Deviation Point Curriculum Learning for Trajectory Outlier Detection in Cooperative Intelligent Transport Systems

Usman Ahmed, Gautam Srivastava, Youcef Djenouri, and Jerry Chun-Wei Lin*

***Abstract*—Cooperative Intelligent Transport Systems (C-ITS) are emerging in the field of transportation systems, which can be used to provide safety, sustainability, efficiency, communication and cooperation between vehicles, roadside units, and traffic command centres. With improved network structure and traffic mobility, a large amount of trajectory-based data is generated. Trajectory-based knowledge graphs help to give semantic and interconnection capabilities for intelligent transport systems. Prior works consider trajectory as the single point of deviation for the individual outliers. However, in real-world transportation systems, trajectory outliers can be seen in the groups, e.g., a group of vehicles that deviates from a single point based on the maintenance of streets in the vicinity of the intelligent transportation system. In this paper, we propose a trajectory deviation point embedding and deep clustering method for outlier detection. We first initiate network structure and nodes' neighbours to construct a structural embedding by preserving nodes relationships. We then implement a method to learn the latent representation of deviation points in road network structures. A hierarchy multilayer graph is designed with a biased random walk to generate a set of sequences. This sequence is implemented to tune the node embeddings. After that, embedding values of the node were averaged to get the trip embedding. Finally, LSTM-based pairwise classification method is initiated to cluster the embedding with similarity-based measures. The results obtained from the experiments indicate that the proposed learning trajectory embedding captured structural identity and increased *F-measure* by 5.06% and 2.4% while compared with generic *Node2Vec* and *Struct2Vec* methods.**

***Index Terms*—Trajectory analysis, outlier detection, data mining, road traffic management, smart city application.**

I. Introduction

Cooperative Intelligent Transport Systems *(C-ITS)* help to provide sustainability, improve safety and comfort by taking advantage of the communications and cooperation of the participants. Traffic management systems are the key to intelligent and modern traffic in Urban centres. The core component in the C-ITS includes the traffic managements system connected with roadside units and vehicles. Due to mobility, the traffic management systems generate a large amount of traffic data. Primarily that data includes city road networks and trajectories. This huge set of trajectories often comes from the sensors networks integrated with C-ITS. Livability and sustainability of the urban domain require accurate and timely knowledge of urban transportation system is essential [1]. For instance, it helps to understand the different paths of a road segment and provide the most reliable information of its traffic management systems to facilitate optimization goal, i.e., low-carbon transportation [2]. Smart cities use methods like Global Positioning System (GPS) receivers, road sensors, and in-vehicle sensors for monitoring traffic conditions [3]. Each moving object produces trajectories, as results in producing an unlimited number of sequence points. The sequence points are used by the high-performance computing resources to analyze the traffic flow. The sequential points data gathering at every road intersection have a higher telecommunication bandwidth and computation capacity to structure the data for monitoring tasks. This data collection and monitoring process become even trickier in network dynamics, e.g., the time variants available bandwidths. Trajectory databases have a lot of real-world applications with respect to mobile traffic networks [4]–[6] and intelligent transportation systems [7]–[10]. Detecting abnormal trajectories has high implications in traffic flow analysis. The topic of this research also revolves around trajectory outlier detection based on dynamic graphs. The outlier detection technique aims to determine the unusual observations from the normal observations [11]. The basic purpose is to extract inconsistent observation from the normal flow [12, 13]. Current methods consider the single view of the outliers in a sub or whole trajectory. Instead of individual outliers, a group of outliers also exists in the sub trajectory or the whole trajectory. Useful and derived patterns are used to detect group trajectory outliers or for deviation points extraction of individuals and groups of trajectory outliers. The novel urban traffic monitoring schemes should be less demanding on the deployable traffic detectors and intelligently adapts to the limited and resource-efficient environments. Instead of monitoring the massive links or network tomography, a smaller number of paths should be compared with that reduces complexity. In addition , network tomography helps to analyze road segments to its neighbour's road segments.

U. Ahmed and J. C. W. Lin are with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, 5063, Bergen, Norway. Email: usman.ahmed@hvl.no, jerrylin@ieee.org. Website: http://ikelab.net. (*Corresponding author: Jerry Chun-Wei Lin)

G. Srivastava is with the Department of Mathematics & Computer Science, Brandon University, Brandon, Canada and with the Research Center for Interneural Computing, China Medical University, Taiwan. Email: SRIVASTAVAG@brandonu.ca

Y. Djenouri is with the Department of Mathematics and Cybernetics, SINTEF Digital, Olso. Email: Youcef.Djenouri@sintef.no

A. Motivation

The motivation behind the proposed work is formulated in Fig. 1, where *Taxis 1,2,3,4* using the same route for the destination. However, *Taxis 5* and *6* deviate from the same point

This is the author accepted version of an article published in IEEE Transactions on intelligent transportation systems http://dx.doi.org/10.1109/TITS.2021.3131793

(a) Single instance (b) Group outlier example 1 (c) Group outlier example 2

Fig. 1: Trajectory outlier detection motivation example

instead of having the same source and destination point as the rest of the *Taxis*. Traditional methods used the user threshold values to find the similarity of the trajectory outliers [11, 14, 15]. The similarity score is calculated of each trajectory and then the fixed user threshold values decide whether the trajectory is an outlier. They do not examine the structure differences between each trajectory. In Fig. 1(b), *Taxis 5* and *6* deviate from the same deviate point but follow two different trajectories. However, in Fig. 1(c), the two *Taxis* deviate from the same point and also follow the same trajectory. Detecting deviation points based on the trajectory analysis helps C-ITS discover useful knowledge about the follow and deviations points. Detecting the individuals and deviation points of the individuals (Fig. 1(a)) helps to identify the potential *Taxi* frauds and individual deviation point *(IDP)*. This helps the C-ITS decision-makers apply appropriate measures and thus better identify potential frauds by placing the perfect positioned surveillance cameras.

Detection of group trajectory in Figs. 1(b) and 1(c), allows the C-ITS to detect *Taxi* outliers. The deviation from the same point and following different trajectories have substantial implications to avoid circumstances on the main trajectory, such as traffic jams rather than *Taxi* fraud. However, a group of *Taxis* outlier with different deviation points with the same trajectory might partner in the taxi fraud. Network structure and temporal information are important in trajectory outlier detection. The major problem in traditional methods [11, 14, 15] is that the existing *Taxis* fraud detection algorithms detect a group of trajectory outliers and spatio-temporal deviation points identification.

B. Contributions

Previous works address the individual trajectory outlier detection method where Group of trajectory outlier (GTO) detection is done based on the deviation points. This paper designs and implements the structural embedding method to assess the structural trajectory similarity between nodes. Two nodes with a similar local structure based on their trajectory will be considered similar. The context is used to build and learn the latent representation for the trajectory of nodes. After embedding the extraction method, we introduce the curriculum learning method to detect the outliers in the extracted trajectory embeddings using a self-supervised way. In particular, we have the following key contributions:

1) We introduce a trajectory embedding method based on the structural similarity to describe the identifiability of the trajectory-based network connected over C-ITS.
2) We propose a curriculum learning-based self-supervised learning algorithm to obtain the deviation point-based trajectory outliers.
3) The embedding structures method helps to detect trajectory outliers and deviation points with increased *F-measure* by 5.06% and 2.4% while compared with *Node2Vec* and *Struct2Vec* methods.

The rest of the paper is arranged as follows. Section II describes the connected works. Section III describes the problem definition and methodology. Section IV discusses the outcomes. Section V describes the finding of the work. Section VII concludes by providing a summary and Section VI future work recommendations.

II. Literature review

Trajectory and outlier detection methods are broken down into two categories, i.e., offline and online methods. The huge data volumes and complex semantics structure in outlier detection have imposed significant technical challenges. Zhang *et al.* [20] proposed a graph-based method for detecting multi-levels of *Taxi* trip outliers in a large-scale urban traffic network. The efficient spatial analysis helps to find the shortest path and centralities measures. The method makes use of the shortest path computation algorithm and a spatial join algorithm. The contraction hierarchy of these algorithms is implemented to snap, pickup, and drop-off locations. Kong *et al.* [16] used the local outlier factor to index score for anomaly detection. Zhu *et al.* [21] proposed the algorithm to detect time-independent outliers. The method utilizes the route of the same source and destination point. A threshold value is set to determine the outliers and normal trajectory. The outliers trajectory extractor from the isolation-based anomalous trajectory algorithm has distinct properties. Zhongjian *et al.* [18] proposed a group

TABLE I: Comparison of the application and method used in the trajectory analysis.

References	Year	Trajectory Data	Application	Methods
Chao *et al.* [14]	2013	Yes	No	iBoat - adaptive window based
Xiangjie *et al.* [16]	2018	Yes	Yes	Statistical learning
Zhipeng *et al.* [17]	2013	Yes	No	Density-based trajectory outlier detection
Zhongjian *et al.* [18]	2017	Yes	No	Clustering
Jiali *et al.* [12]	2018	Yes	Yes	Outlier detection over distributed trajectory streams)
Jiali *et al.* [19]	2017	Yes	No	TF-outlier and MO-outlier detection upon trajectory stream)
Hosseinpoor *et al.* [13]	2018	Yes	Yes	Dempster-shafer - degrees of uncertainty

route method that contains the central points-based clusters. The central points are used as usual routes. The distance-based method is used to compute the trajectories scores. The trajectories that exceed a similarity threshold are termed outliers. Zhou *et al.* [22] identifies *Taxi* fraud by matching the patterns in the taximeter database, i.e., where each trajectory point is metered or unmetered. The method used the stochastic gradient model.

A. Online methods

Online methods detect the sub-trajectory that are significantly different. Chen *et al.* [14] proposed the adaptive working window method to detect the *Taxi* detours. Yu *et al.* [23] also proposed a sub-trajectory-based outlier detection method. The method used a neighbour set of sub-trajectory to calculate the similarity scores. Wu *et al.* [24] introduced the probabilistic model that used the entropy-based inverse reinforcement learning method. The method transforms mapped trajectories into historical trajectory actions. The method used probability threshold values to identify sub-trajectory. Mao *et al.* [19] introduce trajectory fragmentation-based method. Two consecutive points make up each fragment of a trajectory. The local trajectory helps to identify the fragment outliers. Yu *et al.* [25] applied the slice-based outliers method to define sub-trajectory. Slice uses the direction of the route to determine line segments. If the number of neighbours of a trajectory slice is less than a given threshold, it is considered a slice outlier.

B. Group Outlier Detection

There are a few solutions presented to the problem of group outlier detection. Chalapathy *et al.* [26] suggested using a deep generative model to get values for group outliers. The standard back-propagation algorithm is used to estimate the input data by group reference function. Tang *et al.* [27] proposed a contextual outlier detection using a similarity-sharing group of points. These points share similarities in some areas and might differ to some others. The contextual outliers are derived using a statistical significance test that is greater than a certain threshold.

Li *et al.* [28] suggested the assignment of feature weights on each group outlier and the use of chain rule entropy to determine the connection between different groups. Contextual outlier detection in high and sparse dimensional spaces was performed using parallel computing. Other techniques combine individual outliers into similar clusters [27, 29]. Each cluster is then termed as a group of outliers. Soleimani *et al.* [29] suggested a supervised learning approach that combines anomalous patterns when the membership of outliers is previously unknown. This approach has found its applications in document modeling, where non-uniform topics can be detected from a collection of documents.

Human abnormal data is also based on the learning algorithm. The studies used the data mining method to find the correlated patterns for abnormal behaviour and then applied the convolution neural network to learn collective behaviour [11]. Another method used the group trajectory to detect the outliers based on the pruning density method [30]. The moving object and their behaviour are also analyzed that detect the outlier over distributed trajectory streams [15]. The method used the behaviour and local neighbour to explore trajectory distributions [15]. The hidden and abnormal patterns are also analyzed [31] that used the spatial and temporal features of the moving object to detect the abnormality.

The literature studies addressed above are shown in Table I, in which it contains a different trajectory detection methods. However, there is no study conducted that detects deviation points in a group of trajectory outliers. The methods were made to detect the individual outliers and sub-trajectory-based outliers. All methods used the concept of group outlier based on the set of candidate groups rather than individuals where the distribution-based methods are used. However, the distribution does not correspond to real-world data. This paper used the structural trajectory representation and then used the deep unsupervised clustering method to detect trajectories having outliers and deviation points.

III. Problem Statement

A trajectory is the sequence of a point represented by the geographical location (longitude, latitude, and time). For trajectory outlier detection, some preliminary definitions are discussed below.

Definition 1: (Trajectory Database): A set of raw trajectories is a database represented as $T = \{T_1, T_2, \ldots, T_m\}$, where each trajectory T_i is a set of sequences represented as a time-ordered points $(p_1, \ldots, p_n)$, and each point represents geographical location and is considered as a node in graph construction.

Definition 2: (Mapped Trajectory Database): A set represents the sequences of spatiotemporal regions $\Lambda = \{\Lambda_1, \ldots, \Lambda_m\}$, in which each mapped trajectory Λ_i represents a region, and $(R_1, \ldots, R_n)$ can be retrieved by mapping every point in T_i to its closest region R_i.

The mapped trajectory is used as the instance to train the embedding vectors.

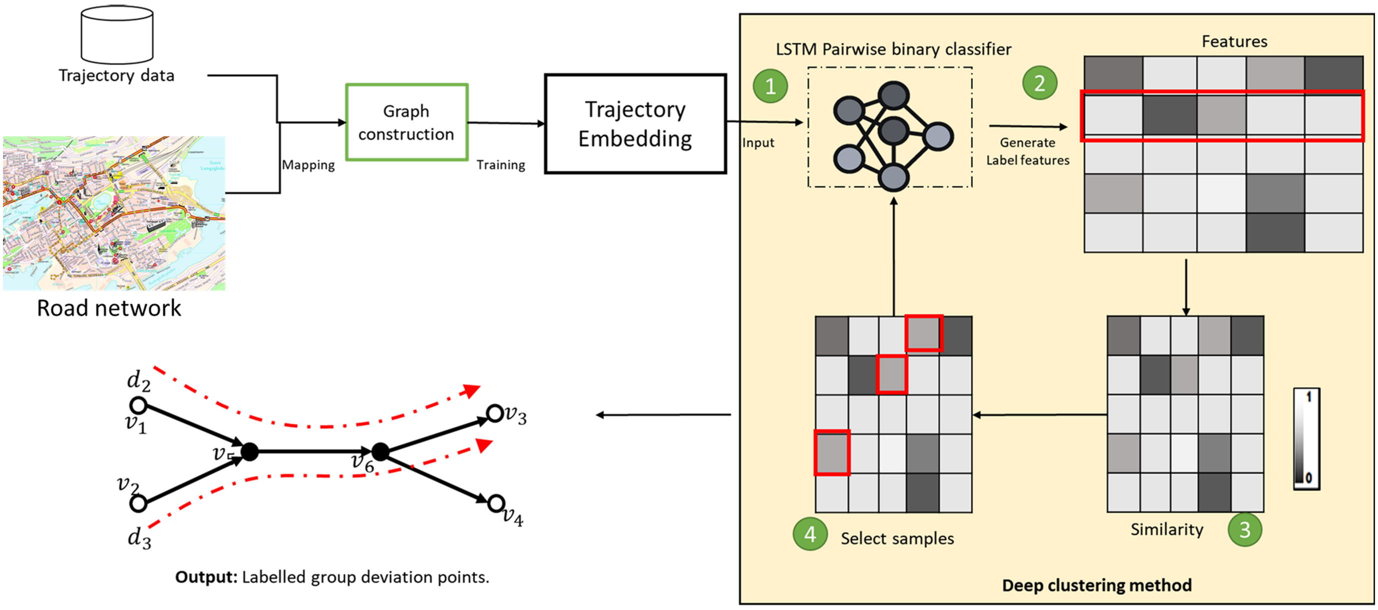


Fig. 2: The designed general framework.

Definition 3: (Trajectory Dissimilarity): Two trajectories in a given graph, $d(\Lambda_i, \Lambda_j)$, is measured by using the learning based method.

The trajectory candidate is a directed graph that represents the potential candidate for having a group of trajectory outliers. These trajectories are measured by using dissimilarity measures. We used the average method to calculate the structure of the group of nodes that appeared in the trajectory.

Definition 4: (Trajectory Candidates): It is a set of reference graph that is given to trajectory outlier detection algorithm, and denoted as $\mathcal{G}^+ = \{\Lambda_1^+, \Lambda_2^+ \dots \Lambda_l^+\}$.

Definition 5: (Group Trajectory Outlier (GTO) detection): It extracts the set of trajectories that represents the deviates from the starting point x with at least one trajectory in $\mathcal{G}$, and x is highly ordering all remaining starting points in $\mathcal{G}$.

A. Deviation point representation clustering

This work proposed the trajectory embedding method that maps the trajectory data into latent representation. The trajectory embedding maps graph nodes to the vector into real numbers in multidimensional space. To extract valuable insights about the nodes and their edges, we used multi hops ring structure-based method to preserve the structural representation. The learnt embedding vectors are then used to combine by the group trajectory outlier clustering method. The purpose of structurally preserve trajectory embedding is to expand the knowledge based on the k hops. So that we can detect the deviation points, which helps the adaptive clustering method to detect a group of trajectories based on the deviation points. The flowchart of the developed model is mentioned in Fig. 2. The input is unlabeled trajectory data, where we first train the proposed trajectory embeddings based on the structure and deviation points. The feature labels are generated for the pairwise classification models. We then calculate the cosine similarity and assign a label for each trip. We selected and omitted training samples for each batch. We utilize the labeled data for the binary pairwise classification model to reduce its error. After that, we iteratively perform the samples for model training. Clustering favours the trip that has the structure similarity and same deviation points of the label features.

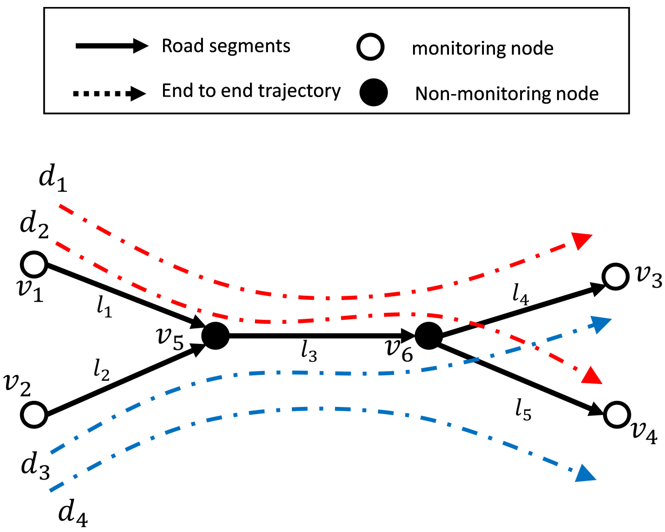


Fig. 3: Illustration of the Cooperative Intelligent Transport System where road network and end-to-end trajectory are monitored. It has five road segments to monitor while only four cameras (i.e., monitoring nodes) are deployed.

B. Graph Construction

The road network is mentioned in Fig. 3 as a directed acyclic graph $G = (V, L)$, where trajectories of the vehicle are monitored by following the road segments. V and L respectively showed the set of road node and segments. As illustrated in Fig. 3, we have $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $L = \{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5\}$. Some nodes are monitored by cameras networks called as monitoring nodes η. When nodes originates a trajectory and follows road segments, deviation point trajectory are monitored in $\mathcal{D}$. For example in Fig. 3, we used $\eta = 4$ monitoring nodes, and get the trajectory of the different trips $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$, respectively. When a trajectory ℓ_j with respect to nodes v_k), its deviation point can be monitored asd_i, we then mark their route relationship by $\ell_j \in d_i$ and $v_k \in d_i$). We considered the trajectory as a sequential directed acyclic graph G where location points represent space and time. In this research, we exclude the round trip travels. The edge between each node represents intermediate points, as explained in the Fig. 3, where trajectory is represented from the $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$ intermediate nodes.

C. Trajectory structural embedding

After the road network, latent learning representation is used for nodes that consists of the following operations.

1) A node's structural similarity between vertex pairs is determined. We used the hierarchy-based multi-edge deep neural network, which helps assess the structural trajectory similarity at each level.
2) The weighted multilayer graph is constructed where each node is represented at every layer. The layer of the deep neural network responds to the hierarchy features that represents trajectory similarity. The edge weights on every road segments lines within each layer are inversely proportional to its structural similarity.
3) We used biased random walk on the multilayer hierarchy graph to generate the road segments sequences. The sequences are the set of road segments that are more trajectory structurally similar.
4) We used the SkipGram method to get a latent representation by giving the context mentioned above.

We used the ordered degree sequence of a set $S \subset V$ of nodes. Let $T_k(x)$ nodes represents the hop count of k distances in G. For instance, $T_1(x)$ denotes the set of neighbors of vertex x at distance of 1. $T_k(x)$ denotes the trajectory growth of nodes at distance k. By comparing the ordered degree sequences of the trajectory nodes from both x and y (two nodes in the node network), we impose a hierarchy to measure structural similarity. We denote the learning function $f_k(x, y)$ represents the trajectory structural distance between x and y. We consider their neighbourhoods k (all road network nodes at a distance less than or equal to k. The function is defined in Equation 1.

$$\begin{gathered} f_k(x,y) = f_{k-1}(x,y) + g\left(s\left(T_k(x)\right), s\left(T_k(y)\right)\right) \\ k \geq 0 \text{ and } \left|T_k(x)\right|, \left|T_k(y)\right| > 0 \end{gathered} \tag{1}$$

The function of Equation 1 is only defined when both x or y have the edge at a distance k. The distance between ordered degree sequences can be measured by the $f_k(x, y)$. Using the trajectory growth at a distance, k helps compute the degree sequences of nodes at the same distance from x and y. We used Dynamic Time Warping (DTW) to calculate the distance between two ordered degree sequences. This method helps extract useful distances that cope better with sequences of different sizes and loosely compress sequence patterns [18, 20]. The DTW helps to find optimal alignment between the trajectory growth sequences of x and y, given a distance function $d(x, y)$ for each element in the sequences, DTW matched the sequence in the way that the sum of the distances between match elements is minimized [32]. Since trajectory growth is represented by a node's degree sequences with a neighbour, we then used the distance function mentioned in Equation 2.

$$d(a,b) = \frac{\max(x,y)}{\min(x,y)} - 1 \tag{2}$$

Two identical nodes with ordered sequences will have zero distance ($x = y$ then $d(x, y) = 0$). To construct the contexts, the multilayer weighted graph encodes the nodes as deviation points trajectory. $G = (V, L)$ is the road network connected at the k^* hops diameter. We define the multilayer graph by using the k -hop neighbourhoods of the nodes. The weight of the nodes is assigned by using the function defined above. The edge weight $\begin{pmatrix} n \\ 2 \end{pmatrix}$ edges in a layer is given by Equation 3.

$$w_k(x,y) = e^{-f_k(x,y)} \tag{3}$$

For weighted edges, we have nk^* vertices and at most $k^* \begin{pmatrix} n \\ 2 \end{pmatrix} + 2n\left(k^* - 1\right)$.

The multilayer graph generates the contextual information for the trajectory deviation point. The structural similarity based on the trajectory deviation points nodes required absolutely no label information. We used a biased random walk that moves around the multilayer graph with random choices and weighted sequences. The random walk first decides to move around or stay at each layer with probability ($q > 0$) the random walk stays in the current layer). The probability of node u to node v in layer k for staying in the current layer is given by Equation 4.

$$p_k(x,y) = \frac{e^{-f_k(x,y)}}{Z_k(x)}, \tag{4}$$

where $Z_k(x)$ is the normalization factor for vertex x in layer k, which is simply given by Equation 5.

$$Z_k(x) = \sum_{\substack{v \in V \\ v \neq x}} e^{-f_k(x,y)} \tag{5}$$

The random walk method prefers to step only those nodes that are trajectory deviation point-based structurally similar. This will result in the context of a node u in structurally similar nodes without their label information and position in the network. The node x in V starts the random walk in its corresponding vertex at layer 0; the walks have a fixed shorter length (number of steps). The process is repeated for a certain number of times and giving rise to multiple independent walks (multiple contexts for node x). We used the Skip Gram method for representation learning. The generated sequence is used (biased random walks in a multilayer graph). The skip-gram method aims to maximize the context in a sequence, where a node's context is given by window size w centre to it. We also used the hierarchical softmax, where conditional symbol probabilities are used by binary tree classifiers. Each node is assigned by a specific path in the classification tree in V. In this setting, the definition is then given in the Equation 6.

$$P\left(v_x \mid v_y\right)=\prod_{k=1}^{h} C\left(n\left(v_x, k\right), v_y\right), \tag{6}$$

where C is a binary classifier presented in every node in the tree.

D. Binary pairwise classification model

To cluster the sequence of trajectory represented as trips, we trained the binary pairwise deep neural network. A recurrent neural network (RNN) with a gated unit is used. The LSTM network preserved the memory for long-distance information and performed well for sequential tasks. We used the element-wise average method in the designed model. Equation 7 is then used to determine the learning function F for relation R.

$$F\left(\boldsymbol{v}^x, \boldsymbol{v}^y\right)=\begin{cases}1 \text{ (if } \boldsymbol{v}^x \text{ and } \boldsymbol{v}^y \text{ satisfy } R) \\ -1 \text{ (otherwise)}\end{cases} \tag{7}$$

Variable R represents trajectory based on deviation points. If the trip follows normal trajectory, they are grouped; otherwise, they are grouped as abnormal. In this research, each trip consists of different nodes. The obtained embedding was discussed in the previous section. Then, we used the embedding average method to represent a collection of nodes as a single trip for each trip. The binary pairwise classification model used the label features from trajectory averaging. Then, by using cosine similarity, we select the more structurally similar together cosine similarity trips. We used the optimization method based on the degree to reduce the search space and run a bigger network [32]. $g\left(\mathbf{v}_x, \mathbf{v}_y; \mathbf{w}\right)$ can be formulated in Equations 8 and 9, respectively.

$$g\left(\mathbf{v}_x, \mathbf{v}_y; \mathbf{w}\right)=f\left(\mathbf{v}_x; \mathbf{w}\right) \cdot f\left(\mathbf{v}_y; \mathbf{w}\right) \tag{8}$$

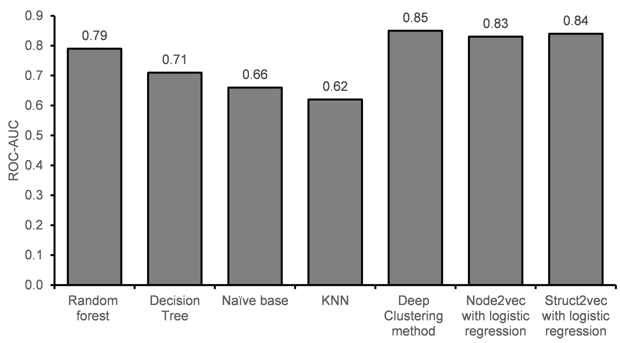
$$R_{xy}:=\begin{cases}1, \text{ if } \mathbf{l}_x \cdot \mathbf{l}_y \geq u(\lambda) \\ 0, \text{ if } \mathbf{l}_i \cdot \mathbf{l}_y < l(\lambda), \quad i, j=1, \cdots, n \\ \text{None, otherwise,}\end{cases} \tag{9}$$

where $u(\lambda)$ and $l(\lambda)$ are used to control the selection of similar and dissimilar samples. "None" represents the omitted training samples $(\mathbf{v}_x, \mathbf{v}_y, R_{xy})$. We attempt to control clustering production with curriculum learning by increasing the samples per batch [33]. The reason is that trajectories that are very similar together having a very high likelihood to be selected in the training samples. Then RNN progress in finding the optimized trip feature labels is performed gradually by taking the batch of difficult samples. In the clustering process, λ is gradually increased. Moreover, $u(\lambda)=l(\lambda)$ is satisfied iff all the samples are used for training. Algorithm 1 explains the details of each step, where trips with nodes information are given, training embedding f_w and λ are calculated based on the gradient values (*Algorithm 1*, *Input*). $u(\lambda)$ and $l(\lambda)$ are sample controlling methods (*Algorithm 1*, *Input*). We start with m samples. After that, the first small batch with average embedding is selected (*Algorithm 1*, *Lines 1 to 4*), we then calculate similarity to get the label of the trips (*Algorithm 1*, *Line 5*). After that, we apply the gradient method for updating (*Algorithm 1*, *Line 6*). We used the Argmax method to get the clustering classes for the final prediction of the output deviation points for the testing samples (*Algorithm 1*, *Lines 8 to 10*).

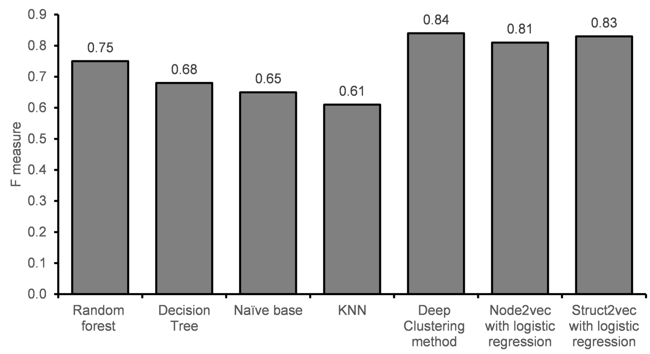
Algorithm 1 Deep clustering using trajectory embedding

INPUT: $T=\{t_i\}_{i=1}^{n}$, λ, $u(\lambda)$, $l(\lambda)$, n represents number of trajectory, w represents embedding size, m represents numbers of instances per batch.
OUTPUT: Cluster label c_i of $t_i \in Trip$

1: **while** $K \leq \{1, 2, ; \frac{n}{m}\}$ **do**
2: Select training samples from T;
3: Apply average method using the embedding;
4: Calculate similarity using Eqs. 8 and 9;
5: Update λ by using the gradient descent algorithm;
6: **end while**
7: **while** $T_i \in Trip$ **do**
8: $\{l_i\}=F(T_i; w)$;
9: $\{c_i\}=argmax_h(l_{ih})$;
10: **end while**
11: **Return** Cluster label c_i.



(a) ROC-AUC

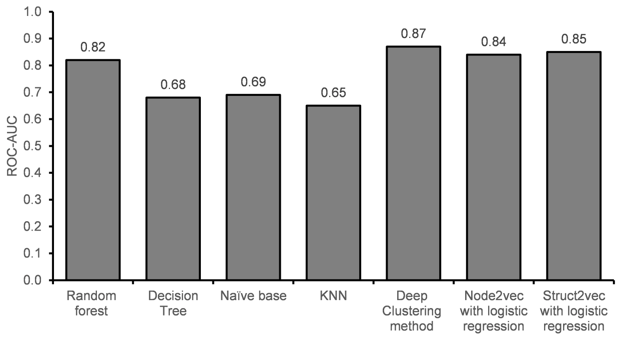


(b) F-measure

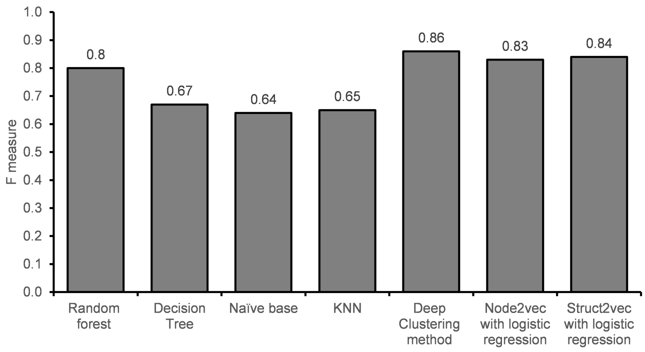
Fig. 4: Performance comparison of the intelligent dataset with deep clustering method.

IV. Experimental Results

Rigorous experiments have been executed to assess the proposed algorithm on different trajectory databases in four steps. In all experiments, the period was set to 5 minutes. A $64-$bit computer was used for the serial implementation, which features a core interval i7 processor running Windows 10 and 16 GB of RAM. We have injected simulated GTOs in our experiments because the trajectory data sets available are not relatable in real-world scenarios. The properties of our simulated dataset are Injecting Individual trajectory outliers: Noise was added frequently with a probability $p \sim \mathcal{U}(0.8, 1.0)$ and a given threshold for generating individual trajectory outliers. For the Injecting GTOs, noise was added several times to the individual trajectory outliers with a probability $p \sim \mathcal{U}(0, 1.0)$ and a threshold. The initial noise points for trajectory outliers are regarded as deviation points and labeled.



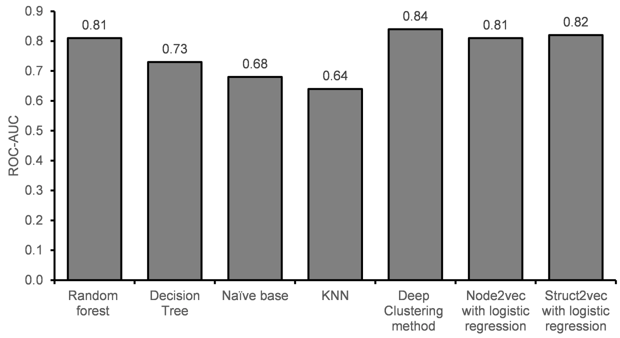
(a) ROC-AUC



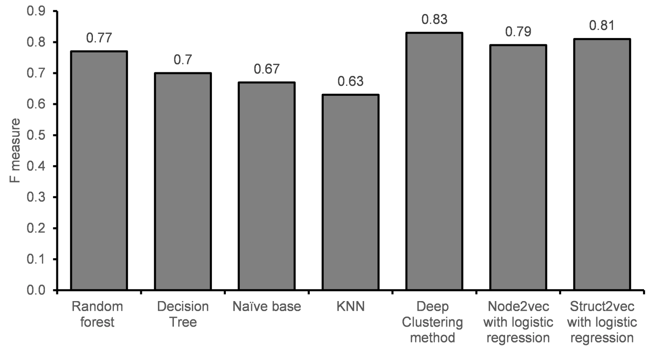
(b) F-measure

Fig. 5: Performance comparison of the climate dataset with deep clustering method.

We used Random Forest, Naïve Bayes, K-nearest neighbour, and decision tree for comparison [34]. The random forest used the ensemble of the decision tree model and divided each tree for random features. The decision tree used the tree-based structure containing roots, nodes, and leaf nodes. The decision-making ability is applied to the internal nodes. Classification is performed based on the first nodes. The naive-based classifier used the probabilistic model, which takes feature probability and calculates likelihood to classify instance class. KNN used the neighbour feature vector values to predict the output class. It is instance-based learning, also known as the lazy learning method. The following experimentation makes use of *F-measure* and *ROC-AUC*, which are usually used for the evaluation of the outlier detection methods.



(a) ROC-AUC



(b) F-measure

Fig. 6: Performance comparison of the environment dataset with deep clustering method.

A. Data Description

We used three datasets, i.e., Intelligent Transportation, Climate Change, and Environment. The intelligent transportation dataset has the real trajectories derived from 01/07/2013 to 30/06/2014 of 442 *Taxi* in Porto, Portugal, and the database from the ECML PKDD 2015 competition [1] has been used in these evaluations. Further information about this trajectory database can be found in [35]. The climate change dataset has Atlantic hurricanes track [35], which holds the parameters of latitude, longitude, maximum sustained surface wind, and minimum sea-level pressure of hurricane trajectories in the USA six hourly intervals for the period from 1851 to 2018. The number of trajectories of this dataset is $52,775$. The environment data used Starkey Projects, where animal movement data is also included in the dataset, displayed using the radio-telemetry locations of elk, deer, and cattle, collected from 1989 to 1999. The locations have been saved at 30-minute intervals. It has 100 trajectories and more than $40,000$ different points. This is a sparse dataset.

B. Results

Several experiments have been executed by using the *F-measure* and *ROC-AUC* performance measure. It has been

[1] https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i

TABLE II: The results for the intelligent transport, environment, and climate datasets.

Dataset	**Intelligent Transport**		**Environment**		**Climate**	
Classifier	F-measure	ROC-AUC	F-measure	ROC-AUC	F-measure	ROC-AUC
Random forest	0.75	0.79	0.77	0.81	0.80	0.82
Decision Tree	0.68	0.71	0.70	0.73	0.67	0.68
Naïve base	0.65	0.66	0.67	0.68	0.64	0.69
KNN	0.61	0.62	0.63	0.64	0.65	0.65
Deep clustering method	**0.84**	**0.85**	**0.83**	**0.84**	**0.86**	**0.87**
Node2vec with logistic regression [36]	0.81	0.83	0.79	0.81	0.83	0.84
Struct2vec with logistic regression [32]	0.83	0.84	0.81	0.82	0.84	0.85

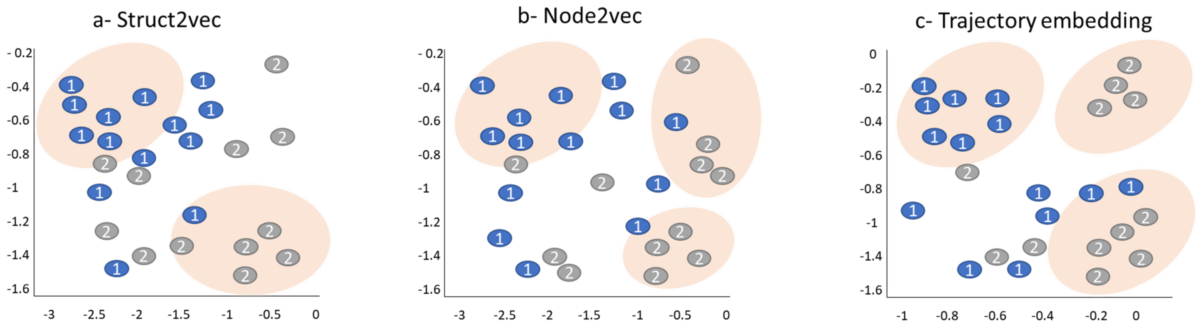


Fig. 7: The embedding visualization. class $1 : normal\ trajectory, 2 : abnormal\ trajectory$

observed from the results that the trajectory databases (intelligent transportation and environment), deep clustering method model outperformed classification. Although both models using the same embedding, the deep clustering-based model can achieve optimized results. We learn a latent representation of the trajectory network using the proposed method, struc2vec [32] and node2vec [36] using the grid search method. This method does not require any labels. The latent representation for each node becomes a feature for the pairwise classification method for the proposed clustering method. At the same time, we used logistic regression for the struc2vec and node2vec learning methods. The traditional classifier used the trained embedding for the classification of the trajectories.

Fig. 4 and Table II illustrated the F-measure and ROCAUC of the traditional classifiers. The results illustrate that the proposed dynamic method outperforms the other heuristic regarding the F-measure. It is valid for trajectory databases. However, solutions formed based on the traditional classification performed better by the embedding method. The illustrations also show that solutions exhibited high performance based on the clustering approach model, and it required additional time as two deep nets were trained.

Similarly, climate data also be used to classify trajectory as mentioned in Fig. 5 and Table II. The proposed model has reached good accuracy. The climate dataset has more deviation points that result in more error rates for trajectories. In the climate dataset, deep clustering achieved the highest ROCAUC of 0.86. This illustrates that having more deviation points can help the learning algorithm perform better due to the structural similarity of the embedding. The struc2vec and node2vec also performed better and achieved 0.83 and 0.84 F-measures.

In Fig. 6 and Table II, we used an environmental dataset and analyzed the proposed method. The proposed method achieves 0.83 accuracy, whereas the struc2vec method achieved 0.81 *F-measure*. This result is more evident for solutions formulated on deep clustering learning. The proposed method was able to group the correct class by using train embedding. The knowledge graph can further be used to increase the training instances.

We visualized the trajectory embedding to its clustering class $1 : normal, 2 : abnormal$ shown in Fig 7. The node2vec and struc2vec fail to group in the latent space trips that are structurally similar (mirrored nodes). The proposed method can learn the features that correctly identify the deviation points and node identity. Mirror pairs or nodes representing the same trips stay close in the latent space, and averaging the trip nodes represents the complex structural hierarchy in the representation of groups.

V. Discussion

The conventional trajectory outlier detection methods detect individual trajectory outliers, whereas the algorithms in this paper detect group trajectory outliers. Introducing approaches from different fields has helped to improve the detection of a group of trajectory outliers. The concepts applied include node-based clustering, feature analysis, evolutionary-based optimization method, and stack-based learning. The GTO solutions discussed in this paper can be an application where numerous trajectories are involved, like the smart city application and urban analysis. This paper uses trajectory deviation point structural embedding to detect GTO. However, there is still room for improvement and more exploration in this area. Applying data mining and machine learning approaches to application domains requires methodological refinement and adaptation [29]. We recommend that advanced techniques can be used for GTO, such as incorporating traditional outlier detection techniques (i.e., Local Outlier Factor). New tools and simulation techniques should be developed for better visualization and understanding of the GTOs. GTOs should be used in other fields such as climate change analysis. Hurricanes can be analyzed using GTOs to find a group of hurricane trajectories that deviates from normal hurricane trajectories. This can help save many by pointing out areas that are more likely to get affected by hurricanes.

VI. Future Work

We will use the proposed method with an attention network to handle missing ground truth labels in the future. With the usage of active learning along with the proposed model, data labeling and handling missing values are to be solved. This method can be further extended to labeled and benchmark public trajectory outlier detection problems. The improved embedding can also be used to evaluate the internal ranking for the trajectory outliers. Some extensions of the proposed model can be applied in the fields of intelligent transport systems: trajectory outliers correlations, the dissimilarity between-group trajectories, processing large and big trajectory under constraints with the usage of reduced embedding vectors.

VII. Conclusion

Deviation points in trajectory data is a concept of symmetry in the network. The concept is strongly related to important problems in social sciences. In this paper, a trajectory-based embedding method is designed where node structure concerning its neighbours is considered. The concept of the hierarchical metric is initiated by the order of the degree sequence of nodes. The weight of the multilayer graph with similarity metrics generates the context. We show that structural embedding in the deviation point analysis and group-based trip analysis plays a vital role. The clustering method then uses node embedding to adopt and cluster the nodes according to the latent representation. The learning method adopts a dynamic network behaviour. Experimental results show that the trained model outperforms traditional classifiers and other embeddings with an F-measure of 0.87. In the future, a network-based node structural similarity method will include betweenness, closeness, and cliques analysis for further analysis and discussion.

Acknowledgment

This research is partially supported by the National Centre for Research and Development under the project Automated Guided Vehicles integrated with Collaborative Robots for Smart Industry Perspective and the Project Contract no. is: NOR/POLNOR/CoBotAGV/0027/2019 -00.

References

[1] H. Song, R. Srinivasan, T. Sookoor, and S. Jeschke, *Smart cities: foundations, principles, and applications*. John Wiley & Sons, 2017.

[2] A. Choudhury, U. Bhattacharya, and R. Chaki, "Dynamic scheduling of traffic signal (dsts) management in urban area network," in *IFIP International Conference on Computer Information Systems and Industrial Management*. Springer, 2019, pp. 132–143.

[3] Y. Djenouri, A. Belhadi, J. C.-W. Lin, and A. Cano, "Adapted k-nearest neighbors for detecting anomalies on spatio–temporal traffic flow," *IEEE Access*, vol. 7, pp. 10 015–10 027, 2019.

[4] M. Ghahramani, M. Zhou, and C. T. Hon, "Mobile phone data analysis: A spatial exploration toward hotspot detection," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 351–362, 2018.

[5] T. Kieu, B. Yang, and C. S. Jensen, "Outlier detection for multidimensional time series using deep neural networks," in *2018 19th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2018, pp. 125–134.

[6] H. Senaratne, M. Mueller, M. Behrisch, F. Lalanne, J. Bustos-Jiménez, J. Schneidewind, D. Keim, and T. Schreck, "Urban mobility analysis with mobile network data: A visual analytics approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1537–1546, 2017.

[7] G. Ajaeiya, I. H. Elhajj, A. Chehab, A. Kayssi, and M. Kneppers, "Mobile apps identification based on network flows," *Knowledge and Information Systems*, vol. 55, no. 3, pp. 771–796, 2018.

[8] N. Cao, C. Lin, Q. Zhu, Y.-R. Lin, X. Teng, and X. Wen, "Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 23–33, 2017.

[9] I. Kalamaras, A. Zamichos, A. Salamanis, A. Drosou, D. D. Kehagias, G. Margaritis, S. Papadopoulos, and D. Tzovaras, "An interactive visual analytics platform for smart intelligent transportation systems management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 487–496, 2017.

[10] E. Karbab, D. Djenouri, S. Boulkaboul, and A. Bagula, "Car park management with networked wireless sensors and active rfid," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*. IEEE, 2015, pp. 373–378.

[11] A. Belhadi, Y. Djenouri, G. Srivastava, D. Djenouri, J. C.-W. Lin, and G. Fortino, "Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection," *Information Fusion*, vol. 65, pp. 13–20, 2021.

[12] J. Mao, P. Sun, C. Jin, and A. Zhou, "Outlier detection over distributed trajectory streams," in *Proceedings of the 2018 SIAM international conference on data mining*. SIAM, 2018, pp. 64–72.

[13] A. H. Milaghardan, R. A. Abbaspour, and C. Claramunt, "A dempster-shafer based approach to the detection of trajectory stop points," *Computers, Environment and Urban Systems*, vol. 70, pp. 189–196, 2018.

[14] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, and Z. Wang, "iboat: Isolation-based online anomalous trajectory detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 806–818, 2013.

[15] J. Mao, J. Liu, C. Jin, and A. Zhou, "Feature grouping–based trajectory outlier detection over distributed streams," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, no. 2, pp. 1–23, 2021.

[16] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: Long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[17] Z. Liu, D. Pi, and J. Jiang, "Density-based trajectory outlier detection algorithm," *Journal of Systems Engineering and Electronics*, vol. 24, no. 2, pp. 335–340, 2013.

[18] Z. Lv, J. Xu, P. Zhao, G. Liu, L. Zhao, and X. Zhou, "Outlier trajectory detection: A trajectory analytics based approach," in *International Conference on Database Systems for Advanced Applications*. Springer, 2017, pp. 231–246.

[19] J. Mao, T. Wang, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2696–2709, 2017.

[20] J. Zhang, "Smarter outlier detection and deeper understanding of large-scale taxi trip records," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing - UrbComp '12*. ACM Press, 2012.

[21] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Time-dependent popular routes based trajectory outlier detection," in *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 16–30.

[22] X. Zhou, Y. Ding, F. Peng, Q. Luo, and L. M. Ni, "Detecting unmetered taxi rides from trajectory data," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017.

[23] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, "Outlier detection over massive-scale trajectory streams," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 2, pp. 1–33, 2017.

[24] H. Wu, W. Sun, and B. Zheng, "A fast trajectory outlier detection approach via driving behavior modeling," in *Proceedings of the 2017*

ACM on Conference on Information and Knowledge Management*, 2017, pp. 837–846.
[25] Q. Yu, Y. Luo, C. Chen, and X. Wang, "Trajectory outlier detection approach based on common slices sub-sequence," *Applied Intelligence*, vol. 48, no. 9, pp. 2661–2680, 2018.
[26] R. Chalapathy, E. Toth, and S. Chawla, "Group anomaly detection using deep generative models," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 173–189.
[27] G. Tang, J. Pei, J. Bailey, and G. Dong, "Mining multidimensional contextual outliers from categorical relational data," *Intelligent Data Analysis*, vol. 19, no. 5, pp. 1171–1192, 2015.
[28] J. Li, J. Zhang, N. Pang, and X. Qin, "Weighted outlier detection of high-dimensional categorical data using feature grouping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4295–4308, 2018.
[29] H. Soleimani and D. J. Miller, "Atd: Anomalous topic discovery in high dimensional discrete data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2267–2280, 2016.
[30] Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "Trajectory outlier detection: New problems and solutions for smart cities," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 2, pp. 1–28, 2021.
[31] X. Cai, B. Aydin, A. Ji, and R. Angryk, "A framework for local outlier detection from spatio-temporal trajectory datasets," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5682–5689.
[32] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 385–394.
[33] S. Pan, P. Li, C. Yi, D. Zeng, Y.-C. Liang, and G. Hu, "Edge intelligence empowered urban traffic monitoring: A network tomography perspective," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2198–2211, 2020.
[34] U. Ahmed, J. C.-W. Lin, G. Srivastava, and M. Aleem, "A load balance multi-scheduling model for opencl kernel tasks in an integrated cluster," *Soft Computing*, vol. 25, no. 1, pp. 407–420, 2021.
[35] Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "Trajectory outlier detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 2, pp. 1–28, 2021.
[36] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

Usman Ahmed Usman Ahmed is a PhD candidate at the Western Norway University of Applied Sciences (HVL). He has rich experience in building and scaling high-performance systems based on data mining, natural language processing, and machine learning. His research interests are sequential data mining, heterogeneous computing, natural language processing, recommendation systems, and machine learning.

Gautam Srivastava was awarded his B.Sc. degree from Briar Cliff University in the U.S.A. in the year 2004, followed by his M.Sc. and Ph.D. degrees from the University of Victoria in Victoria, British Columbia, Canada in the years 2006 and 2012, respectively. He then taught for 3 years at the University of Victoria in the Department of Computer Science, where he was regarded as one of the top undergraduate professors in the Computer Science Course Instruction at the University. From there in the year 2014, he joined a tenure-track position at Brandon University in Brandon, Manitoba, Canada, where he currently is active in various professional and scholarly activities. He was promoted to the rank of Associate Professor in January 2018. In his 5 years as a research academic, he has published a total of 140 papers in high-impact conferences in many countries and in high status journals (SCI, SCIE) and has also delivered invited guest lectures on Big Data, Cloud Computing, Internet of Things, and Cryptography at many universities worldwide. He is an IEEE Senior Member and also an Associate editor of the world renowned IEEE Access journal.

Youcef Djenouri received the Ph.D. degree in computer engineering from the University of Science and Technology USTHB Algiers, Algeria, in 2014. In 2017, he was a Postdoctoral Research with Southern Denmark University, where he has working on urban traffic data analysis. He is currently at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He also granted a postdoctoral fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM). He is working on topics related to artificial intelligence and data mining, with focus on association rules mining, frequent itemsets mining, parallel computing, swarm and evolutionary algorithms and pruning association rules. He has published more than 60 refereed research articles, in the areas of data mining, parallel computing, and artificial intelligence.

Jerry Chun-Wei Lin received his Ph.D. from the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan in 2010. He is currently a full Professor with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, Bergen, Norway. He has published more than 450+ research articles in refereed journals (IEEE TKDE, IEEE TFS, IEEE TNNLS, IEEE TCYB, IEEE TII, IEEE TITS, IEEE TNSE, IEEE TETCI, IEEE SysJ, IEEE SensJ, IEEE IOTJ, ACM TKDD, ACM TDS, ACM TMIS, ACM TOIT, ACM TIST, ACM TOSN) and international conferences (IEEE ICDE, IEEE ICDM, PKDD, PAKDD). His research interests include data mining, soft computing, artificial intelligence and machine learning, and privacy preserving and security technologies. He is the Editor-in-Chief of the International Journal of Data Science and Pattern Recognition, the Guest Editor/Associate Editor of IEEE TFS, IEEE TII, IEEE TITS, IEEE JBHI, ACM TMIS, ACM TOIT, ACM TALLIP, among others. He is the Fellow of IET (FIET), ACM Distinguished Member and IEEE Senior Member.