Deep Fuzzy Contrast-Set Deviation Point Representation and Trajectory Detection

Usman Ahmed, Jerry Chun-Wei Lin, *Senior Member, IEEE*, and Gautam Srivastava, *Senior Member, IEEE*

***Abstract*—Cooperative intelligent transportation systems (ITS) are on the rise in the field of transportation. The trajectory-based knowledge graph enables the ITS to have semantic and connectivity capabilities. This article presents the approach of embedding trajectory deviation points and deep clustering. We constructed the structural embedding by maintaining the relationship between the nodes based on the network structure and the neighbors of the nodes. This approach was then used to learn the latent representation based on the deviation points in the road network structure. We generated a set of sequences using a hierarchical multilayer network and a biased random walk. This research proposes a fuzzy contrast-based model that identifies deviation points using the deep network for weighted position nodes. This sequence is used to fine-tune the embedding of the nodes. We then averaged the embedding values of the nodes to obtain the travel embedding. Next, we extracted the embedding of the contrast set using a pairwise classification approach based on similarity metrics. Numerical studies show that the proposed learning trajectory embedding approach successfully captures the structural identity and outperforms competing strategies. The deep contrast set approach enables highly accurate detection of outliers in the trajectory and deviation locations.**

***Index Terms*—Outlier detection, road traffic management, sustainable transportation, trajectory analysis.**

I. Introduction

To ENSURE the urban domain's viability and livability, a precise and timely understanding of any urban-based transportation system can be critical [1]. As an example, it aids in comprehending the many courses a road section takes and provides the most trustworthy information about its traffic management systems, facilitating optimization goals such as low-carbon transportation. Smart cities monitor traffic using systems such as global positioning system (GPS) receivers, road sensors, and in-vehicle sensors [2], [3]. Every moving object generates trajectories and the resulting sequence points, which are infinite. The sequence points were analyzed using high-performance computing resources. Collecting data from sequential points at each road intersection requires greater transmission bandwidth and more computing capacity to organize the data in a manner suitable for monitoring purposes. When the dynamics of the network are taken into account, this data collection and monitoring procedure becomes even more complicated, such as time variations in available bandwidths. The trajectory database has a wide range of real-world applications in mobile traffic networks [4], [5], [6] and intelligent transportation systems (ITSs) [7], [8], [9]. The detection of aberrant trajectories has a significant impact on traffic flow studies. Additionally, this research focuses on detecting trajectory outliers using dynamic graphs. The outlier identification technique seeks to distinguish extraordinary from typical observations [10], [11], [12].

Livability and the sustainability of any usable urban domain, may require an accurate and timely knowledge of the urban-based transportation systems [1]. As an example, it can help to gain an understanding of many different paths of roads and avenues and be able to provide a collection of reliable information on the traffic management systems for the facilitation of optimized goals. One such example is through low-carbon-based transportation as eluded to in [2]. Methods such as GPS, receivers, traffic and road use, and vehicular sensors for traffic monitoring are used in modern smart cities [3]. Trajectories are produced by all moving objects in the smart city, which creates a close to an infinite sequence of points to analyze. The point sequence or set of points can be used in high-performing computational resources to analyze traffic flow. The set of points is gathered through road intersections using a higher level of telecommunication bandwidth and a larger computational capacity for structuring the data to be used in any monitoring tasks. Data collection operations and the entire monitoring process itself, can become even trickier due to the dynamics of networks. For example, the variation in the time of available bandwidth can cause issues. All trajectory databases are used in many real-life applications especially dealing with networks for mobile traffic control [4], [5], [6]. Several studies have been investigated for trajectory databases in various ITS [7], [8], [9], [13], [14]. The detection of abnormal trajectories then can have severe implications for any traffic flow analysis. Instead of the previous works, our methodology revolves around trajectory outlier detection using dynamic graphs. The designed novel trajectory outlier

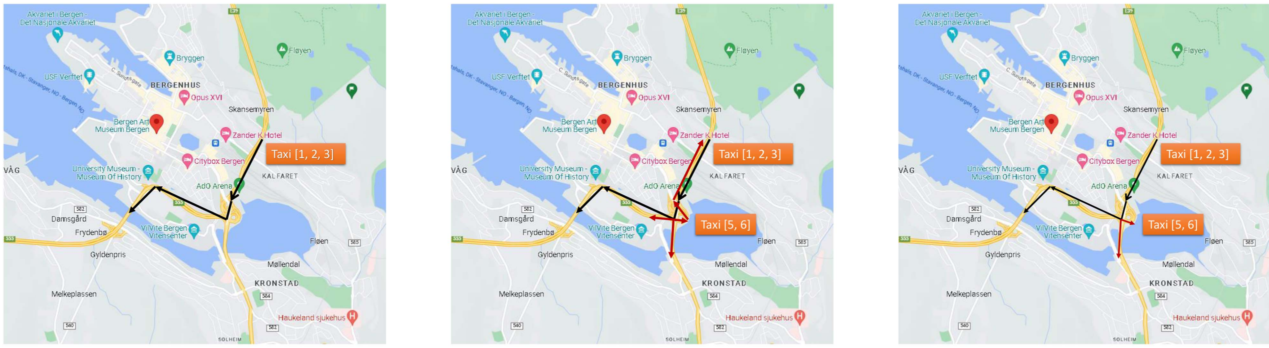
Manuscript received 29 April 2022; revised 14 July 2022; accepted 5 August 2022. Date of publication 10 August 2022; date of current version 3 February 2023. This work supported in part by the National Centre for Research and Development under the project Automated Guided Vehicles integrated with Collaborative Robots for Smart Industry Perspective under Grant NOR/POLNOR/CoBotAGV/0027/2019-00 and in part by the Western Norway University of Applied Sciences, Bergen, Norway. *(Corresponding author: Jerry Chun-Wei Lin.)*

Usman Ahmed and Jerry Chun-Wei Lin are with the Department of Computer Science, Electrical Engineering and Mathematical Sciences, Western Norway University of Applied Sciences, 5063 Bergen, Norway (e-mail: Usman.Ahmed@hvl.no; jerrylin@ieee.org).

Gautam Srivastava is with the Department of Mathematics and Computer Science, Brandon University, Brandon, MB R7A 6A9, Canada, and also with the Research Centre for Interneural Computing, China Medical University, Taichung 404327, Taiwan (e-mail: srivastavag@brandonu.ca).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TFUZZ.2022.3197876.

Digital Object Identifier 10.1109/TFUZZ.2022.3197876



(a) (b) (c)

Fig. 1. Deviation point for group base trajectory. (a) Single deviation point. (b) Group deviation point 1. (c) Group deviation point 2.

detection methods aim to help determine abnormal traffic flows using daily observations [10], [12], [15].

The fundamental goal is to remove inconsistencies from the regular flow [16], [17]. Current techniques take a solitary perspective of outliers within a sub or whole trajectory. In addition to individual outliers, a subtrajectory or whole trajectory may contain a collection of outliers. Patterns can be used and created to detect group outliers in trajectory data or to derive deviation points from trajectory data for both individuals and groups of outliers. Innovative urban traffic monitoring solutions should put less strain on mobile traffic monitoring equipment and adapt effectively to an environment where resources and space are limited. It would be more efficient to compare a smaller number of paths instead of monitoring a large number of links or performing network tomography. Network tomography enables the analysis of road segments about their adjacent road segments. It is well established that efficient road segment time estimate via network tomography is possible [18]. A collection of people outliers more closely related to one another is described as a group of trajectory outliers defined by shared characteristics, and divergence from the set of points results in abnormality.

A. Motivation

By leveraging the participants' communications and collaboration, the Cooperative Intelligent Transportation System *(C-ITS)* contributes to sustainability, increases safety, and provides comfort. Traffic management systems are critical to the sensible and contemporary movement of people and goods in urban areas. The C-core ITS's component is its traffic control system, linked to roadside devices and automobiles. Traffic management systems create a considerable volume of traffic data due to mobility. That data mainly comprises city road networks and their associated trajectory data. This massive collection of trajectory data are frequently derived from sensor networks linked with C-ITS.

The motivation for the proposed work is summarized in Fig. 1, where *Taxis 1, 2, 3, and 4* all take the same path to their destination. However, *Taxi 5* and *6*, depart from the same spot rather than sharing a common origin and destination with the other taxis. In the past, user thresholds have been used to determine the degree of similarity between trajectories and their outliers [19], [20], [21]. The similarity score computed for every trajectory determines whether or not the trajectory is an outlier. They do not attempt to compare the structure of every trajectory. In Fig. 1(b), *Taxis 5* and *6* depart from the same deviation location but have two distinct routes. However, in Fig. 1(c), the two taxis depart from the same spot and travel in the same direction. Identifying deviation points via trajectory analysis enables C-ITS to get valuable insight into the following deviation points. They are detecting an individual deviation points (IDP) by Fig. 1(a) which assist in identifying possible taxi frauds and IDP. This helps C-ITS decision makers to apply suitable controls and better detect potential fraud by strategically deploying surveillance cameras.

They are tracing the trajectory of a group. Fig. 1(b) and (c) demonstrate how the C-ITS may be able to identify critical deviation points for camera placement. The deviation from the same destination and following divergent routes has great implications for avoiding circumstances on the main route, such as traffic congestion, but cab fraud is not one of them. On the other hand, there is a possibility that a group of cabs that are statistical outliers and have separate deviation points, but still follow the same trend, work together to commit cab fraud. The network structure and temporal information, are critical for detecting trajectory outliers. The primary issue with previous techniques is that existing taxi fraud detection algorithms identify a collection of outliers in the trajectory and spatiotemporal deviation spots.

In this paper, we provide a fuzzy contrast set knowledge strategy for automatically identifying the most relevant transportation nodes for deviation point classification. This semantic knowledge defines the relationships between transportation nodes. It is concerned with identifying points of divergence in a transportation network. Additionally, it involves showing intervals for classifying deviation nodes about contrast sets. The use of fuzzy ontologies reduces the requirement for expert system implementation. After detecting the contrasts, the system allocates a node for deviation point analysis based on the individual and group trajectory.

B. Contribution

This article discusses the approach for detecting individual trajectory outliers instead of the method for detecting groups of trajectory outliers (GTOs) based on deviation points. We assessed the structural trajectory similarity between nodes using the structural embedding approach. Two nodes with a similar local structure due to their journey are deemed comparable. The structural resemblance exists regardless of the trajectory's position in the network or the neighborhood of the trajectory nodes. The approach employs a hierarchy to determine trajectory similarity, which aids in preserving deviation point analysis. The out-of-degree trajectory nodes are considered at the bottom of the hierarchy. Additionally, the complete network is utilized for a similar trajectory at the top. In the hierarchical multilayer network, weighted random walk traversal is employed by considering the two frequency contexts of the trajectory nodes. We use the context to generate and learn a latent representation of the path taken by the node. After embedding the extraction technique, we provide the contrast setting and fuzzy method for classification to perform self-supervised deviation detection in the extracted trajectory embeddings.

ITS is becoming increasingly popular. The trajectory-based knowledge graph provides ITS semantic and connectivity capabilities. This study embeds trajectory deviation points and deep clustering. We created the structural embedding by keeping the network structure and the neighbors of the nodes. This method was used to learn the latent representation from the deviation points of the road network. We constructed sequences using a hierarchical multilayer network and a biased random walk. This research provides a fuzzy contrast-based model that identifies deviation sites using position-weighted nodes. We averaged the node embeddings to obtain a travel embedding. We then extracted the embedding of the contrast set using similarity-based pairwise classification. Numerical experiments show that learning trajectory embedding captures structural identity and outperforms alternative techniques. A deep contrast set enables accurate detection of outliers and deviations in the trajectory. We have the following contributions in particular.

1) We propose a technique for describing the identifiability of a trajectory-based network connected via C-ITS based on structural similarity.
2) The contrastive fuzzy learning technique picks data points from the pool whose prediction likelihoods diverge from their neighbors in the training set.
3) The learned representation is utilized for training and classifying the input nodes for deviation point detection based on the attention-based method.

II. Literature Review

The approaches for detecting outliers in a trajectory are classified into offline and online methods. Massive data quantities and a complicated semantic structure have put major technological obstacles to outlier detection. Zhu et al. [22] introduced the approach for detecting time-independent outliers. The technique follows the same route from the source to the destination. Outliers and the typical course can be distinguished using a predefined threshold. The isolation-based approach to anomalous trajectories uses an outlier trajectory extractor characterized by a number of properties. Lv et al. [23] presented a group route approach based on central locations. The core locations serve as conventional routes. The trajectory scores are computed using the distance-based technique. Outliers are trajectories that deviate from a predefined level of similarity.

The online approaches identify considerably distinct subtrajectories. Chen et al. [24] described a technique for detecting taxi diversions using an adaptive working window. Lee et al. [20] discussed the partitioned approach based on trajectories for separating subtrajectories. This approach considers the projection and angular dimensions while computing t segments. Yu et al. [25] also suggested an outlier identification approach based on subtrajectory analysis. The approach calculated similarity scores using a neighbor set of subtrajectories. Wu et al. [26] developed a probabilistic model that made advantage of entropy-based inverse reinforcement learning. The approach converts mapped trajectories to actions associated with historical trajectories. The approach identified subtrajectories by utilizing probability threshold values. Mao et al. [27] presented an approach based on trajectory fragmentation. Every segment of a trajectory is composed of two successive points. The local trajectory assists in identifying outliers in the fragments. Yu et al. [28] developed subtrajectories using the slice-based outliers approach. Slice determines line segments based on the direction of the route. If and only if the number of neighboring trajectory slices is fewer than a specified value, the slice is called an outlier.

There are a few proposed solutions to the challenge of detecting group outliers. Outliers have been identified using statistical methods [29], [30], [31]. Chalapathy et al. [32] recommended that values for group outliers be obtained using a deep generative model. The group reference function utilizes the usual back-propagation process to estimate the input data. For learning, the Monte Carlo method [33] and Gibb's sampling [34] are then employed to obtain better performance. Das et al. [35] used Bayesian network anomaly detection and included the relationship between data outliers to identify anomalous patterns. The correlation score between individual outliers was calculated using the training data's potential outlier values. Tang et al. [36] argued that contextual outlier detection is a set of points that share similarities. These points are similar in some ways as well as dissimilar in others. The contextual outliers are determined using a statistical significance test that exceeds a predefined threshold.

Internet of Things (IoT) technology advancements leads to widespread deployment of IoT systems across a city or country. The vast data gathered and processed by IoT systems presents security issues. Intelligent network intrusion detection systems (NIDS) have been developed to avoid smart application exploitation of IoT data [37]. Existing techniques may have inadequate and uneven attack data while training the detection model, making the system vulnerable to unforeseen attacks. In the article [37], a novel hierarchical adversarial attack (HAA) generating approach is proposed to achieve a level-aware black-box adversarial attack strategy for IoT systems with a constrained budget. Building a shadow graph neural network (GNN) model, a saliency map-based approach is devised to produce adversarial instances by finding and changing essential feature components with low perturbations. A hierarchical node selection technique based on random walk with restart (RWR) is designed to choose nodes with high attack priority, considering structural aspects and overall loss changes in the targeted IoT network. The proposed HAA generating technique is examined with three baseline methodologies. Comparison results show it can reduce classification precision by more than 30% in graph convolutional networks (GCN) and jumping knowledge networks (JK-Net) for network-based intrusion detection system (NIDS) in IoT scenarios.

Hierarchical hybrid network (HHN) models are developed to represent multitype interactions among diverse entities [38]. Various measures are defined to assess internal or exterior correlations between layers inside one layer. A deep reinforcement learning based router generates optimal HHN routing actions. An intelligent router develops a restart-based random walk algorithm based on hierarchical network effects and various correlations. An intelligent recommendation process is devised and deployed in scholarly big data situations. Experiments using digital bibliography & library project (DBLP) and ResearchGate data demonstrate the model's usefulness.

Zhou et al. [39] focus on academic influence-aware multidimensional network analysis using scientific big data from multiple sources. It is proposed and specified that a set of measures are used to quantify correlations in activity-based collaborative relationships, specialty-based connections, and topic-based citation ability among academic entities (such as researchers and articles) within a constructed multidimensional network model. An improved algorithm based on RWR is also developed to provide academics with research collaborations navigation for their future ventures.

Li et al. [40] proposed assigning feature weights to every group outlier and determining the link between distinct groups using chain rule entropy. Parallel computing was used to discover contextual outliers in high- and low-dimensional domains. Xiong et al. [41] proposed a solution that identifies two types of group anomalies: 1) a group of individual abnormal points; 2) a group of normal points with an abnormal distribution. Other approaches group comparable outliers together [35], [36], [42]. Every cluster is thus referred to as an outlier group. Soleimani and Miller [42] proposed a supervised learning strategy for combining outliers when their membership is unclear. This technique has found application in document modeling, where nonuniform subjects may be identified in a collection of documents. Sun et al. [43] described an unusual group-based method for medical fraud. This abnormal group issue was recast as the maximum clique enumeration problem [44], with the patients' set seen as a collection of vertices. Each edge suggests that the two patients connected are comparable. Different partitioning approaches have been investigated to minimize the graph's size, given that maximal clique enumeration is an non-deterministic polynomial-time hardness (NP-hard) task.

Zhang et al. [45] provided two deep learning based frameworks with innovative spatiotemporal electroencephalogram (EEG) representations to detect human intentions. Both convolutional and recurrent neural networks explore spatial and temporal information in cascade or parallel. Extensive experiments on a large EEG dataset (108 participants) show that the proposed systems achieve 98.3% accuracy and outperform the state-of-the-art and baseline models. The proposed models are tested with a real brain-computer interface (BCI) and achieve a recognition accuracy of 93% over five instructional intentions, showing high generalization across intentions and BCI platforms.

Luo et al. [46] used a novel semisupervised feature selection method. The model assumes that instances with comparable labels should be neighbors. Instead of using a predefined similarity graph, the local structure finds the ideal graph simultaneously. An adaptive loss function is used to measure label fitness, which improves the robustness of the model to lossy films. They presented an effective alternating optimization technique to solve the challenging problem and analyzed its convergence and computational complexity. Extensive experimental results on benchmark datasets demonstrate the usefulness and superiority of the proposed approach.

Chen et al. proposed a semisupervised deep model for identifying imbalanced activities from wearable data [47]. It simultaneously accounts for multimodal sensor data (e.g., interpersonal variability and interclass similarity), limited labeled data, and class imbalance problems. The method provides a semisupervised framework for extracting and preserving latent activity patterns. The recurrent convolutional attention networks use multimodal sensory data to detect prominent regions suggestive of human actions. Experimental results show that the proposed model outperforms state-of-the-art semisupervised and supervised approaches using 10% labeled training data. The technique is robust even with imbalanced training datasets.

The primary literature discusses a different approach for detecting trajectory. However, no study has been undertaken to discover deviation points in a group of outliers' trajectories. Individual outliers and subtrajectory-based outliers, are detected using these approaches. All methods apply the concept of group outliers to a collection of possible groups rather than individuals, as is the case with distribution-based methods. However, the distribution does not correlate with observed occurrences in reality. The structural trajectory representation was used to locate deviation points in trajectories, followed by the fuzzy contrast set approach.

III. Preliminary and Problem Statement

We can define a trajectory as a sequence of points by geographical location (latitude, then longitude, also including time). When being used in any fashion related to trajectory outlier detection, we discuss some of the preliminary definitions as follows:

Definition 1 (Trajectory Database): We can define that a collection of raw trajectories is stored in the database as T $= \{T_1, T_2, \ldots, T_m\}$, where each and every trajectory, T_i, is a collection of time-ordered points $(p_1, \ldots, p_n)$, and each and every point representing a geographical location and acting as a node in the graph construction.

Definition 2 (Mapped Trajectory Database): We can define that a set depicts the sequences of spatiotemporal areas in the mapped trajectory database, e.g., $\Lambda = \{\Lambda_1, \ldots, \Lambda_m\}$, In this

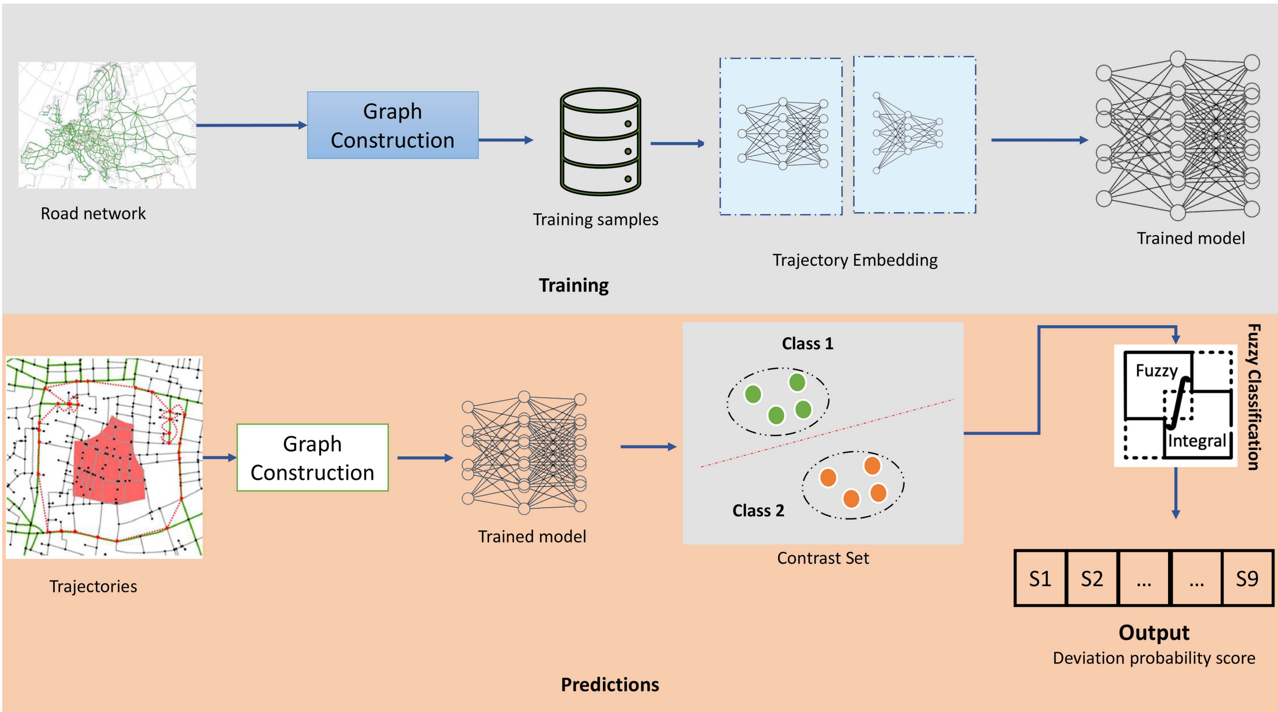


Fig. 2. Training and prediction of contrast set based deep fuzzy classification.

case where each and every mapped trajectory Λ_i represents region, e.g., $(R_1, \ldots, R_n)$, may be able to be obtained by mapping every point in T_i to its nearest region R_i.

Definition 3 (Trajectory Dissimilarity): A distance of two trajectories of the presented graph is then defined as $d(\Lambda_i, \Lambda_j)$, which can be calculated by a specific learning-based model.

The trajectory candidate is then considered as a directed graph that illustrates the potential of the trajectory containing a collection of outliers. These trajectories can be determined using dissimilarity measurements.

Definition 4 (Trajectory Candidate): We can define that a collection of reference graphs that the trajectory outlier detection method is provided, indicated by $\mathcal{G}^+ = \{\Lambda_1^+, \Lambda_2^+ \ldots \Lambda_l^+\}$.

Definition 5 (The Group Trajectory Outlier [GTO]): The set of trajectories can thus be discovered by the GTO detection model, in which it can diverge from x. Notice that x represents a starting point aligned with a trajectory, at least, in $\mathcal{G}$. Notice that x is highly ordered against the other starting points mentioned in $\mathcal{G}$.

A. *Proposed Model*

This work provided a method for embedding trajectory data into a latent representation. Trajectory embedding converts network nodes into vectors in multidimensional space. The road network represents the structure of the road, while the trajectories represent the vehicles passing through these roads at specific time intervals. We used a multihop ring structure based technique to maintain the structural representation and to obtain important information about the nodes and their edges. Group trajectory outlier clustering is then used to integrate the learned embedding vectors. Structurally obtained trajectory embedding aims to extend knowledge based on k hops for finding the deviation points. This helps the adaptive clustering approach to identify a set of trajectories based on the deviation points. Fig. 2 shows a flowchart of the developed framework. The input dataset consists of labeled trajectory data. The output represents the probability score of the instances S, where we begin by training the suggested trajectory embedding on the structure and deviation points. The function generates labeled features for paired classification models. The cosine similarity is then calculated, and every journey is labeled.

This article characterizes two data points as contrastive if and only if their model encodings are similar, yet their model predictions are considerably different (maximally disagreeing predictive likelihoods). For instance binary classification model of sample X_1(0.75, 0.25), X_2(0.65, 0.35), X_3(0.55, 0.45), and X_4(0.35, 0.65), Then, the most contrastive pair would be (X_1, X_4) as similar deviation point embedding contains different probability scoring. The data points with similar model embedding and dissimilar model outputs should be located around the model's decision border. Hence, contrastive examples are associated with gaining tough instances at the model's decision boundary. Contrast-based boundary does not ensure that the contrastive instances are located around the model's decision boundary. The second condition should require that the model identify the two examples as separate classes to guarantee that a pair of contrasting cases is placed on the boundary (i.e., different predictions). On the other hand, determining the distance between an example and the decision boundary of the model is computationally difficult, and approximations based on adversarial examples are computationally costly.

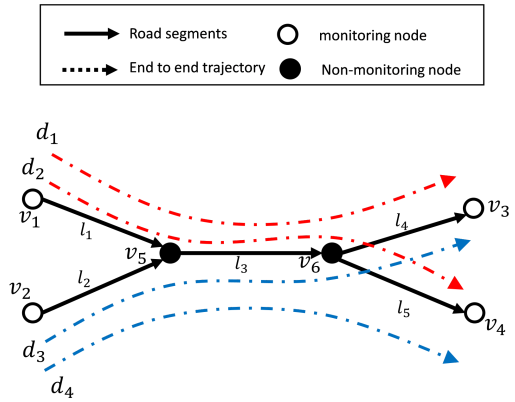


Fig. 3. Illustration of the C-ITS, which monitors the road network and its course from beginning to end. It is responsible for monitoring five road sections, although only four cameras (monitoring nodes) are installed.

B. Graph Construction

We can then define that a road network is represented by Fig. 3, which is considered as a directed acyclic graph, i.e., $G = (V, L)$. Thus, the road segments can determine the progress of the vehicle, which is then described as follows. Two variables, i.e., V and L, are then represented as the collection of road nodes and segments, respectively. From the presented Fig. 3, it follows that $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ and $L = \{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5\}$. The η is defined as the camera networks dubbed monitoring nodes and thus can be used to measure specific nodes. While the nodes generate a trajectory and the road segments, the trajectory of the deviation points is then discovered and defined in $\mathcal{D}$. Consider an example as follows, in Fig. 2, η is then set as 4, where η here is the number of monitoring nodes used to obtain the trajectories by several excursions, e.g., $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$. If a trajectory ℓ_j to nodes v and k has a deviation point defined as d_i that can be tracked and monitored, the route relation is then considered as $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$. The trip is thus modeled and considered as a sequential directed acyclic graph, which can be defined as G. It is then the location points that can represent space and time in the designed model. Note that round trip is not considered in the designed model. From Fig. 3, we can see that the edge connection between each node is represented as intermediate positions, where the trajectory is then considered as the $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$ nodes, showing an intermediate nature in the designed approach.

C. Structural Embedding of Trajectory

We employed the latent learning representation for nodes following the road network by performing the following procedures.

1) We determine the structural similarity of nodes between vertex pairs. We employed a multiedge deep neural network with a hierarchy structure to analyze structural trajectory similarity at every hierarchy level.
2) The construction of the weighted multilayer graph involves the representation of each node in each layer of the graph. The layer of the deep neural network that responds to hierarchy features indicating trajectory similarity is called the hidden layer. The edge weight of each road segment is inversely proportional to the degree to which it is structurally similar to the other road segments within each layer.
3) The sequences of road segments were created using a biased random walk on a multilayer hierarchical network. The sequences are a collection of architecturally-related road segments.
4) By providing the context indicated above, we employed the skip-gram approach to obtain a latent representation.

The ordered degree sequence of a collection of nodes $S \subset V$ was employed. Let $T_k(x)$ nodes reflect the total number of hops over k distances in G. For example, $T_1(x)$ specifies the set of vertices x's 1 neighbors. The term $T_k(x)$ refers to the trajectory development of nodes at a distance of k. By comparing the ordered degree sequences of the trajectory nodes of x and y, we establish a hierarchy to evaluate structural similarity (two nodes in the node network). The term "learning function" refers to the capability of an individual to acquire knowledge. The trajectory's structural distance between x and y is denoted by $f_k(x, y)$. We consider their immediate surroundings to be k (all road network nodes within a distance of k).

In this work, we see that k can assist in the actual computational aspects of the used degree sequences for the nodes located at an equivalent distance when measuring x to y, making use of trajectory growth for any distance. In this work, the distance between the 2 degrees of sequences ordered is calculated using dynamic time warping (DWT). This well-known approach can enable the actual extracting of any usable distance that can be, to a larger extent, tolerant of any sequences with varying lengths and in the compression of loose sequence patterns. The method DWT can itself help determine the optimal alignment of the trajectory growth sequences of x and y. Given a distance function $d(x, y)$ for each element in the sequences, the technique DWT finds matches for the sequences such that the sum of distances between elements found to be similar is reduced. We use (1) as a distance function since the evolution of the trajectories is represented by the degree sequences of the nodes that have a neighbor

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

The random walk approach seeks to stride only those architecturally similar nodes based on trajectory deviation points. This results in the context of a node u being defined by architecturally comparable nodes but without their label information or network position. The node x in V initiates the random walk in its layer 0 associated vertex. The walks are all the same length (number of steps). This method is done indefinitely, resulting in numerous independent walks (plural contexts for node x). For the representations learning, we employed the skip-gram approach. The resulting sequence is applied to the model (biased random walks in a multilayer graph). In this case, we then employ the skip-gram model that can help us find the maximal content inside of a sequence in which the context of a node is defined by the window size w that is centred on it. Additionally, we employed

TABLE I
SET OF INSTANCES WITH DISTINCT CLASSES

Instance#	Class	$Node_1$	$Node_2$	$Node_3$	$Node_4$	$Node_5$
I_1	c1	v_1	v_2	v_3	-	-
I_2	c1	-	v_2	v_3	v_4	-
I_3	c2	v_1	-	-	v_3	-
I_4	c1	-	v_2	v_3	$v4$	v_5
I_5	c2	v_1	-	v_3	v_4	-

TABLE II
CONTRAST SET WITH MINIMUM THRESHOLD OF 0.60

Contrast set	Count	Support	Class-1	Class-2	$Sup(instance, Class_1)$	$Sup(instance, Class_2)$	Support-Difference
v_1	3	0.60	1	2	0.33	0.67	0.33
v_2	3	0.60	3	0	1	0	1
v_2 , v_3	3	0.60	3	0	1	0	1
v_3 , v_4	3	0.60	2	1	0.67	0.33	0.33

TABLE III
CONTRAST SET WITH THE CONTEXTS

<table>
<tr><th>Instance</th><th>Class</th><th colspan="4">Nodes</th></tr>
<tr><td>I_1</td><td>c_1</td><td>v_1</td><td>v_2</td><td colspan="2">v_2, v_3</td></tr>
<tr><td>I_2</td><td>c_1</td><td colspan="2">v_2</td><td>v_2, v_3</td><td>v_3, v_4</td></tr>
<tr><td>I_3</td><td>c_2</td><td colspan="4">v_1</td></tr>
<tr><td>I_4</td><td>c_1</td><td colspan="2">v_1</td><td>v_2, v_3</td><td>v_3, v_4</td></tr>
<tr><td>I_5</td><td>c_2</td><td colspan="2">v_1</td><td colspan="2">v_3, v_4</td></tr>
</table>

the hierarchical softmax method, which employs binary tree classifiers to compute conditional symbol probabilities. Every node in V is allocated a unique route through the classification tree.

D. Binary Pairwise Classification Model

To classify the travel series accurately, we use the binary paired deep neural network in the training phase. For the binary classification, we used the developed contrast set as mentioned in Section III-E. This model uses a recurrent neural network and a gated unit. Since the long short-term memory (LSTM) model can be used to preserve the memory, which is very suitable to preserve the ling distance data and use it in the ordered sequential tasks, we apply this mechanism to measure the average element by element. The following function may be able to compute the learning function F associated with the given relation R.

The variable R denotes the deviation points based trajectory. If and only if the journey follows a typical trajectory, they are classified as usual; otherwise, they are aberrant. Every journey in this work is comprised of distinct nodes. We obtain the embedding for the preceding section's discussion. We then utilized embedding averaging to represent a group of nodes as a single trip for every trip. Labeling features collected from trajectory averaging were used to create binary pairwise classification models. Based on cosine similarity, we select the most similar trajectories that are more structurally connected. We minimized the search space using the degree-based optimization technique, which allowed us to run a larger network [48].

E. Contrast Set

Example: Consider the dataset mentioned in Table I, this contains five instances and two $Classes$, i.e., c_1 and c_2. In this example, suppose that an instance is defined as I_w, $\{v_1, v_3\}$, which shows in three different instances, e.g., $I_1,\ I_2,\ and\ I_4$.

Algorithm 1: Contrast Set Fuzzy Classification.

Input: Transport network (G, V, E) and Mapping M.
Output: Contrast set with fuzzy classification rules.

1: $Tajactories \leftarrow Mapping$(G , V,E).
2: $Vectors \leftarrow Embedding(Tajactories)$).
3: $Sets \leftarrow Contrast_{set}(Vectos)$.
4: **Fuzzification:** Convert the values of the support difference to fuzzy inputs.
5: **Fuzzy Rules generation:** Create rules based on fuzzy membership functions.
6: **Defuzzification:** Convert fuzzy rules to crisp rules.
7: Improve fuzzy inferences by utilizing the contrast set approach.
8: Validate the model with validation data.
9: Conduct a statistical test to validate the fuzzy inference rules for the contrast set.
10: **Return:** Inference rules.

The support value for $\{v_2, v_3\}$ of the database $\mathcal{L}$ can thus be measured by $sup(\{v_2, v_3\}, L) = 3/5 = 0.6$. In addition, for the set $\{v_2, v_3\}$ regarding the distinct class c_1, the support value is then measured by $sup(\{v_2, v_3\}, c_1) = 3/3 = 1$. The reason is that we do have three instances in this example, i.e., instance $\{I_1, I_2, I_4\}$) with a class c_1 contains $\{v_2, v_3\}$. In the same way, the support of $\{v_2, v_3\}$ with respect to the second class c_2 is measured as $sup(\{v_2, v_3\}, c_2) = 3/0 = 0$. The detailed calculation is mentioned in the Table II. The final contrast sets are mentioned in the Table III. Notice that if and only if minimum support and difference is set to $\delta = 0.6$, then $\{v_2, v_3\}$ is said to be a contrast set as $sup(\{v_2, v_3\}, L) = 0.6 \geq \sigma$ and $\text{Diff}(\{v_2, v_3\}, L) = 1 \geq \delta$.

From the results showing in Table III, a contrast set is then considered as $\{v_2\}$, we have the similarity context $\mathcal{N}_S(b) = \{v_2 v_3\}$ as both words share the deviation point v_2. The cooccurrence metric for first instance is $(v_1,\ v_2, (v_2, v_3))$ since v_2 cooccurs with v_1 in instance $I_1,\ v_2$ cooccurs with $\{v_2,\ v_3\}$ in instance $I_{1,2,4}$ and v_2 cooccurs with $\{v_3,\ v_4\}$ in instance $I_{2,4}$. The similarity and cooccurrence help to measure the contrast sets as accurate based on the cooccurrence and their individual instance sets. We describe how the cooccurrence context helps capture the contrast sets with the above example. We now discuss the usefulness of the fuzzy sets.

A fuzzy inference system must contain input and output variables in addition to a collection of fuzzy rules to be of the Mamdani type. This is the only case in which this condition must be satisfied. Both the input and output variables are represented as a collection of fuzzy sets. Input and output variables are treated somewhat differently by fuzzy rules, although they are quite comparable. Input variables use the system's input values to fuzzify their sets during the execution process. In this way, input variables can determine the degree to which an input value belongs to all fuzzy sets associated with a variable. Every rule somehow contributes to the output variables; the sum of these contributions determines the system's output. Thus, fuzzy rules possess the form's structure.

F. Fuzzy Inference System

To represent the relationship between input and output, fuzzy knowledge is used in conjunction with *if-then rules*. The approach involves developing fuzzy rules, fuzzification, generating rules for inference, and defuzzification to obtain crisp outputs. The membership values transform data from defined membership functions into membership degrees between 0 and 1. Using the triangular membership functions, we modified the difference values of the contrast set for fuzzification. Equation (2) is then designed to illustrate the procedure

$$f(x) = \begin{cases} 0 & \text{if } x <= i \\ \frac{x-i}{j-i} & \text{if } i \leq x \leq j \\ \frac{k-x}{k-j} & \text{if } j \leq x \leq k \\ 0 & \text{if } x \geq k. \end{cases} \tag{2}$$

1) Fuzzy Rule Generation: The contrast set rules and fuzzification values are critical since they aid in classifying the attention positionally weighted items. After contrast set creation, the itemset may be categorized into linguistic rules. The deviation points and class assignments, assist in creating several rules.

2) Defuzzification: The fuzzification processes are given the testing data. The fuzzified input matches the inference rules using the membership function values. The inference rules are derived from the linguistic values and then translated to a fuzzy score using the weighted technique. A classification decision is generated from the fuzzy score.

The flow of the operation is described in Algorithm 1. In doing so, we first map the graph-based model according to Section III-B, III-C, III-D, and III-E (Algorithm 1 - line 1). Using the mapping and the trajectories, the vectors (Algorithm 1 - line 2) are created. The contrast set is created based on the vectors (Algorithm 1 - line 3). Then, the contrasts are used from the fuzzy model discussed in Section III-F (Algorithm 1 - lines 4–7). The fuzzy results are used and compared with the validation set (Algorithm 1 - lines 8–9). The inference of the model is used to generate rules as output (Algorithm 1 - line 10). In addition, inference is used to classify instances for deviation point analysis.

IV. Experimental Results

In four phases, rigorous tests were performed to evaluate the developed algorithms on various trajectory databases. Every experiment was conducted for 5 min. The serial implementation was carried out on a 64-b machine equipped with a core Core i7 processor, running on Windows 10 and 16 GB of RAM. We used simulated GTOs in our tests since the existing trajectory datasets are inapplicable to real-world settings. Our simulated dataset has the following properties: Injecting outliers from individual trajectory: Frequent additions of noise with a probability of a $p \sim \mathcal{U}(0.8, 1.0)$ and a specified threshold for creating individual trajectory outliers were made. Injecting GTOs: Noise was introduced numerous times to the individual trajectory outliers using a probability and a threshold of $p \sim \mathcal{U}(0.0, 1.0)$. The initial noise points associated with trajectory outliers are labeled as deviation points.

For comparison, we utilized a random forest comparison [49]. The random forest algorithm separated every tree into random features using the ensemble of the decision tree model. The decision tree structure is a tree-based approach, with roots, nodes, and leaf nodes. Internal nodes are given decision-making capability. Classification is carried out based on the initial nodes. The following experimental employs the F-measure and receiver operating characteristic - area under curve (ROC-AUC) measures, often used to evaluate outlier identification algorithms.

The prediction models are trained offline in the proposed model. The effort of using a predictor includes feature extraction and prediction generation. Since a feature is extracted at compile time, the feature extraction effort is insignificant. Training the prediction model is performed only once and incurs only a one-time cost. Overall, the prediction model has a small overhead of 3 s. We did not consider the overhead cost of the proposed method and the others in our experiment and comparison.

A. Data Description

Three datasets were analyzed: Intelligent transportation, climate change, and environment. The intelligent transportation dataset contains the real-world trajectories of 442 cabs in Porto, Portugal, from 01/07/2013 to 30/06/2014, and the database from the european conference on machine learning and principles, and practice of knowledge discovery in databases (ECML-PKDD) 2015 competition[1] assessments made use of contains further information on this trajectory database [50]. The Atlantic hurricane track is a climate change dataset that contains the latitude, longitude, maximum sustained surface wind, and lowest sea-level pressure of hurricane trajectories in the USA at six-hourly intervals from 1851 to 2018 [50]. This data collection has 52 775 trajectories. The environment data were derived from Starkey Projects, which contained animal movement data. The dataset was visualized using radio-telemetry positions of elk, deer, and cattle recorded between 1989 and 1999. At 30-min intervals, the positions have been kept. It contains 100 distinct trajectories as well as almost 40 000 distinct spots. This is a sparsely populated dataset.

B. Results

Numerous studies were conducted utilizing the F-measure as well as ROC-AUC performance. The results indicate that deep clustering surpassed classification for trajectory datasets (intelligent transportation, climate change, or the environment) even though both models use the same embedding. The deep fuzzy based model is capable of producing optimal outcomes. We use the grid search approach to learn a latent representation of the trajectory network using the suggested method, struc2vec [48] and node2vec [51]. The latent representation for every node is then used as a feature in the proposed clustering approach's paired classification procedure. Simultaneously, we utilized logistic regression for training the stuc2vec and node2vec algorithms. The conventional classifier classified the trajectories using the training embedding.

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

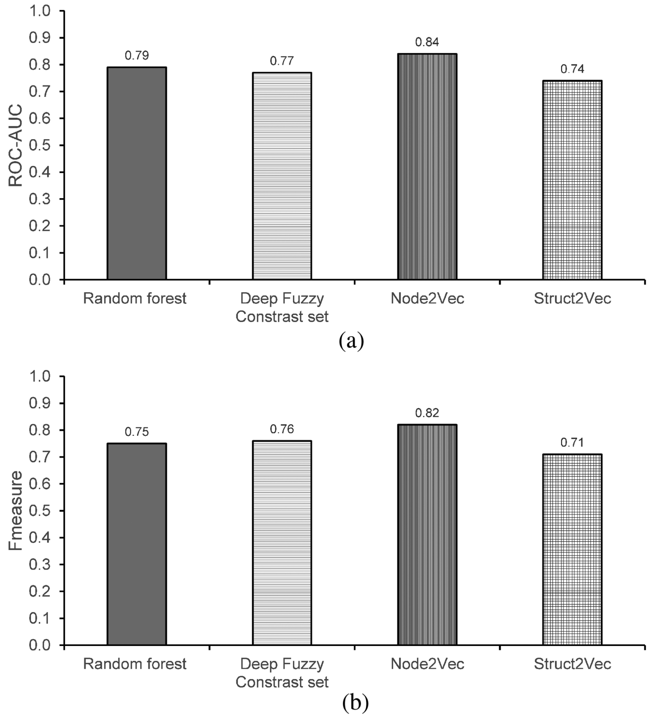


Fig. 4. Evaluation of the intelligent dataset with deep fuzzy contrast set method. (a) ROC-AUC. (b) F-measure.

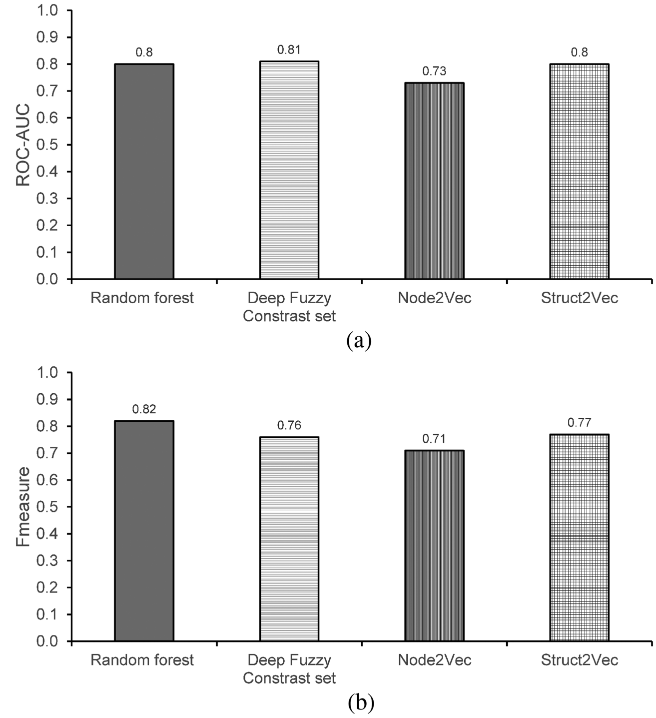


Fig. 5. Evaluation of the climate dataset with deep fuzzy contrast set method. (a) ROC-AUC. (b) F-measure.

The F-measure and ROC-AUC of traditional classifiers are depicted in Fig. 4. The results reveal that the given dynamic approach performs better than the other heuristics when using the F-measure criterion, indicating that the designed model is particularly relevant to trajectory data. Traditional classification outperformed embedding model. The results show that solutions performed well using the clustering approach model, which required more time for deep net training.

Similarly, as seen in Fig. 5, climate data may be used to categorize trajectory. The suggested model achieved a high level of accuracy. The climate dataset has a more significant number of deviation points, which results in a higher error rate for trajectories. Deep clustering earned the greatest ROC-AUC of 0.81 on the climate dataset. This demonstrates how having more deviation points may aid the learning algorithm's performance owing to the embedding's structural similarity. The stur2vec and node2vec fared better, achieving 0.73 and 0.80 F-measure, respectively.

In Fig. 6, we investigated the developed technique using the environment dataset. The developed approach shows 0.82 accuracy, whereas the stuc2vec method achieves 0.81. This is especially true for the developed deep contrast set learning model. By utilizing train embedding, the developed strategy could properly group the class. Additionally, the knowledge graph may be leveraged to expand the number of training examples.

The node2vec and sturc2vec fail to group similar latent space trips (mirrored nodes). The designed technique may be able to learn the attributes that allow for accurate identification of deviation locations and node identities. In the latent space, mirror pairs or nodes representing the same trips remain close together, and averaging the trip nodes represents the complicated structural hierarchy inherent in group representation.

Typical methods for detecting outliers in trajectory data focus on identifying individual outliers; however, the algorithms discussed in this paper focus on identifying group outliers. We were able to improve the detection of multiple outliers in trajectories by using techniques from a variety of domains. Stack-based learning, evolutionary optimization, feature analysis, and node-based clustering are some of the techniques used. The GTO solutions discussed in this paper can be used in a variety of contexts, such as developing novel city apps and performing urban analysis. Structural embedding of trajectory deviation points has been used for GTO detection in this work. Nevertheless, there is still much potential for development and additional research in this area. When it comes to using data mining and machine learning algorithms in application domains, both methodological improvements and adaptations are needed. We recommend implementing advanced methods such as merging traditional outlier identification methods for GTO. GTOs have potential applications in a variety of sectors, including the study of climate change. GTOs allow identification of a subset of hurricanes that exhibit deviations from the norm, which can then be the subject of research. By identifying areas that are more likely to be affected by storms, this information can be used to save lives and identify areas that are more likely to be affected.

When considering the F-measure criteria, the results show that the proposed dynamic approach performs better than the other heuristics. This shows that the designed model is quite relevant

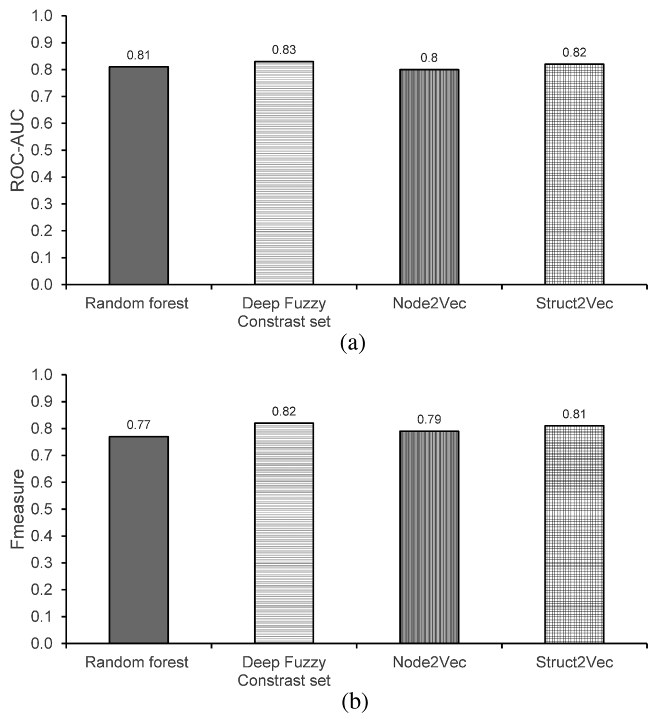


Fig. 6. Evaluation of the environmental dataset with deep fuzzy contrast set method. (a) ROC-AUC. (b) F-measure.

for the trajectory data. On the other hand, the solutions developed using the conventional classification method performed significantly better than those developed using the embedding method. Moreover, the examples show that satisfactory results were obtained by using the model with clustering approach, which required more training time since the training of deep networks was required. A higher error rate for the trajectories is produced, since the climate dataset has a significantly larger number of deviation points. The ROC-AUC, obtained by deep clustering on the climate dataset, was the highest at 0.81. This illustrates how increasing the number of deviation points in the embedding can potentially improve the effectiveness of the learning process due to the structural similarity between the two datasets. Both stur2vec and node2vec performed significantly better with an F-measure of 0.73 and 0.80, respectively. We examined the newly proposed method using the environmental dataset. In comparison, the stuc2vec method only achieves an accuracy of 0.81, while the proposed method has an accuracy of 0.82. This is especially true for systems that learn from deep contrast sets. The developed technique was able to correctly categorize students within the class because it used train embedding. In addition, the knowledge graph has the potential to be used to increase the total number of training examples.

V. Conclusion

ITS is gaining popularity in recent decades. Trajectory-based knowledge graphs give ITS semantic and connection capabilities. This study embeds trajectory deviation points and deep clustering. We created the structural embedding by keeping the nodes' network structure and neighbors. This method was used to learn the latent representation from road network deviation points. We constructed sequences using a hierarchical multilayer network and a biased random walk. This paper provides a fuzzy contrast based model that identifies deviation sites using position-weighted nodes. We averaged node embeddings to get travel embedding. Next, we extracted the contrast set's embedding using similarity-based pairwise classification. Numerical experiments demonstrate that learning trajectory embedding captures the structural identity and outperforms alternative techniques. A deep contrast set allows accurate detection of trajectory outliers and deviations.

VI. Future Work

When considering such systems, no matter what the domain, the issues of security and privacy are often weighed in favor of ease of use and saving computing power; therefore, we believe it is a logical next step to ensure that the positive steps are achieved in this work have not come at the expense of data security and privacy. In any data collection for outlier detection, it is essential for the security of the overall system that the data be transmitted and stored securely. In addition, for such systems, it would be interesting to see if some of the tasks could be offloaded to the collection devices.

References

[1] H. Song, R. Srinivasan, T. Sookoor, and S. Jeschke, *Smart Cities: Foundations, Principles, and Applications*. New York, NY, USA: Wiley, 2017.

[2] A. Choudhury, U. Bhattacharya, and R. Chaki, "Dynamic scheduling of traffic signal (DSTS) management in urban area network," in *Proc. IFIP Int. Conf. Comput. Inf. Syst. Ind. Manage.*, 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. 10015–10027, 2019.

[4] M. Ghahramani, M. Zhou, and C. T. Hon, "Mobile phone data analysis: A spatial exploration toward hotspot detection," *IEEE Trans. Automat. Sci. Eng.*, vol. 16, no. 1, pp. 351–362, Jan. 2018.

[5] T. Kieu, B. Yang, and C. S. Jensen, "Outlier detection for multidimensional time series using deep neural networks," in *Proc. IEEE Int. Conf. Mobile Data Manage.*, 2018, pp. 125–134.

[6] H. Senaratne et al., "Urban mobility analysis with mobile network data: A visual analytics approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 5, pp. 1537–1546, May 2018.

[7] G. Ajaeiya, I. H. Elhajj, A. Chehab, A. Kayssi, and M. Kneppers, "Mobile apps identification based on network flows," *Knowl. Inf. Syst.*, 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 Trans. Visual. Comput. Graph.*, vol. 24, no. 1, pp. 23–33, Jan. 2018.

[9] W. Li, H. Song, and F. Zeng, "Policy-based secure and trustworthy sensing for Internet of Things in smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 716–723, Apr. 2018.

[10] R. Conforti, M. La Rosa, and A. H. ter Hofstede, "Filtering out infrequent behavior from business process event logs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 300–314, Feb. 2017.

[11] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[12] M. Salehi, C. Leckie, J. C. Bezdek, T. Vaithianathan, and X. Zhang, "Fast memory efficient local outlier detection in data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3246–3260, Dec. 2016.

[13] I. Kalamaras et al., "An interactive visual analytics platform for smart intelligent transportation systems management," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 487–496, Feb. 2018.

[14] E. Karbab, D. Djenouri, S. Boulkaboul, and A. Bagula, "Car park management with networked wireless sensors and active RFID," in *Proc. IEEE Int. Conf. Electro/Inf. Technol.*, 2015, pp. 373–378.

[15] A. Sharifi, "A typology of smart city assessment tools and indicator sets," *Sustain. Cities Soc.*, vol. 53, 2020, Art. no. 101936.
[16] J. Mao, P. Sun, C. Jin, and A. Zhou, "Outlier detection over distributed trajectory streams," in *Proc. SIAM Int. Conf. Data Mining*, 2018, pp. 64–72.
[17] A. H. Milaghardan, R. A. Abbaspour, and C. Claramunt, "A Dempster-Shafer based approach to the detection of trajectory stop points," *Comput. Environ. Urban Syst.*, vol. 70, pp. 189–196, 2018.
[18] 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 Trans. Intell. Transp. Syst.*, vol. 22, no. 4, pp. 2198–2211, Apr. 2021.
[19] Y. Ge, H. Xiong, Z.-H. Zhou, H. Ozdemir, J. Yu, and K. C. Lee, "Top-eye: Top-k evolving trajectory outlier detection," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1733–1736.
[20] J.-G. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proc. IEEE Int. Conf. Data Eng.*, 2008, pp. 140–149.
[21] Z. Liu, D. Pi, and J. Jiang, "Density-based trajectory outlier detection algorithm," *J. Syst. Eng. Electron.*, vol. 24, no. 2, pp. 335–340, 2013.
[22] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Effective and efficient trajectory outlier detection based on time-dependent popular route," *World Wide Web*, vol. 20, no. 1, pp. 111–134, 2016.
[23] Z. Lv, J. Xu, P. Zhao, G. Liu, L. Zhao, and X. Zhou, "Outlier trajectory detection: A trajectory analytics based approach," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2017, pp. 231–246.
[24] C. Chen et al., "iBOAT: Isolation-based online anomalous trajectory detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 806–818, Jun. 2013.
[25] Y. Yu, L. Cao, E. A. Rundensteiner, and Q. Wang, "Outlier detection over massive-scale trajectory streams," *ACM Trans. Database Syst.*, vol. 42, no. 2, pp. 1–33, 2017.
[26] H. Wu, W. Sun, and B. Zheng, "A fast trajectory outlier detection approach via driving behavior modeling," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 837–846.
[27] J. Mao, T. Wang, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2696–2709, Dec. 2017.
[28] Q. Yu, Y. Luo, C. Chen, and X. Wang, "Trajectory outlier detection approach based on common slices sub-sequence," *Appl. Intell.*, vol. 48, no. 9, pp. 2661–2680, 2018.
[29] S. Li, M. Shao, and Y. Fu, "Multi-view low-rank analysis with applications to outlier detection," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 3, pp. 1–22, 2018.
[30] E. Toth and S. Chawla, "Group deviation detection methods: A survey," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–38, 2018.
[31] R. Yu, X. He, and Y. Liu, "Glad: Group anomaly detection in social media analysis," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 2, pp. 1–22, 2015.
[32] R. Chalapathy, E. Toth, and S. Chawla, "Group anomaly detection using deep generative models," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2018, pp. 173–189.
[33] G. Celeux, D. Chauveau, and J. Diebolt, "Stochastic versions of the em algorithm: An experimental study in the mixture case," *J. Stat. Comput. Simul.*, vol. 55, no. 4, pp. 287–314, 1996.
[34] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
[35] K. Das, J. Schneider, and D. B. Neill, "Anomaly pattern detection in categorical datasets," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 169–176.
[36] G. Tang, J. Pei, J. Bailey, and G. Dong, "Mining multidimensional contextual outliers from categorical relational data," *Intell. Data Anal.*, vol. 19, no. 5, pp. 1171–1192, 2015.
[37] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, "Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9310–9319, Jun. 15, 2022, doi: 10.1109/JIOT.2021.3130434.
[38] X. Zhou, W. Liang, K. I.-K. Wang, and L. T. Yang, "Deep correlation mining based on hierarchical hybrid networks for heterogeneous Big Data recommendations," *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 1, pp. 171–178, Feb. 2021.
[39] X. Zhou, W. Liang, K. I.-K. Wang, R. Huang, and Q. Jin, "Academic influence aware and multidimensional network analysis for research collaboration navigation based on scholarly Big Data," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 1, pp. 246–257, Jan.–Mar. 2021.
[40] J. Li, J. Zhang, N. Pang, and X. Qin, "Weighted outlier detection of high-dimensional categorical data using feature grouping," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 11, pp. 4295–4308, Nov. 2020.
[41] L. Xiong, B. Póczos, J. Schneider, A. Connolly, and J. VanderPlas, "Hierarchical probabilistic models for group anomaly detection," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2011, pp. 789–797.
[42] H. Soleimani and D. J. Miller, "ATD: Anomalous topic discovery in high dimensional discrete data," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2267–2280, Sep. 2016.
[43] C. Sun, Z. Yan, Q. Li, Y. Zheng, X. Lu, and L. Cui, "Abnormal group-based joint medical fraud detection," *IEEE Access*, vol. 7, pp. 13589–13596, 2018.
[44] N. Modani and K. Dey, "Large maximal cliques enumeration in sparse graphs," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 1377–1378.
[45] D. Zhang, L. Yao, K. Chen, S. Wang, X. Chang, and Y. Liu, "Making sense of spatio-temporal preserving representations for EEG-based human intention recognition," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3033–3044, Jul. 2020.
[46] M. Luo, X. Chang, L. Nie, Y. Yang, A. G. Hauptmann, and Q. Zheng, "An adaptive semisupervised feature analysis for video semantic recognition," *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 648–660, Feb. 2018.
[47] K. Chen, L. Yao, D. Zhang, X. Wang, X. Chang, and F. Nie, "A semisupervised recurrent convolutional attention model for human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1747–1756, May 2020.
[48] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 385–394.
[49] 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 Comput.*, vol. 25, no. 1, pp. 407–420, 2021.
[50] Y. Djenouri, D. Djenouri, and J. C.-W. Lin, "Trajectory outlier detection," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, pp. 1–28, 2021.
[51] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.