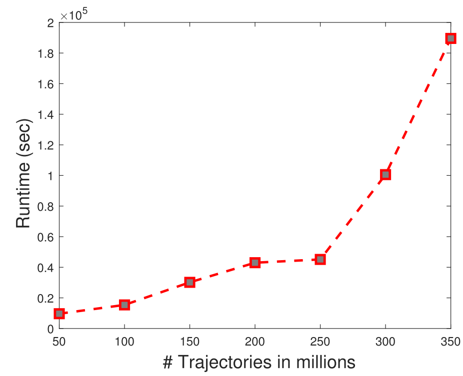
In this section, we provide insight on several open research issues regarding trajectory outlier detection. Some research issues can be considered as general topics that can be applied in any problem with intensive computing, and other issues are problem specific and only relevant to trajectory outlier detection. The open research issues around algorithms, taxonomy, and evaluation are described next.

5.1 Algorithms

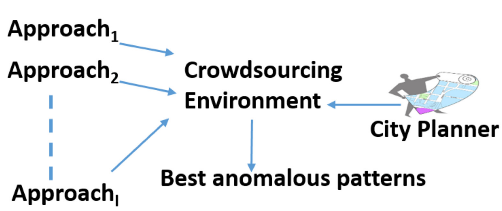
5.1.1 Quality Improvements. Several algorithms could be investigated to improve the quality of the detected trajectory outliers.



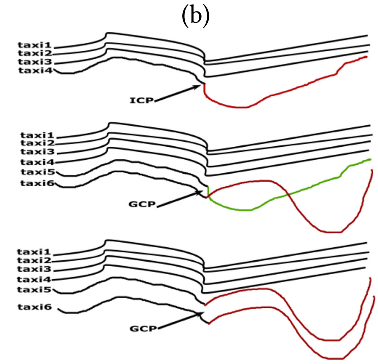
(a)



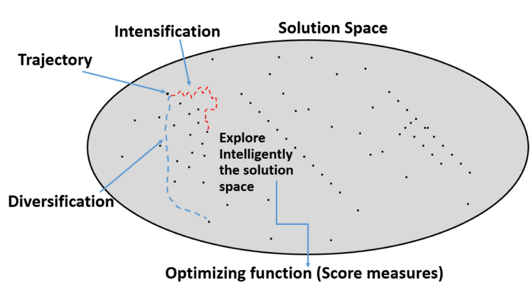
(b)



(c)



(d)



(e)

Fig. 5. Open research issues: quality improvements (a), runtime improvements (b), crowdsourcing (c), multi-view outliers (d), and bio-inspired computing (e).

Studying the different dependencies between training trajectories. Figure 5(a) presents a sketch of the TrajViz tool [50] for exploring pattern mining algorithms on a San Francisco taxi dataset.[5] This figure reveals different levels of dependencies between trajectories. Highly correlated trajectories are shown in red (i.e., these trajectories share a large number of sub-trajectories). However, a small number of correlated trajectories are shown in blue (i.e., these trajectories share few sub-trajectories). Incorporating trajectory pattern mining approaches [15, 45, 52, 74] and exploring the discovered patterns with the existing trajectory outlier detection is a challenging problem and

[5]https://github.com/flash121123/TrajViz.

may improve the quality of the returned outliers. In particular, datasets with a high degree of correlations, such as the San Francisco Taxi dataset, are considered as one of the big challenges in trajectory outlier detection.

Adapting more specific advanced methods. In this context, many adaptations have been investigated regarding specific scenarios, such as spatial data [112], graph data [73], or time series and sequence data [17]. All of these special scenarios are somehow related to possible scenarios in tackling trajectory databases, and thus methods proposed in the literature for these scenarios could also be relevant for adaptations to trajectories and for tackling these different aspects simultaneously.

5.1.2 Runtime Improvements. Existing trajectory outlier detection techniques are very expensive with regard to computing time, particularly when augmenting the number of points. Figure 5(b) presents the runtime of the TRAOD algorithm [71] on 10 consecutive days of the Beijing taxi dataset, containing 350 million trajectories. The figure reveals that by increasing the number of trajectories, TRAOD has become highly time consuming. For instance, with 50 million trajectories, TROAD needs around 2.5 hours; however, with 350 million trajectories, TRAOD needs more than 52 hours (approximately 3 consecutive days) to identify outliers. Fortunately, TRAOD is improved by proposing Spark-based implementations [31, 87], but many other trajectory outlier detection algorithms are not yet improved. Proposing high-performance computing tools may be helpful when dealing with trajectory outliers in real-time environments [24, 42]. However, several questions should be addressed. For example, which architectures should be used? How do we efficiently partition the data among the different jobs? How can we design a parallel approach respecting high-performance computing challenges such as reducing communication and synchronization cost or increasing load balancing and optimizing memory management?

5.1.3 Multi-View Outliers. Consider the three sketched examples of taxi trajectories illustrated in Figure 5(d). Each taxi trajectory starts from the source point and ends at the destination point. Traditional trajectory outlier detection algorithms may detect outliers in the top portion, illustrated in red, where *taxi4* follows a normal trip from source to destination until a given point, which it highly deviates from the *taxi1*, *taxi2*, and *taxi3*. However, these algorithms [23] cannot determine the individual change point (ICP) for this case. Moreover, the whole process is not able to identify outliers presented in red and green colors in middle and bottom portions. In the middle portion, different taxis (*taxi5* and *taxi6*) deviated from the normal trip at the same group change point (GCP) but followed different trajectories (green trajectory for *taxi5* and red trajectory for *taxi6*). However, in the case illustrated in the bottom portion, two taxis deviated from the normal trip at the same GCP and followed the same trajectory shown in red. Detecting these different kinds of outliers could help city planners extract and discover relevant knowledge. For instance, the case in the top portion allows determination of individual taxi fraud, and detecting frequent individual taxi fraud at the same ICP allows city planners to make good decisions, such as putting up a surveillance camera at this point. In the cases in the middle and bottom portions, detecting group trajectory outliers allows city planners to make fair decisions regarding the taxis outliers. When taxis deviate from the same GCP but follow different trajectories, there is a strong probability that their aim is to avoid circumstances such as traffic jams rather than it being a situation of taxi fraud. However, when there are group taxis outliers at the same or different GCPs with the same trajectory, there is a strong probability that they are partners in taxi fraud. The problem relates to existing taxi fraud detection algorithms, in which case we can ask these questions: how do we derive individual trajectory outliers (ITOs) and group trajectory outliers (GTOs), and how do we determine ICP and GCP?

5.1.4 Bio-Inspired Computing. Several works have been developed using bio-inspired computing for solving outlier detection problems. Some works are based on evolutionary algorithms [35, 75, 119, 125], whereas others are based on swarm intelligence algorithms [62, 94, 114, 115]. Adopting these techniques for trajectory data is an open research issue. With regard to Figure 5(e), several questions should be addressed. First, how do we define the solution space? Each trajectory is considered as a solution, and the challenge here is to define a good representation of trajectories to efficiently perform the different bio-inspired operators. Second, how do we explore the trajectory space? It is important to explore the trajectory space in an efficient way to find the anomalous trajectories. Bio-inspired approaches mainly provide two components: the exploitation search, which allows one to focus on exploring a local region for offering good solutions, and the exploration search, which aims to generate diverse solutions to explore the whole space on a global scale [19]. In this context, the challenge is to provide intelligent specified operators for trajectory data and to explore the trajectory data with respect to both exploitation and exploration criteria.

5.2 Taxonomies

As shown in this work, several solutions with regard to trajectory outlier detection have been applied in different industrial applications. Other industrial applications that could be relevant to trajectory outlier detection are discussed next.

5.2.1 Health-Care Data. Trajectory outlier detection could be applied in health-care data [131] for monitoring the movement of patientssuch as to discover events like a bad fall, encountering a sudden slowdown, and becoming lost.

5.2.2 Smart Building. Detecting trajectory outliers can be useful in the smart building domain [37] to discover anomalous situations. An example would be a resident starting water for a bath but does not turn it off before going to bed.

5.2.3 Environmental Sensor Data. Trajectory outlier detection could be helpful in environmental sensor data [8] by identifying measurement errors in a wind speed data stream, or by discovering outliers from the rain, sea surface temperature, relative humidity, precipitation time series, and so on.

5.2.4 Biological Data. Trajectory outlier detection could also be relevant to biological data [124] by discovering abnormalities from electrocardiograms in zoology and anthropology data or deriving abnormal migration behaviors of different species.

An open research issue for applying trajectory outlier detection in industrial applications is to understand the categories of algorithms (distance based, density based, pattern mining based, etc.), data representation (sequence data, time series, images, and videos), and data output (trajectory and sub-trajectory outliers). In addition to the taxonomies proposed in this research work, it will be beneficial if the research community in trajectory outlier detection proposes other taxonomies deeply focused on each industrial application.

5.3 Evaluation

5.3.1 Crowdsourcing. Solutions regarding trajectory outlier detection could identify different anomalous patterns from the same trajectory data. The problem is how to decide which patterns are useful to city planners. To improve the usefulness of the detected patterns, a crowdsourcing approach may be applied [59], where different trajectory outlier detection approaches should work together to identify the best anomalous patterns delivered to city planners. New definitions and corrections related to the term *crowdsourcing* were first introduced by Howe [57]. In this context, crowdsourcing is not only used by a crowd of people but is used by crowds of agents such as approaches and machines. Figure 5(c) explains how to introduce crowdsourcing in trajectory outlier detection approaches. Agents represented by approaches and programs find the trajectory outliers locally and report them back to the city planners. The city planners use crowdsourcing environments to find the best anomalous patterns for their applications. Adapting the existing crowdsourcing-based evaluation strategies [1, 120, 122] may give the city planner a fair interpretation of the trajectory outliers.

5.3.2 Missing of Ground Truth. Missing of the ground truth is a common problem in evaluating trajectory outlier detection algorithms [79, 82, 91, 145, 150]. As challenges for future research on the aspect of quality assessment of trajectory outlier detection results, the following issues and research questions can be observed:

(1) Defining useful, publicly available benchmark data for trajectory outlier detection problems is beneficial for analyzing trajectory outlier detection algorithms.
(2) It would be very useful to identify meaningful criteria for an internal evaluation of trajectory outlier detection. One way to address this challenging issue is to provide unified ranking-function scores to rank trajectory outliers. These functions should be independent from the entire process for retrieving trajectory outliers.

6 CONCLUSION

Trajectory outlier detection has been largely used in management information systems for a long time. It includes solutions regarding the algorithm used. First, distance is based on similarity and neighborhood computation, and second, density is based on density estimation to determine outliers. Both of these two categories have high accuracy but ignore the different dependencies between trajectory. Third, with regard to pattern mining, these algorithms are not only capable of deriving trajectory outliers but also study the correlation between the chunks of trajectories causing outliers. This helps to extract useful patterns and deals with exciting applications: causal interaction and congested patterns. Nevertheless, these approaches are highly time consuming because they require multiple scans of the trajectory database. Fourth, other approaches explore different machine learning techniques to determine anomalies such as SVMs, ensemble learning, feature selection, and granular computing. Algorithms have become sophisticated in their ability to deal with real-time situations. However, such solutions have reached their limits due to the increased complexity of real-life industrial data captured in different domains, such as transportation, manufacturing, video surveillance, and climate change. We have provided a comprehensive survey with intensive evaluations of the existing solutions in trajectory outlier detection. We conclude by showing the most challenging issues that should be addressed in the future for boosting overall performance of the anomaly detection process. From our intensive research studies on existing trajectory outlier detection algorithms, we recommend the following algorithms under different conditions. First, if the problem at hand needs to study different correlations among the trajectory outliers and the results require high precision, we recommend iBOAT [29]. Second, if it is straightforward to determine the dissimilarity between the different trajectories, we recommend TN-Outlier [134]. Third, if the application to be designed should be performed using real-time processing under large and big trajectory databases, we recommend iBDD [76].Fourth, if the database to be processed is incomplete, we recommend BT-miner [81]. Fifth, if the database to be processed disposes training historical trajectories with undefined classes (normal and abnormal), we recommend ROSE [85]. We have seen only the tip of the iceberg, and much exploration is needed and further progress is required in all directions to reach mature solutions for dealing with large-scale trajectory data.

REFERENCES

[1] Simon à Campo, Vasssilis-Javed Khan, Konstantinos Papangelis, and Panos Markopoulos. 2019. Community heuristics for user interface evaluation of crowdsourcing platforms. *Future Generation Computer Systems* 95 (2019), 775–789.

[2] Palwasha Afsar, Paulo Cortez, and Henrique Santos. 2015. Automatic visual detection of human behavior: A review from 2000 to 2014. *Expert Systems with Applications* 42 (2015), 6935–6956.

[3] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. *ACM SIGMOD Record* 22, 2 (1993), 207–216.

[4] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. 2016. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems* 55 (2016), 278–288.

[5] Alessia Albanese, Sankar K. Pal, and Alfredo Petrosino. 2014. Rough sets, kernel set, and spatiotemporal outlier detection. *IEEE Transactions on Knowledge and Data Engineering* 26, 1 (2014), 194–207.

[6] Nawaf O. Alsrehin, Ahmad F. Klaib, and Aws Magableh. 2019. Intelligent transportation and control systems using data mining and machine learning techniques: A comprehensive study. *IEEE Access* 7 (2019), 49830–49857.

[7] Shin Ando, Theerasak Thanomphongphan, Yoichi Seki, and Einoshin Suzuki. 2015. Ensemble anomaly detection from multi-resolution trajectory features. *Data Mining and Knowledge Discovery* 29, 1 (2015), 39–83.

[8] Fabrizio Angiulli and Fabio Fassetti. 2007. Detecting distance-based outliers in streams of data. In *Proceedings of the ACM Conference on Information and Knowledge Management*. 811–820.

[9] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2018. Spatio-temporal data mining: A survey of problems and methods. *ACM Computing Surveys* 51, 4 (2018), 1–41.

[10] Prithu Banerjee, Pranali Yawalkar, and Sayan Ranu. 2016. Mantra: A scalable approach to mining temporally anomalous sub-trajectories. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1415–1424.

[11] Vic Barnett and T. Lewis. 1995. Outliers in statistical. *Biometrical Journal* 37, 2 (1995), 256–256.

[12] Sabyasachi Basu and Martin Meckesheimer. 2007. Automatic outlier detection for time series: An application to sensor data. *Knowledge and Information Systems* 11, 2 (2007), 137–154.

[13] Asma Belhadi, Youcef Djenouri, Jerry Chun-Wei Lin, , and Alberto Cano. 2020. A data-driven approach for Twitter hashtag recommendation. *IEEE Access* 8 (2020), 79182–79191.

[14] Asma Belhadi, Youcef Djenouri, and Jerry Chun-Wei Lin. 2019. Comparative study on trajectory outlier detection algorithms. In *Proceedings of the International Conference on Data Mining Workshops*. 415–423.

[15] Asma Belhadi, Youcef Djenouri, Jerry Chun-Wei Lin, and Alberto Cano. 2020. A general-purpose distributed pattern mining system. *Applied Intelligence*. Open Access. March 18, 2020. DOI: https://doi.org/10.1007/s10489-020-01664-w

[16] Asma Belhadi, Youcef Djenouri, Jerry Chun-Wei Lin, Chongsheng Zhang, and Alberto Cano. 2020. Exploring pattern mining algorithms for hashtag retrieval problem. *IEEE Access* 8 (2020), 10569–10583.

[17] Seif-Eddine Benkabou, Khalid Benabdeslem, and Bruno Canitia. 2018. Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowledge and Information Systems* 54, 2 (2018), 463–486.

[18] Kiran Bhowmick and Meera Narvekar. 2018. Trajectory outlier detection for traffic events: A survey. In *Intelligent Computing and Information and Communication*, S. Bhalla, V. Bhateja, A. Chandavale, A. S. Hiwale, and S. C. Satapathy (Eds.). Springer, 37–46.

[19] Christian Blum and Andrea Roli. 2003. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys* 35, 3 (2003), 268–308.

[20] Samir Bouindour, Hichem Snoussi, Mohamad Mazen Hittawe, Nacef Tazi, and Tian Wang. 2019. An on-line and adaptive method for detecting abnormal events in videos using spatio-temporal ConvNet. *Applied Sciences* 9, 4 (2019), 757.

[21] Luc Brun, Alessia Saggese, and Mario Vento. 2014. Dynamic scene understanding for behavior analysis based on string kernels. *IEEE Transactions on Circuits and Systems for Video Technology* 24 (2014), 1669–1681.

[22] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. 2011. Robust principal component analysis? *Journal of the ACM* 58, 3 (2011), 11.

[23] Alberto Cano. 2017. An ensemble approach to multi-view multi-instance learning. *Knowledge-Based Systems* 136 (2017), 46–57.

[24] Alberto Cano. 2018. A survey on graphic processing unit computing for large-scale data mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 1 (2018), e1232.

[25] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. arXiv:1901.03407.

[26] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys* 41 (2009), 1–15.

[27] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2010. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering* 24, 5 (2010), 823–839.

[28] Moon-Hwan Chang, Chaochao Chen, Diganta Das, and Michael Pecht. 2014. Anomaly detection of light-emitting diodes using the similarity-based metric test. *IEEE Transactions on Industrial Informatics* 10, 3 (2014), 1852–1863.

[29] Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, Shijian Li, and Zonghui Wang. 2013. iBOAT: Isolation-based online anomalous trajectory detection. *IEEE Transactions on Intelligent Transportation Systems* 14, 2 (2013), 806–818.

[30] Jingying Chen, Maylor K. Leung, and Yongsheng Gao. 2003. Noisy logo recognition using line segment Hausdorff distance. *Pattern Recognition* 36, 4 (2003), 943–955.

[31] Yuanyuan Chen, Jingyi Yu, and Yong Gao. 2017. Detecting trajectory outliers based on Spark. In *Proceedings of the International Conference on Geoinformatics*. 1–5.

[32] Yunqiang Chen, Xiang S. Zhou, and Thomas S. Huang. 2001. One-class SVM for learning in image retrieval. In *Proceedings of the International Conference on Image Processing*, Vol. 1. 34–37.

[33] Wenqing Chu, Hongyang Xue, Chengwei Yao, and Deng Cai. 2019. Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos. *IEEE Transactions on Multimedia* 21, 1 (2019), 246–255.

[34] Michael O. Cruz, Hendrik Macedo, and Adolfo Guimaraes. 2015. Grouping similar trajectories for carpooling purposes. In *Proceedings of the Brazilian Conference on Intelligent Systems*. 234–239.

[35] Xiaowu Deng, Peng Jiang, Xiaoning Peng, and Chunqiao Mi. 2019. An intelligent outlier detection method with one class support Tucker machine and genetic algorithm toward big sensor data in Internet of Things. *IEEE Transactions on Industrial Electronics* 66, 6 (2019), 4672–4683.

[36] Thierry Denoeux. 1995. A *k*-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics* 25, 5 (1995), 804–813.

[37] Djamel Djenouri, Roufaida Laidi, Youcef Djenouri, and Ilangko Balasingham. 2019. Machine learning for smart building applications: Review and taxonomy. *ACM Computing Surveys* 52, 2 (2019), 24.

[38] Youcef Djenouri, Asma Belhadi, Jerry Chun-Wei Lin, and Alberto Cano. 2019. Adapted *k* nearest neighbors for detecting anomalies on spatio-temporal traffic flow. *IEEE Access* 7 (2019), 10015–10027.

[39] Youcef Djenouri, Asma Belhadi, Jerry Chun-Wei Lin, Djamel Djenouri, and Alberto Cano. 2019. A survey on urban traffic anomalies detection algorithms. *IEEE Access* 7 (2019), 12192–12205.

[40] Youcef Djenouri and Marco Comuzzi. 2017. Combining Apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem. *Information Sciences* 420 (2017), 1–15.

[41] Youcef Djenouri, Marco Comuzzi, and Djamel Djenouri. 2017. SS-FIM: Single scan for frequent itemsets mining in transactional databases. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 644–654.

[42] Youcef Djenouri, Djamel Djenouri, Asma Belhadi, and Alberto Cano. 2019. Exploiting GPU and cluster parallelism in single scan frequent itemset mining. *Information Sciences* 496 (2019), 363–377.

[43] Youcef Djenouri and Arthur Zimek. 2018. Outlier detection in urban traffic data. In *Proceedings of the International Conference on Web Intelligence, Mining, and Semantics*. 3.

[44] Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. 2018. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition* 74 (2018), 406–421.

[45] Emre Eftelioglu, Shashi Shekhar, Dev Oliver, Xun Zhou, Michael R. Evans, Yiqun Xie, James M. Kang, Renee Laubscher, and Christopher Farah. 2014. Ring-shaped hotspot detection: A summary of results. In *Proceedings of the IEEE International Conference on Data Mining*. 815–820.

[46] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, Vol. 96. 226–231.

[47] Zhenni Feng and Yanmin Zhu. 2016. A survey on trajectory data mining: Techniques and applications. *IEEE Access* 4 (2016), 2056–2067.

[48] Gilberto Fernandes, Joel J. P. C. Rodrigues, Luiz Fernando Carvalho, Jalal F. Al-Muhtadi, and Mario Lemes Proença. 2019. A comprehensive survey on network anomaly detection. *Telecommunication Systems* 70, 3 (2019), 447–489.

[49] Auroop R. Ganguly and Karsten Steinhaeuser. 2008. Data mining for climate change and impacts. In *Proceedings of the IEEE International Conference on Data Mining Workshops*. 385–394.

[50] Yifeng Gao, Qingzhe Li, Xiaosheng Li, Jessica Lin, and Huzefa Rangwala. 2017. Trajviz: A tool for visualizing patterns and anomalies in trajectory. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. 428–431.

[51] Yong Ge, Hui Xiong, Zhi-Hua Zhou, Hasan Ozdemir, Jannite Yu, and Kuo Chu Lee. 2010. TOP-EYE: Top-*k* evolving trajectory outlier detection. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. 1733–1736.

[52] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory pattern mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 330–339.
[53] Prasanta Gogoi, D. K. Bhattacharyya, Bhogeswar Borah, and Jugal K. Kalita. 2011. A survey of outlier detection methods in network anomaly identification. *Computer Journal* 54, 4 (2011), 570–588.
[54] Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. 2014. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering* 26, 9 (2014), 2250–2267.
[55] Riyaz Ahamed Ariyaluran Habeeb, Fariza Nasaruddin, Abdullah Gani, Ibrahim Abaker Targio Hashem, Ejaz Ahmed, and Muhammad Imran. 2019. Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management* 45 (2019), 289–307.
[56] Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial Intelligence Review* 22, 2 (2004), 85–126.
[57] Jeff Howe. 2006. The rise of crowdsourcing. *Wired Magazine* 14, 6 (2006), 1–4.
[58] Weiming Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. 2006. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), 1450–1464.
[59] Chao Huang, Xian Wu, and Dong Wang. 2016. Crowdsourcing-based urban anomaly prediction system for smart cities. In *Proceedings of the ACM International on Conference on Information and Knowledge Management*. 1969–1972.
[60] Yingfu Huang and Qiurong Zhang. 2019. Identification of anomaly behavior of ships based on KNN and LOF combination algorithm. In *AIP Conference Proceedings*, Vol. 2073. AIP Publishing LLC, 020090.
[61] Fan Jiang, Ying Wu, and Aggelos K. Katsaggelos. 2009. A dynamic hierarchical clustering method for trajectory-based unusual video event detection. *IEEE Transactions on Image Processing* 18 (2009), 907–913.
[62] Vagia Kaltsa, Alexia Briassouli, Ioannis Kompatsiaris, Leontios J. Hadjileontiadis, and Michael Gerasimos Strintzis. 2015. Swarm intelligence for detecting interesting events in crowded environments. *IEEE Transactions on Image Processing* 24, 7 (2015), 2153–2166.
[63] JooSeuk Kim and Clayton D. Scott. 2012. Robust kernel density estimation. *Journal of Machine Learning Research* 13 (2012), 2529–2565.
[64] Xiangjie Kong, Ximeng Song, Feng Xia, Haochen Guo, Jinzhong Wang, and Amr Tolba. 2018. LoTAD: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web* 21, 3 (2018), 825–847.
[65] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009. LoOP: Local outlier probabilities. In *Proceedings of the ACM Conference on Information and Knowledge Management*. 1649–1652.
[66] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*. 1097–1105.
[67] Alp Kut and Derya Birant. 2006. Spatio-temporal outlier detection in large databases. *Journal of Computing and Information Technology* 14, 4 (2006), 291–297.
[68] Jinsong Lan, Cheng Long, Raymond Chi-Wing Wong, Youyang Chen, Yanjie Fu, Danhuai Guo, Shuguang Liu, Yong Ge, Yuanchun Zhou, and Jianhui Li. 2014. A new framework for traffic anomaly detection. In *Proceedings of the SIAM International Conference on Data Mining*. 875–883.
[69] Rikard Laxhammar and Göran Falkman. 2011. Sequential conformal anomaly detection in trajectories based on Hausdorff distance. In *Proceedings of the International Conference on Information Fusion*. 1–8.
[70] Rikard Laxhammar and Göran Falkman. 2014. Online learning and sequential anomaly detection in trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 6 (2014), 1158–1173.
[71] Jae-Gil Lee, Jiawei Han, and Xiaolei Li. 2008. Trajectory outlier detection: A partition-and-detect framework. In *Proceedings of the International Conference on Data Engineering*. 140–149.
[72] Po-Ruey Lei. 2016. A framework for anomaly detection in maritime trajectory behavior. *Knowledge and Information Systems* 47, 1 (2016), 189–214.
[73] Sheng Li, Ming Shao, and Yun Fu. 2018. Multi-view low-rank analysis with applications to outlier detection. *ACM Transactions on Knowledge Discovery from Data* 12, 3 (2018), 32.
[74] Zhenhui Li, Bolin Ding, Jiawei Han, and Roland Kays. 2010. Swarm: Mining relaxed temporal moving object clusters. *Proceedings of the VLDB Endowment* 3, 1–2 (2010), 723–734.
[75] Chang-Chun Lin and An-Pin Chen. 2004. Fuzzy discriminant analysis with outlier detection by genetic algorithm. *Computers & Operations Research* 31, 6 (2004), 877–888.
[76] Qiang Lin, Daqing Zhang, Kay Connelly, Hongbo Ni, Zhiwen Yu, and Xingshe Zhou. 2015. Disorientation detection by mining GPS trajectories for cognitively-impaired elders. *Pervasive and Mobile Computing* 19 (2015), 71–85.
[77] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *Proceedings of the IEEE International Conference on Data Mining*. 413–422.
[78] Honghai Liu, Shengyong Chen, and Naoyuki Kubota. 2013. Intelligent video systems and analytics: A survey. *IEEE Transactions on Industrial Informatics* 9, 3 (2013), 1222–1233.
[79] Hongfu Liu, Jun Li, Yue Wu, and Yun Fu. 2019. Clustering with outlier removal. arXiv:1801.01899.

[80] Liangxu Liu, Shaojie Qiao, Yongping Zhang, and JinSong Hu. 2012. An efficient outlying trajectories mining approach based on relative distance. *International Journal of Geographical Information Science* 26, 10 (2012), 1789–1810.
[81] Siyuan Liu, Lei Chen, and Lionel M. Ni. 2014. Anomaly detection from incomplete data. *ACM Transactions on Knowledge Discovery from Data* 9, 2 (2014), 11.
[82] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. 2019. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering* (2019).
[83] Fangjun Luan, Yunting Zhang, Keyan Cao, and Qi Li. 2017. Based local density trajectory outlier detection with partition-and-detect framework. In *Proceedings of the 2017 13th International Conference on Natural Computation, Fuzzy Systems, and Knowledge Discovery.* 1708–1714.
[84] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. *Berkeley Symposium on Mathematical Statistics and Probability* 1, 14 (1967), 281–297.
[85] Francesco Maiorano and Alfredo Petrosino. 2016. Granular trajectory based anomaly detection for surveillance. In *Proceedings of the International Conference on Pattern Recognition.* 2066–2072.
[86] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.* 55–60.
[87] Jiali Mao, Pengda Sun, Cheqing Jin, and Aoying Zhou. 2018. Outlier detection over distributed trajectory streams. In *Proceedings of the SIAM International Conference on Data Mining.* 64–72.
[88] Jiali Mao, Tao Wang, Cheqing Jin, and Aoying Zhou. 2017. Feature grouping-based outlier detection upon streaming trajectories. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2696–2709.
[89] Elio Masciari. 2011. Trajectory outlier detection using an analytical approach. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'11).* 377–384.
[90] Shaohui Mei, Genliang Guan, Zhiyong Wang, Shuai Wan, Mingyi He, and David Dagan Feng. 2015. Video summarization via minimum sparse reconstruction. *Pattern Recognition* 48 (2015), 522–533.
[91] Fanrong Meng, Guan Yuan, Shaoqian Lv, Zhixiao Wang, and Shixiong Xia. 2019. An overview on trajectory outlier detection. *Artificial Intelligence Review* 52, 4 (2019), 2437–2456.
[92] Thomas J. Misa and Philip L. Frana. 2010. An interview with Edsger W. Dijkstra. *Communications of the ACM* 53, 8 (2010), 41–47.
[93] Siti Aishah Mohd Selamat, Simant Prakoonwit, Reza Sahandi, Wajid Khan, and Manoharan Ramachandran. 2018. Big data analytics—A review of data-mining models for small and medium enterprises in the transportation sector. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 3 (2018), e1238.
[94] Ammar W. Mohemmed, Mengjie Zhang, and Will N. Browne. 2010. Particle swarm optimisation for outlier detection. In *Proceedings of the Conference on Genetic and Evolutionary Computation.* 83–84.
[95] Md Mohibullah, Md Zakir Hossain, and Mahmudul Hasan. 2015. Comparison of Euclidean distance function and Manhattan distance function using k-mediods. *International Journal of Computer Science and Information Security* 13, 10 (2015), 61.
[96] Brendan Morris and Mohan Trivedi. 2008. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Transactions on Circuits and Systems for Video Technology* 18 (2008), 1114–1127.
[97] Raymond T. Ng and Jiawei Han. 1994. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94).* 144–155.
[98] Adam Nowosielski, Dariusz Frejlichowski, Paweł Forczmański, Katarzyna Gościewska, and Radosław Hofman. 2016. Automatic analysis of vehicle trajectory applied to visual surveillance. *Image Processing and Communications Challenges* 7 (2016), 89–96.
[99] Julian Oehling and David J. Barry. 2019. Using machine learning methods in airline flight data monitoring to generate new operational safety knowledge from existing data. *Safety Science* 114 (2019), 89–104.
[100] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. 2008. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology* 18, 11 (2008), 1544–1554.
[101] Yael Pritch, Alex Rav-Acha, and Shmuel Peleg. 2008. Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008), 1971–1984.
[102] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence—Video to text. In *Proceedings of the IEEE International Conference on Computer Vision.* 4534–4542.
[103] Iq Reviessay Pulshashi, Hyerim Bae, Hyunsuk Choi, and Seunghwan Mun. 2018. Smoothing of trajectory data recorded in harsh environments and detection of outlying trajectories. In *Proceedings of the International Conference on Emerging Databases.* 89–98.

[104] Kun Qin, Yulong Wang, and Bijun Wang. 2018. Detecting anomalous trajectories using the Dempster-Shafer evidence theory considering trajectory features from taxi GNSS data. *Information* 9, 10 (2018), 258.

[105] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record* 29, 2 (2000), 427–438.

[106] Amjad Rehman and Tanzila Saba. 2014. Features extraction for soccer video semantic analysis: Current achievements and remaining issues. *Artificial Intelligence Review* 41 (2014), 451–461.

[107] Nasir Saeed, Tareq Y. Al-Naffouri, and Mohamed-Slim Alouini. 2019. Outlier detection and optimal anchor placement for 3-D underwater optical wireless sensor network localization. *IEEE Transactions on Communications* 67, 1 (2019), 611–622.

[108] Muhammad Aamir Saleem, Waqas Nawaz, Young-Koo Lee, and Sungyoung Lee. 2013. Road segment partitioning towards anomalous trajectory detection for surveillance applications. In *Proceedings of the International Conference on Information Reuse and Integration.* 610–617.

[109] Ignacio San Román, Isaac Martín de Diego, Cristina Conde, and Enrique Cabello. 2019. Outlier trajectory detection through a context-aware distance. *Pattern Analysis and Applications* 22, 3 (2019), 831–839.

[110] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. 2017. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems* 42, 3 (2017), 19.

[111] Erich Schubert, Arthur Zimek, and Hans-Peter Kriegel. 2014. Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection. *Data Mining and Knowledge Discovery* 28, 1 (2014), 190–237.

[112] Shashi Shekhar, Chang-Tien Lu, and Pusheng Zhang. 2001. Detecting graph-based spatial outliers: Algorithms and applications (a summary of results). In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 371–376.

[113] Oscar S. Siordia, Isaac Martín de Diego, Cristina Conde, and Enrique Cabello. 2014. Subjective traffic safety experts' knowledge for driving-risk definition. *IEEE Transactions on Intelligent Transportation Systems* 15, 4 (2014), 1823–1834.

[114] Adam Slowik and Halina Kwasnicka. 2018. Nature inspired methods and their industry applications—Swarm intelligence algorithms. *IEEE Transactions on Industrial Informatics* 14, 3 (2018), 1004–1015.

[115] Guoli Song, Zheng Huang, Yiwen Zhao, Xingang Zhao, Yunhui Liu, Min Bao, Jianda Han, and Peng Li. 2019. A noninvasive system for the automatic detection of gliomas based on hybrid features and PSO-KSVM. *IEEE Access* 7 (2019), 13842–13855.

[116] Qing Song, Wenjie Hu, and Wenfang Xie. 2002. Robust support vector machine with bullet hole image classification. *IEEE Transactions Systems, Man, and Cybernetics* 32 (2002), 440–448.

[117] Lin Sun, Daqing Zhang, Chao Chen, Pablo Samuel Castro, Shijian Li, and Zonghui Wang. 2013. Real time anomalous trajectory detection and analysis. *Mobile Networks and Applications* 18, 3 (2013), 341–356.

[118] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. 2015. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics* 11, 3 (2015), 812–820.

[119] Jussi Tolvi. 2004. Genetic algorithms for outlier detection and variable selection in linear regression models. *Soft Computing* 8, 8 (2004), 527–533.

[120] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. 2020. Spatial crowdsourcing: A survey. *VLDB Journal* 29, 1 (2020), 217–250.

[121] Sarvesh Vishwakarma and Anupam Agrawal. 2013. A survey on activity recognition and behavior understanding in video surveillance. *Visual Computer* 29 (2013), 983–1009.

[122] Tian Wang, Hao Luo, Xi Zheng, and Mande Xie. 2019. Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing. *ACM Transactions on Intelligent Systems and Technology* 10, 6 (2019), 1–19.

[123] Yulong Wang, Kun Qin, Yixiang Chen, and Pengxiang Zhao. 2018. Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi GPS data. *ISPRS International Journal of Geo-Information* 7, 1 (2018), 25.

[124] Li Wei, Eamonn Keogh, and Xiaopeng Xi. 2006. SAXually explicit images: Finding unusual shapes. In *Proceedings of the International Conference on Data Mining.* 711–720.

[125] James M. Whitacre, Tuan Q. Pham, and Ruhul A. Sarker. 2006. Use of statistical outlier detection method in adaptive evolutionary algorithms. In *Proceedings of the Conference on Genetic and Evolutionary Computation.* 1345–1352.

[126] Dumidu Wijayasekara, Ondrej Linda, Milos Manic, and Craig Rieger. 2014. Mining building energy management system data using fuzzy anomaly detection and linguistic descriptions. *IEEE Transactions on Industrial Informatics* 10, 3 (2014), 1829–1840.

[127] Graham Williams, Rohan Baxter, Hongxing He, Simon Hawkins, and Lifang Gu. 2002. A comparative study of RNN for outlier detection in data mining. In *Proceedings of the IEEE International Conference on Data Mining.* 709–712.

[128] Hao Wu, Weiwei Sun, and Baihua Zheng. 2017. A fast trajectory outlier detection approach via driving behavior modeling. In *Proceedings of the ACM Conference on Information and Knowledge Management*. 837–846.

[129] Tao Xiang and Shaogang Gong. 2008. Video behavior profiling for anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2008), 893–908.

[130] Zhe Xiao, Xiuju Fu, Liye Zhang, and Rick Siow Mong Goh. 2020. Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems* 21, 5 (2020), 1796–1825.

[131] Dragomir Yankov, Eamonn Keogh, and Umaa Rebbapragada. 2008. Disk aware discord discovery: Finding unusual time series in terabyte sized datasets. *Knowledge and Information Systems* 17, 2 (2008), 241–262.

[132] Gao Ying, Nie Yiwen, Yang Wei, Xu Hongli, and Huang Liusheng. 2018. Anomalous trajectory detection between regions of interest based on ANPR system. In *Proceedings of the International Conference on Computational Science*. 631–643.

[133] Xia Ying, Zhang Xu, and Wang Guo Yin. 2009. Cluster-based congestion outlier detection method on trajectory data. In *Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery*, Vol. 5. 243–247.

[134] Cao Lei Yu, Yanwei, Elke A. Rundensteiner, and Qin Wang. 2017. Outlier detection over massive-scale trajectory streams. *ACM Transactions on Database Systems* 42, 2 (2017), 10.

[135] Qingying Yu, Yonglong Luo, Chuanming Chen, and Xiaohan Wang. 2018. Trajectory outlier detection approach based on common slices sub-sequence. *Applied Intelligence* 48, 9 (2018), 2661–2680.

[136] Yanwei Yu, Lei Cao, Elke A. Rundensteiner, and Qin Wang. 2014. Detecting moving object outliers in massive-scale trajectory streams. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 422–431.

[137] Mohamed H. Zaki and Tarek Sayed. 2013. A framework for automated road-users classification using movement trajectories. *Transportation Research Part C: Emerging Technologies* 33 (2013), 50–73.

[138] Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. 2011. iBAT: Detecting anomalous taxi trajectories from GPS traces. In *Proceedings of the International Conference on Ubiquitous Computing*. 99–108.

[139] Fan Zhang, Hansaka Angel Dias Edirisinghe Kodituwakku, Wesley Hines, and Jamie Baalis Coble. 2019. Multi-layer data-driven cyber-attack detection system for industrial control systems based on network, system and process data. *IEEE Transactions on Industrial Informatics* 15, 7 (2019), 4362–4369.

[140] Tong Zhang, Di Wen, and Xiaoqing Ding. 2016. Person-based video summarization by tracking and clustering temporal face sequences. In *Proceedings of the IS & TSPIE Electronic Imaging Symposium,* Vol. 8664. 86640O-1–86640O-6.

[141] Tao Zhang, Shuai Zhao, and Junliang Chen. 2018. Ship trajectory outlier detection service system based on collaborative computing. In *Proceedings of the IEEE World Congress on Services*. 15–16.

[142] Yu Zheng. 2015. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology* 6, 3 (2015), 29.

[143] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology* 5, 3 (2014), 38.

[144] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining interesting locations and travel sequences from GPS trajectories. In *Proceedings of the International Conference on World Wide Web*. 791–800.

[145] Zhong Zheng, Soora Rasouli, and Harry Timmermans. 2014. Evaluating the accuracy of GPS-based taxi trajectory records. *Procedia Environmental Sciences* 22 (2014), 186–198.

[146] Chong Zhou and Randy C. Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 665–674.

[147] Jie Zhu, Wei Jiang, An Liu, Guanfeng Liu, and Lei Zhao. 2015. Time-dependent popular routes based trajectory outlier detection. In *Proceedings of the International Conference on Web Information Systems Engineering*. 16–30.

[148] Zhihua Zhu, Di Yao, Jianhui Huang, Hanqiang Li, and Jingping Bi. 2018. Sub-trajectory-and trajectory-neighbor-based outlier detection over trajectory streams. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. 551–563.

[149] Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. 2008. Maximum entropy inverse reinforcement learning. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI'08)*, Vol. 8. 1433–1438.

[150] Arthur Zimek, Ricardo J. G. B. Campello, and Jörg Sander. 2014. Ensembles for unsupervised outlier detection: Challenges and research questions a position paper. *ACM SIGKDD Explorations Newsletter* 15, 1 (2014), 11–22.

[151] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. 2012. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining* 5 (2012), 363–387.

[152] Justin Zobel and Alistair Moffat. 2006. Inverted files for text search engines. *ACM Computing Surveys* 38, 2 (2006), 6.

Received January 2020; revised May 2020; accepted May 2020