

Elsevier required licence: © <2021>. This manuscript version is made available under the CC-BY-NC-ND 4.0 license <http://creativecommons.org/licenses/by-nc-nd/4.0/>
The definitive publisher version is available online at <https://doi.org/10.1016/j.knosys.2021.106833>

Offline and Online Detection of Anomalous Patterns from Bus Trajectories for Traffic Insight Analysis

Xiaocai Zhang^a, Yi Zheng^a, Zhixun Zhao^a, Yuansheng Liu^a, Michael Blumenstein^b, Jinyan Li^{a,*}

^aAdvanced Analytics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia

^bCentre for Artificial Intelligence, School of Computer Science, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia

Abstract

Existing methods for traffic anomaly detection are modelled on taxi trajectory datasets. The concern is that the data may contain much inaccuracy about the actual traffic situations, because taxi drivers often choose optimal routes to evade from the congestions caused by traffic anomalies. We use bus trajectory data in this work. Bus trajectories can capture real traffic conditions in the road networks without drivers' preference, which are more objective and appropriate for accurately detecting city-wide anomalous patterns for a broad range of insight analyses on traffics. We proposed a deep learning-based feature visualization method to map 3-dimensional features into a red-green-blue (RGB) color space. A color trajectory (CT) is then derived by encoding a trajectory with the RGB colors. With the spatial and temporal properties extracted from the CT, spatio-temporal outliers are detected by a novel offline detection method. We then conduct GIS map fusion to obtain insights for better understanding the traffic anomaly locations, and more importantly the influences on the road affected by the corresponding anomalies. Extended from the offline detection, an online detection method is developed for real-time detection of anomalous patterns. Our proposed methods were tested on 3 real-world bus trajectory datasets to demonstrate the performance of high accuracies, high detection rates and relatively low false alarm rates.

Keywords: Traffic, detection of anomalous patterns, bus trajectory, deep learning, spatio-temporal outliers.

1. Introduction

Detection of anomalous traffic patterns is to figure out those traffic patterns which are not expected but which are very helpful for analysis of traffic accidents, fault detection, flow management, and new infrastructure planning [1]. In fact, anomalous patterns in moving transportation carriers' trajectories can reflect abnormal traffic streams on the transportation networks [2]. These patterns are emerged due to various factors including traffic accidents, traffic controls, parades, sports events, celebrations, disasters or other events. These abnormal patterns can also be propagated along the whole road networks, and they will not disappear automatically

without proper traffic control strategies. Therefore, it is significant to develop a method to automatically figure out these abnormal patterns by data-driven techniques [3, 4]. To our knowledge, existing trajectory-based traffic anomalies/outliers detection is mainly based on city-wide taxi trajectory data [2, 5–15]—many accessible trajectory data sources of *bus* have not been explored for traffic anomalies probing. Bus service operates along almost all the main roads in metropolitan cities everyday, which facilitates commuters substantially. Take the city of Beijing as example, at the end of 2019, 23,010 buses have been on the roads everyday, serving 3.134 billion people with 1,162 regular bus routes during the year¹. Moreover, the global positioning system (GPS) has equipped buses with high-resolution positioning information, which factually underlines the city-wide traffic situations.

Comparing with the taxi trajectory data, models

*Corresponding author.

Email addresses: xiaocai.zhang@student.uts.edu.au (X. Zhang), yi.zheng-8@student.uts.edu.au (Y. Zheng), zhixun.zhao@student.uts.edu.au (Z. Zhao), yuansheng.liu@uts.edu.au (Y. Liu), michael.blumenstein@uts.edu.au (M. Blumenstein), jinyan.li@uts.edu.au (J. Li)

¹http://www.bjbus.com/home/fun_static_page.php?uSec=00000156&uSub=00000157

based on bus trajectory data can take the following advantages: (i) as a public transportation carrier, there is not much risk of privacy leakage regarding bus trajectory data; (ii) easy to get access to the real-time bus data for many cities via API; and (iii) each bus service has its own regular route, and bus trajectory is more independent of the drivers' preference, reflecting more objectively on the real road traffic conditions. This is contrast to taxi trajectory data which may lose much accuracy about traffic congestion situations, since taxi drivers can choose paths for themselves [3, 16]. Especially when a taxi driver gets the traffic information ahead, the driver very likely chooses an optimal route to avoid a foreseeable traffic congestion.

The work in this paper utilizes deep learning architecture for feature extraction from bus trajectory data sources and develops visualization for both offline and online discoveries of anomalous traffic patterns. We also develop methods for detecting the anomaly locations to provide insights of the corresponding anomalies for understanding the influences caused by the anomaly to the road traffic. Our **contributions** of this research include:

1. We present a deep neural network architecture to extract deeply hidden features for generating better features visualization than typical dimensionality reduction methods, and conduct GIS fusion for getting insights into the anomalies, for example, the anomaly locations and their impacts caused to the road traffic.
2. We devise a novel method for an offline detection of anomalous traffic patterns at bus route level. Particularly, unlike introducing machine learning models, we design algorithm on imbalanced data by addressing the discrepancy between different classes of anomaly.
3. Extended from the feature extraction architecture and the offline detection method, we propose an online method for real-time detection of anomalous traffic patterns.
4. We perform comprehensive experiments on 3 real-world datasets to confirm the effectiveness and superiority of the deep feature extraction architecture, the offline and online anomaly detection methods and insight analysis of the anomalous patterns.

The rest of this paper is organized as follows. Section 2 reviews the related work on the studies of anomalous patterns detection. In Section 3, we elaborate on the

methods of feature extraction and visualization, offline and online algorithms for anomalous patterns detection and insight analysis on anomalies. Section 4 presents experimental results and analyses. Section 5 concludes this paper and presents future work.

2. Related work

Anomalous pattern detection aims to detect unexpected patterns, which has been intensively studied in the domain of data mining and knowledge discovery [2]. To our best knowledge, at least four categories of methods were proposed, including dimensionality reduction-based methods [17–21], unsupervised methods [22–28], supervised classification-based methods [14, 29–32] and statistical methods [33–35].

Dimensionality reduction method like principal component analysis (PCA) has been validated effective in anomaly detection [17, 18]. In [19], an improved PCA by introducing Kullback-Leibler divergence was proposed for network anomaly detection. Random projection (RP) was used for dimensionality reduction and detect internet traffic anomaly [20, 21]. Apart from linear methods, a nonlinear dimensionality reduction method (autoencoders) has been presented in [36].

Unsupervised clustering algorithms including k-means [22], density-based and grid-based clustering [23] and one pass clustering [24] were proposed to identify anomalous network patterns. Besides clustering algorithms, one-class support vector machine (OneSVM) with novel kernels was introduced to detect malicious intrusion to computer systems [25, 26]. Recently, unsupervised deep learning-based methods have also been presented for modelling large scale data, and to detect anomalies [27, 28].

Literature [29] sorted the average k-nearest neighbour (kNN) distances in ascending orders, and then outliers were defined when the difference between two nearby distances is greater than a preset threshold [29]. Besides the lazy learning approach, recently, supervised deep learning-based anomaly detection methods have also contributed to solve this problem, including the recurrent neural network (RNN)-based model [14] and long short-term memory (LSTM)-based model [30–32].

There are also some studies introducing statistical methods for outlier detection. Baraba et al. [33] proposed to use transductive confidence machines and hypothesis testing to uncover outliers. Fan et al. [34] presented a continual aggregate statistics method. In [35], a multi-modal distance measure was defined to evaluate the strangeness. Furthermore, statistical testing was applied to estimate the probability of anomaly.

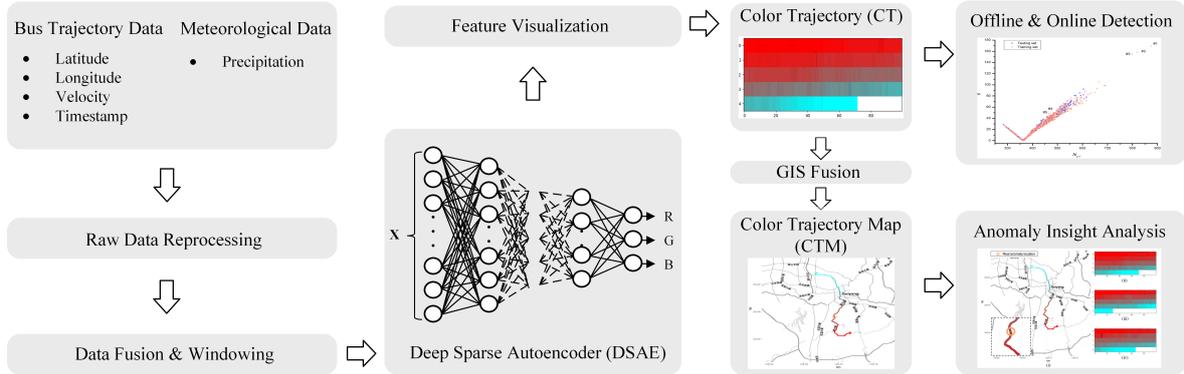


Fig. 1. The workflow of offline and online detection of anomalous traffic patterns and anomaly insight analysis.

Since the anomalous patterns in road traffic possess their own characteristics, some particular methods were presented on the top of aforementioned general anomaly detection methods. Based on the data source utilized, they could be dichotomized into two families: by using trajectory data sources or by using other data sources.

Apart from trajectory data sources, other data sources used for road traffic anomaly detection are mainly non-structured, including social media data [37], video surveillance data [1, 38–40] or heterogeneous traffic data [41]. Literature [37] used text data from Twitter for real-time traffic incident detection. Video data collected from traffic surveillance cameras is employed by [1, 38–40] to detect or classify traffic anomalies. In addition to using single data source, study [41] explored the heterogeneous data sources collected from various vehicle sensors for traffic anomaly detection.

Trajectory-based road traffic anomaly detection has been intensively investigated by many studies, while most of which are based on city-wide taxi trajectories. Studies by [5, 12] used PCA or wavelet transform technique to identify traffic anomalies from taxi trajectory data. In [6] and [8], likelihood ratio test was introduced to represent traffic patterns for fast detection of anomalous patterns. Liu et al. [2] constructed an anomaly detection model by building a region graph, where a node represented a region and the link between every two nodes denoted the traffic flow, and then the extreme outliers could be detected from the graph links. In [9], tensor decomposition technique was employed for learning dynamic context features from taxi traces data, and then anomalous degrees for road segments were calculated. Authors in [13] proposed neighbor-based trajectory outlier definitions, and designed an optimized strategy to detect new outlier classes from massive-scale trajectory streams. In [10], a feature grouping-based anomaly

detection framework was proposed to identify outliers from taxi trajectories. Study work by Wang et al. [7] detected traffic jam events by estimating traffic flow speed on the road. Research [11] demonstrated a method to group taxi trajectories crossing the same source destination cell-pair, then isolation mechanism was employed to detect abnormal trajectory. Wu et al. [15] developed a novel trajectory outlier detection approach by modeling the driving behavior from historical taxi trajectories.

3. Methodology

This section introduces the preliminary definitions, and presents our method for feature extraction and trajectory visualization through deep learning. Then, details of our proposed offline and online methods are described to detect anomalous trajectory and to obtain insights into the anomaly based on the visualized trajectories. A basic workflow is illustrated in Fig. 1. An important step of the method is to feed the bus trajectory data and meteorological data into a well trained deep sparse autoencoder (DSAE) to generate the color trajectory (CT), which provides the basis for trajectory visualization, offline and online detection of anomalies. Another key sector is to produce a color trajectory map (CTM) by GIS fusion for anomaly insight analysis.

3.1. Preliminaries

Definition 1. *Trajectory:* A trajectory \mathbf{T} of a moving objective is a set of time-ordered data points, $\mathbf{T} = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{N-1}, \mathbf{t}_N) \in \mathbb{R}^{D \times N}$, $\mathbf{t}_i = (\varphi_i, \lambda_i, v_i)^T \in \mathbb{R}^3$, where each data point consists of latitude φ_i , longitude λ_i and velocity v_i at the i th timestamp.

Definition 2. *Class A Anomaly:* An anomalous trajectory whose associated spatial and temporal feature values are both very different from the spatial and temporal feature values of its spatio-temporal neighbors.

Definition 3. *Class B Anomaly:* An anomalous trajectory whose associated spatial feature value is very different from the spatial feature values of its temporal neighbors.

3.2. Feature extraction and trajectory visualization using deep learning

The method employs a nonlinear dimensionality reduction method (DSAE) to extract hidden features from bus trajectory data to characterize the trajectories for trajectory visualization.

As mentioned in Definition 1, a trajectory is a time series of data points with the same time interval, each data point is typically consisted of latitude φ , longitude λ and velocity v (unit: km/h). The speed information is popularly available in many existing GPS devices. However, it can also be approximated by algorithm in literature [42] in some cases of speed data lack.

Rainfalls, especially heavy rains, can significantly affect traffic flow characteristics and may lead to traffic congestions or even accidents [43]. We integrate the bus trajectory data with local precipitation data r (unit: mm/h). Thus, \mathbf{t}_i is updated as \mathbf{z}_i denoted by

$$\mathbf{z}_i = (\varphi_i, \lambda_i, v_i, r_i)^T \in \mathbb{R}^4 \quad (1)$$

and \mathbf{T} is updated as \mathbf{Z} denoted by

$$\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_{N-1}, \mathbf{z}_N) \in \mathbb{R}^{(D+1) \times N} \quad (2)$$

Data normalization is conducted to normalize the data in each dimension into range $[-1, 1]$. For example, the dimension of longitude λ is normalized by Eq. (3).

$$\lambda_i' = 2 \left(\frac{\lambda_i - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} \right) - 1 \quad (3)$$

where λ_{\max} and λ_{\min} are the maximum and minimum values of the longitudinal feature in training set.

Windowing operations is performed as it has been validated that windowing processing could smooth the noise in a relevant study [44]. Suppose a time window size ω is set to move \mathbf{z}_i along the time axis. The windowed data point \mathbf{x}_i and time series \mathbf{X} are denoted by

$$\mathbf{x}_i = (\varphi_i', \lambda_i', v_i', r_i', \dots, \varphi_{i+\omega-1}', \lambda_{i+\omega-1}', v_{i+\omega-1}', r_{i+\omega-1}')^T \in \mathbb{R}^{4+\omega} \quad (4)$$

and

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_X-1}, \mathbf{x}_{N_X}) \in \mathbb{R}^{(4+\omega) \times N_X} \quad (5)$$

where $N_X = N - \omega + 1$, ω is an integer and $0 < \omega < N$.

\mathbf{X} is then fed into DSAE, which is a deep neural network stacked by many single sparse autoencoders (SAE). Each single SAE is layer-wise pre-trained before fine-tuning of the whole network. Suppose the visible layer's vector in the l th SAE is denoted by $\mathbf{v}^{(l)} \in \mathbb{R}^{D_V \times N_X}$, then the hidden layer's vector $\mathbf{h}^{(l)}$ and the reconstruction vector $\mathbf{r}^{(l)}$ are defined as

$$\mathbf{h}^{(l)} = \tanh(\mathbf{W}_{en}^{(l)} \cdot \mathbf{v}^{(l)} + \mathbf{b}_{en}^{(l)}) \in \mathbb{R}^{D_H^{(l)} \times N_X} \quad (6)$$

and

$$\mathbf{r}^{(l)} = \tanh(\mathbf{W}_{de}^{(l)} \cdot \mathbf{h}^{(l)} + \mathbf{b}_{de}^{(l)}) \in \mathbb{R}^{D_R^{(l)} \times N_X} \quad (7)$$

where $\mathbf{W}_{en}^{(l)}$ and $\mathbf{W}_{de}^{(l)}$ are the weights of the l th layer of the encoder and decoder, respectively. $\mathbf{b}_{en}^{(l)}$ and $\mathbf{b}_{de}^{(l)}$ are the biases of the l th layer of the encoder and decoder, respectively.

Then, the reconstruction error is calculated by

$$\min L^{(l)} = \frac{1}{2} \|\mathbf{h}^{(l)} - \mathbf{r}^{(l)}\|_2^2 + \alpha \left(\|\mathbf{W}_{en}^{(l)}\|_2^2 + \|\mathbf{W}_{de}^{(l)}\|_2^2 \right) + \beta \sum_{j=1}^{D_H^{(l)}} \text{KL}(\rho \|\hat{\rho}_j^{(l)}\|) \quad (8)$$

and

$$\text{KL}(\rho \|\hat{\rho}_j^{(l)}\|) = \rho \log \frac{\rho}{\hat{\rho}_j^{(l)}} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j^{(l)}} \quad (9)$$

where the L2-norm penalty item is used to prevent overfitting, and the Kullback-Leibler (KL) divergence is mainly for obtaining a sparse hidden layer to generate more outstanding features. α , β and ρ are the preset hyperparameters to control the corresponding penalty items, and $\hat{\rho}_j^{(l)}$ is the average activation of the units in the l th hidden layer.

A 3-neuron layer is embedded as the output of DSAE to get 3-dimensional hidden features for better visualization, representing the red, green and blue channel in the RGB color space, which is denoted by

$$\mathbf{Y} = (\mathbf{o}_r, \mathbf{o}_g, \mathbf{o}_b)^T \in \mathbb{R}^{3 \times N_Y} \quad (10)$$

where $N_Y = N - \omega + 1$.

The red channel \mathbf{o}_r is normalized into range $[0, 255]$ using Eq. (11).

$$\mathbf{R} = \text{Round} \left(\frac{\mathbf{o}_r - \min(\mathbf{o}_r)}{\max(\mathbf{o}_r) - \min(\mathbf{o}_r)} \times 255 \right) \quad (11)$$

and similarly for the green channel (\mathbf{G}) and blue channel (\mathbf{B}).

Then the *color trajectory* (CT) of the trip \mathbf{T} is denoted by

$$\mathbf{CT} = (\mathbf{R}, \mathbf{G}, \mathbf{B})^T \in \mathbb{R}^{3 \times N_{CT}} \quad (12)$$

where $N_{CT} = N - \omega + 1$.

3.3. Offline anomalous traffic patterns detection (OFF-ATPD)

For the i th complete trajectory, we define τ_i as

$$\tau_i = (N_{CT_i}, \mathbf{CT}_i) = (N_{CT_i}, (\mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i)^T) \quad (13)$$

where $N_{CT_i} = N_i - \omega + 1$ is a temporal feature that highly depends on the trajectory duration N_i . A larger N_{CT} indicates that traffic anomaly might have occurred with higher confidence. However, a trip with a normally ranged N_{CT_i} might also be affected by traffic anomalies. Here, τ_i is referred to as a *trajectory representation*.

We choose a trajectory representation τ_k as *exemplar*. We recommend to choose one with a relatively small N_{CT} , as it is more unlikely to be anomaly. We denote $s(\tau_i, \tau_k)$ to represent the similarity between \mathbf{CT}_i and \mathbf{CT}_k (i.e., the color trajectory of the exemplar). If $s(\tau_i, \tau_k)$ is lower, then it is more similar between \mathbf{CT}_i and \mathbf{CT}_k . To compute the similarity, there is a pre-condition that $N_{CT_i} = N_{CT_k}$. If $N_{CT_k} < N_{CT_i}$, we append $N_{CT_i} - N_{CT_k}$ number of points of white color (rgb(255, 255, 255)) to \mathbf{CT}_k to construct a new trajectory representation τ_j to make $N_{CT_i} = N_{CT_j}$, while the temporal feature N_{CT_k} stays the same.

$$\tau_j = (N_{CT_k}, \mathbf{CT}_j) = (N_{CT_k}, (\mathbf{R}_j, \mathbf{G}_j, \mathbf{B}_j)^T) \quad (14)$$

Similarly, if $N_{CT_k} > N_{CT_i}$, we do the same processing on \mathbf{CT}_i , and then get τ_m .

$$\tau_m = (N_{CT_i}, \mathbf{CT}_m) = (N_{CT_i}, (\mathbf{R}_m, \mathbf{G}_m, \mathbf{B}_m)^T) \quad (15)$$

Then, the similarity between \mathbf{CT}_i and \mathbf{CT}_k can be derived by Eq. (16), when $N_{CT_i} = N_{CT_k}$ or Eq. (17), when $N_{CT_i} \neq N_{CT_k}$.

$$s(\tau_i, \tau_k) = \sum_{n=1}^{N_{CT_i}} \left(\frac{(\mathbf{R}_i^n - \mathbf{R}_k^n)^2 + (\mathbf{G}_i^n - \mathbf{G}_k^n)^2 + (\mathbf{B}_i^n - \mathbf{B}_k^n)^2}{255^2 + 255^2 + 255^2} \right) \quad (16)$$

$$s(\tau_i, \tau_k) = \begin{cases} s(\tau_i, \tau_j) & \text{if } N_{CT_i} > N_{CT_k} \\ s(\tau_k, \tau_m) & \text{if } N_{CT_i} < N_{CT_k} \end{cases} \quad (17)$$

Let $d_{ab}^n = \frac{(\mathbf{R}_a^n - \mathbf{R}_b^n)^2 + (\mathbf{G}_a^n - \mathbf{G}_b^n)^2 + (\mathbf{B}_a^n - \mathbf{B}_b^n)^2}{255^2 + 255^2 + 255^2}$. Given a small positive threshold ε , if the similarity between two color

points is smaller than ε , we ignore the nuance and redefine the similarity as 0. Therefore, we have Eq. (18) in Eq. (17).

$$d_{ab}^n = \begin{cases} d_{ab}^n & \text{if } d_{ab}^n \geq \varepsilon \\ 0 & \text{if } d_{ab}^n < \varepsilon \end{cases} \quad (18)$$

For the i th complete trajectory, we have

$$\epsilon_i = (N_{CT_i}, s(\tau_i, \tau_k)) \quad (19)$$

where $s(\tau_i, \tau_k)$ is a spatial feature since it is mainly extracted from buses' GPS spatial positional information, and it can capture the spatial distribution of the moving object.

By mapping all ϵ to a two-dimensional space which is referred to as a *spatio-temporal plane* here, we are able to detect those two classes of traffic anomalies defined in Section 3.1: *class A anomaly* and *class B anomaly*.

The major differences between *class A anomaly* and *class B anomaly* lies in their neighbors definition and the measurement of similarities between their neighbors. *Class A anomaly* considers both spatial and temporal features to define its neighbors and to measure their similarities. However, if the temporal difference between its neighbors is not significant, there might also be abnormal patterns among them. Therefore, *class B anomaly* reveals this abnormal patterns by addressing the spatial differences from its temporal neighbors. Specifically, if there are several bus trajectories possessed the same or similar temporal features (same or similar trajectory durations), we take them as mutual temporal neighbors. However, if the spatial distribution of one of them is significantly different from the rest, it is understandable that there might be some anomalous events that changed the spatial distribution of this trajectory. Such spatial distribution could be reflected by the spatial feature $s(\tau_i, \tau_k)$ aforementioned.

Spatio-temporal outliers' co-ordinate points can be detected using our proposed offline anomalous traffic patterns detection (OFF-ATPD) algorithm (Algorithm 1), where steps 1 to 11 divide the whole training set into different subsets for class A anomaly detection ($\epsilon_{train.C1}$) and class B anomaly detection ($\epsilon_{train.C2}$) by adopting a threshold N_C . For class B anomaly detection (i.e., $\epsilon_i < N_C$), we employ the Boxplot rule with a parameter δ to identify anomalous observations by aggregating all the spatial features of ϵ_i as well as its forward and backward temporal neighbors within η steps (i.e., temporal feature located in $N_{CT_i} \pm \eta$) to form S (steps 12 to 25). On the other hand, class A anomaly can be detected by computing the Euclidean distance from the nearest spatio-temporal neighbor under a threshold r (steps 26 to 31).

295

Algorithm 1 OFF-ATPD algorithm

Parameters: N_C, δ, r, η .
Input: $\epsilon_{train}, \epsilon_i$. // ϵ_i is for test
Output: C_i . // True denotes anomaly

```

1:  $m \leftarrow 0, n \leftarrow 0$ ;
2: for  $\epsilon_j \in \epsilon_{train}$  do
3:    $N_{CT_j} \leftarrow$  Get the temporal feature of  $\epsilon_j$ ;
4:   if  $N_{CT_j} \geq N_C$  then
5:      $m \leftarrow m + 1$ ;
6:      $\epsilon_{train.C1}(m) \leftarrow \epsilon_j$ ;
7:   else
8:      $n \leftarrow n + 1$ ;
9:      $\epsilon_{train.C2}(n) \leftarrow \epsilon_j$ ;
10:  end if
11: end for
12: if  $\epsilon_i < N_C$  then
13:    $TN \leftarrow$  Search the forward and backward temporal neighbors of  $\epsilon_i$  from  $\epsilon_{train.C2}$  within steps of  $\eta$ ;
14:    $S \leftarrow$  Aggregate all the similarities of  $\epsilon_i$  and members in  $TN$ ;
15:    $Q_1 \leftarrow$  Compute the first quartile of  $S$ ;
16:    $Q_3 \leftarrow$  Compute the third quartile of  $S$ ;
17:    $IQR \leftarrow Q_3 - Q_1$ ;
18:    $U \leftarrow Q_3 + \delta * IQR$ ;
19:    $L \leftarrow Q_1 - \delta * IQR$ ;
20:   if  $S(\epsilon_i) > U$  or  $S(\epsilon_i) < L$  then
21:      $C_i \leftarrow$  True;
22:   else
23:      $C_i \leftarrow$  False;
24:   end if
25: else
26:    $D \leftarrow$  Compute the distance between  $\epsilon_i$  and its nearest spatio-temporal neighbor in  $\epsilon_{train.C1}$ ;
27:   if  $D > r$  then
28:      $C_i \leftarrow$  True;
29:   else
30:      $C_i \leftarrow$  False;
31:   end if
32: end if

```

335 **3.4. Insight analysis using anomalous patterns**

We combine the trajectory \mathbf{T} and the color trajectory (\mathbf{CT}) in Eq. (12) to construct a *color trajectory map* (CTM) through conducting GIS fusion with \mathbf{CT} . Note that we have $N_{CT} < N$ after a window size ω was introduced in the windowing process. Then we construct a *location vector* \mathbf{l}_i and a *location matrix* \mathbf{L} :

$$\mathbf{l}_i = (\varphi_i, \lambda_i)^T \in \mathbb{R}^2 \quad (20)$$

and

$$\mathbf{L} = (\mathbf{l}_{\lfloor \frac{w-1}{2} \rfloor + 1}, \mathbf{l}_{\lfloor \frac{w-1}{2} \rfloor + 2}, \dots, \mathbf{l}_{\lfloor \frac{w-1}{2} \rfloor + N - \omega}, \mathbf{l}_{\lfloor \frac{w-1}{2} \rfloor + N - \omega + 1}) \in \mathbb{R}^{2 \times N_L} \quad (21)$$

where $N_L = N_{CT} = N - \omega + 1$.

We also combine the location matrix \mathbf{L} with \mathbf{CT} to generate \mathbf{L}' :

$$\mathbf{L}' = (\mathbf{L}, \mathbf{CT}) \quad (22)$$

For each \mathbf{L}'_i , map the color with the value of $(\mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i)^T$ to coordinate $(\varphi_i, \lambda_i)^T$ on the GIS map, so as to generate the CTM of a whole trajectory.

$$\mathbf{L}'_i = ((\varphi_i, \lambda_i)^T, (\mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i)^T) \quad (23)$$

The color trajectory (i.e., \mathbf{CT} in Eq. (12)) and CTM are linked via the conjunct RGB values. By comparing the \mathbf{CT} of an anomalous trajectory with those non-anomalous trajectories, the most significant difference between them can be found, and the corresponding sector of these colors can be regarded as anomalous. Then, the anomaly occurring location as well as the road influence sector are estimated, by locating the coordinate $(\varphi_i, \lambda_i)^T$ on the CTM via the anomalous colors $(\mathbf{R}_i, \mathbf{G}_i, \mathbf{B}_i)^T$ obtained from last step.

365 **3.5. Online detection of anomalous traffic patterns (ON-ATPD)**

The proposed OFF-ATPD method in Section 3.3 takes the complete bus trajectories as input. It is an offline detection mechanism because the data is ready only after the bus completes the whole trip from the origin place to the terminal stop. In this section, we propose an online anomalous traffic patterns detection (ON-ATPD) algorithm (Algorithm 2), which is a substantial extension to the OFF-ATPD algorithm.

By Algorithm 2, \mathbf{X}_t is the real-time input at timestamp t , derived from Eq. (5). Step 1 computes the color trajectory of input \mathbf{X}_t with DSAE (see details in Section 3.2). Step 2 tests whether the bus has arrived at the terminal of the trip or not. Steps 3 and 4 go to the OFF-ATPD algorithm when the bus reaches the terminal stop. While the bus is still on the way to the terminal stop, steps 6 to 13 append or remove segments from the current color trajectory \mathbf{CT}_t by comparing with the most similar color trajectory from the training set ϵ_{train} . Steps 14 and 15 calculate $\epsilon_{t'}$ with the newly constructed color trajectory $\mathbf{CT}_{t'}$ and apply the OFF-ATPD algorithm for anomaly detection. Since we have defined the nearest neighbor (the most similar) color trajectory of the real-time \mathbf{CT}_t (by step 6), there might be a situation that patterns of the nearest neighbor are quite different from the original complete color trajectory. In order to improve

the reliability of online anomaly detection, we introduce an integer parameter n to decide whether all abnormal patterns adjudged from the the previous $n - 1$ detections and the current detection can yield an anomaly report (steps 16 to 20).

Algorithm 2 ON-ATPD algorithm

Parameters: N_C, δ, r, η, n .

Input: $\epsilon_{train}, \mathbf{X}_t, t \geq n$.

Output: $C_t, t \geq n$. // True denotes anomaly

```

1:  $\mathbf{CT}_t \leftarrow$  Get the color trajectory of  $\mathbf{X}_t$  with DSAE;
2: if  $t$  is the end timestamp of the trip then
385 3:  $\epsilon_t \leftarrow$  Compute the temporal and spatial features
of  $\mathbf{CT}_t$  by Eq. (19) and go to OFF-ATPD algorithm;
4:  $C_t \leftarrow \text{OFF-ATPD}(N_C, \delta, r, \eta, \epsilon_{train}, \epsilon_t)$ ;
5: else
6:  $\mathbf{CT}_{t'} \leftarrow$  Get the most similar color trajectory of
390  $\mathbf{CT}_t$  from  $\epsilon_{train}$ ;
7:  $N_{CT_t} \leftarrow$  Get the temporal feature of  $\mathbf{CT}_t$ ;
8:  $N_{CT_{t'}} \leftarrow$  Get the temporal feature of  $\mathbf{CT}_{t'}$ ;
9: if  $N_{CT_t} < N_{CT_{t'}}$  then
10:  $\mathbf{CT}_{t''} \leftarrow$  Append  $\mathbf{CT}_t$  with the last  $N_{CT_{t'}} -$ 
395  $N_{CT_t}$  color points of  $\mathbf{CT}_{t'}$ ;
11: else
12:  $\mathbf{CT}_{t''} \leftarrow$  Remove the last  $N_{CT_t} - N_{CT_{t'}}$  color
445 points of  $\mathbf{CT}_{t'}$ ;
13: end if
14:  $\epsilon_{t''} \leftarrow$  Compute the temporal and spatial features
of  $\mathbf{CT}_{t''}$  and go to OFF-ATPD algorithm;
15:  $C_t \leftarrow \text{OFF-ATPD}(N_C, \delta, r, \eta, \epsilon_{train}, \epsilon_{t''})$ ;
16: if  $C_{t-n+1}, \dots, C_{t-1}, C_t$  are all True then
17:  $C_t \leftarrow$  True;
18: else
405 19:  $C_t \leftarrow$  False;
20: end if
21: end if

```

4. Experiments and analyses

We have performed comprehensive experiments to answer the following research questions:

RQ1: Is OFF-ATPD effective and sensitive to detect all anomalies (i.e., with a high detection rate)?

RQ2: Is our developed feature visualization method useful for capturing anomaly locations and traffic impacts with the detected anomalies?

RQ3: How does our proposed ON-ATPD perform in real-time traffic anomaly detection?

RQ4: How well do our proposed feature extraction deep architecture and anomaly detection methods perform in comparison with the state-of-the-art methods?

4.1. Experimental settings

4.1.1. Datasets

We use trajectory datasets from 3 bus routes in Guiyang (China) with a duration of 4 months in the year of 2016. All the data (including the local hourly precipitation data) is officially provided by the Guiyang Open Government Data Platform². The first two datasets are collected on weekends, while the last one is from the off-peak hours (except the morning peak from 06:00 to 09:00 and afternoon peak from 17:00 to 19:30) on weekdays. Each dataset is divided into a training set (the first 3 months) and a test set (the following month). All datasets are naturally unbalanced, since traffic anomalous event rarely occurs along the same bus route. The imbalanced ratios (minority/majority) are 0.025, 0.014 and 0.007 for the test sets of Route 66, Route 50 and Route 18, respectively. Table 1 provides a detailed description about these datasets.

4.1.2. Parameters

The parameters are set as: $(\omega, \alpha, \beta, \rho, \epsilon) = (10, 10^{-5}, 10^{-4}, 0.05, 0.01)$ for all the bus routes. The window size ω cannot be set with either too big or too small value, we choose 10 as suggested by the literature work [44]. We set ρ with a value near 0 because the centre of each RGB space axis is 0. In addition, the values of α, β and ϵ are set empirically, but without using a specific parameters tuning method. Parameter δ is a key parameter for detection performance, since too high or too low δ will result in a low detection rate or a high false alarm rate (as illustrated by Fig. 4). The default value range to determine the upper and lower fences is 1.5 in Boxplot rule. We fine tune the value of δ around 1.5. The parameters in both algorithms OFF-ATPD and ON-ATPD are set as the same: $(N_C, \delta, r, \eta) = (450, 2.0, 50, 2)$ for Bus Route 66, $(N_C, \delta, r, \eta) = (500, 1.7, 50, 2)$ for Route 50 and $(N_C, \delta, r, \eta) = (350, 0.9, 40, 2)$ for Route 18, with the understandings and trials from the training set. The Bus Route 18 utilizes a smaller value of N_C as its route is shorter. The setting of the other parameters in algorithm ON-ATPD is discussed in Section 4.4. Moreover, we employ a DSAE of four encoding layers with dimensions $40 \rightarrow 20 \rightarrow 10 \rightarrow 3$ to identify the 3-dimensional hidden features³.

4.1.3. Evaluation metrics

In the performance evaluation, we use measurements accuracy (Acc), detection rate (DR), false alarm rate

²<http://www.gyopendata.gov.cn/city/index.htm>

³The layer number and neuron number can be changed. However, the network should be a deep learning architecture.

Table 1. Datasets description

Route	Day Type	Whole Sample	Training Sample	Test Sample	Training Set Period	Test Set Period	Input Size
66	Weekend	486	324	162	1 Aug.~ 31 Oct.	1 Nov.~ 30 Nov.	118041×40
50	Weekend	1304	950	354	1 Aug.~ 31 Oct.	1 Nov.~ 30 Nov.	406030×40
18	Weekday Off-peak	1117	824	293	1 Sept.~ 30 Nov.	1 Dec.~ 31 Dec.	238555×40

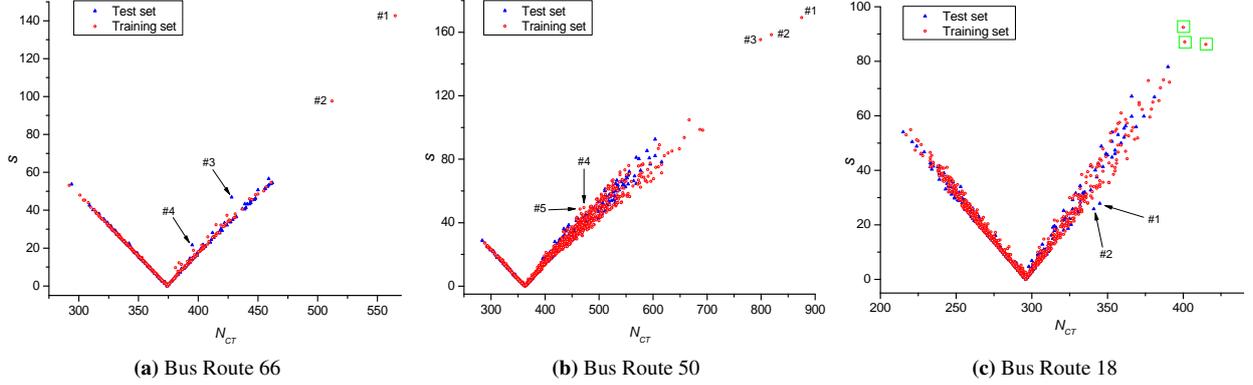


Fig. 2. Spatio-temporal planes of Bus Routes 66, 50 and 18. The Y-axis s stands for the spatial feature which is computed by Eq. (16) or (17), N_{CT} is the temporal feature obtained by $N - \omega + 1$. The objects inside \square were not detected as anomalies, because their spatial and temporal features are not far away from their spatio-temporal neighbors.

(FAR) [45] and area under the ROC curve (AUC). Criteria of Acc, DR and FAR are calculated as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (24)$$

$$DR = \frac{TP}{TP + FN} \quad (25)$$

$$FAR = \frac{FP}{FP + TN} \quad (26)$$

- True Positive (TP): the number of anomalous trajectory correctly detected as anomaly;
- True Negative (TN): the number of non-anomalous trajectory correctly identified as non-anomaly;
- False Positive (FP): the number of non-anomalous trajectory incorrectly identified as anomaly;
- False Negative (FN): the number of anomalous trajectory falsely identified as non-anomaly.

We also define an index named averaged moving standard deviation (AMSD) to evaluate the concentration of the majority samples (negative samples), which is also a criterion for evaluating the hidden feature extraction architecture. A lower AMSD indicates that those non-anomalies are closer to their neighbors. However, from an overall perspective, a higher AMSD value

shows that those non-anomalies are more dissimilar to each other. Method with a higher AMSD might make more false detections, which we should try to avoid in this study. The definition of AMSD can be referred to Eq. (27). Firstly, a window size κ for the windowing operation along the horizontal axis N_{CT} is employed here. Then we compute the sample standard deviation of all the normalized $s(\tau_{ij}, \tau_k)$ (denoted as $\hat{s}(\tau_{ij}, \tau_k)$) within each κ -sized N_{CT} . Following this, we get the mean standard deviation of all κ -sized N_{CT} for AMSD.

$$AMSD = \frac{1}{m} \sum_{i=1}^m \sqrt{\frac{1}{n_i - 1} \sum_{j=1}^{n_i} (\hat{s}(\tau_{ij}, \tau_k) - \bar{s}_i)^2} \quad (27)$$

4.2. Offline detection results about anomalous patterns (answering RQ1)

The performance comparisons between our proposed OFF-ATPD versus the state-of-the-art baselines are listed in Table 4 (note that we have transferred the anomalous observations from the training set to the test set to enlarge the positive sample size for performance evaluation). The proposed OFF-ATPD detects all known anomalies with a high accuracy and a low false alarm rate. The spatio-temporal planes for Bus Routes 66, 50 and 18 are shown in Fig. 2, where those points distributed along the tick (\checkmark) sign exhibit a trend

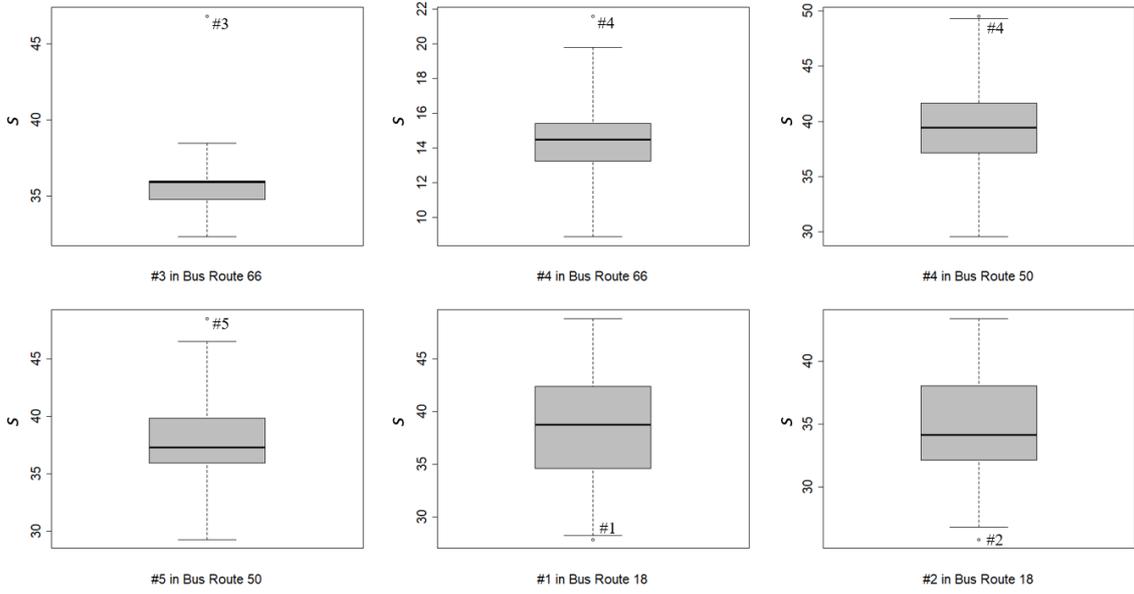


Fig. 3. Illustration of the process of class B anomalies detection (steps 12 to 25 in Algorithm 1). The isolated points that are smaller than the lower fence or larger than the upper fence are identified as anomalies.

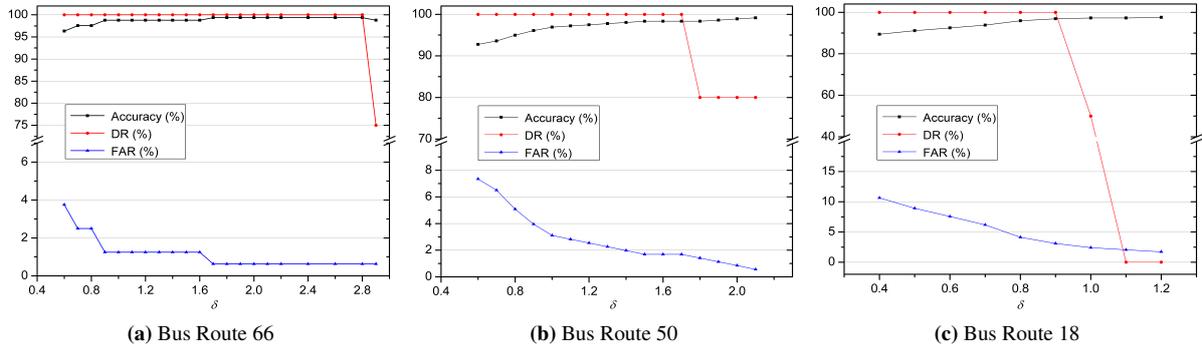


Fig. 4. Performance comparisons under different settings of parameter δ .

Table 2. Detected anomalies for each bus route

Bus Route	Anomaly	Service Date	Running Time	Event	Anomaly Category
66	#1	18 Sept. 2016	07:30 AM-09:01 AM	Event 2	Class A anomaly
	#2	18 Sept. 2016	07:00 AM-08:23 AM	Event 2	Class A anomaly
	#3	26 Nov. 2016	12:43 PM-01:55 PM	Event 3	Class B anomaly
	#4	26 Nov. 2016	12:07 PM-01:09 PM	Event 3	Class B anomaly
50	#1	18 Sept. 2016	06:58 AM-09:25 AM	Event 2	Class A anomaly
	#2	18 Sept. 2016	07:22 AM-09:40 AM	Event 2	Class A anomaly
	#3	18 Sept. 2016	07:34 AM-09:48 AM	Event 2	Class A anomaly
	#4	14 Aug. 2016	07:41 PM-09:01 PM	Event 1	Class B anomaly
	#5	14 Aug. 2016	05:32 PM-06:51 PM	Event 1	Class B anomaly
18	#1	14 Dec. 2016	09:31 AM-10:28 AM	Event 4	Class B anomaly
	#2	14 Dec. 2016	09:50 AM-10:48 AM	Event 4	Class B anomaly

that the s similarity increases with N_{CT} when $N_{CT} > N_{CT_k}$, while it decreases with N_{CT} when $N_{CT} < N_{CT_k}$. Also in Fig. 2 (a), anomalies #1 and #2 are categorized as *class A anomalies* as their spatial and temporal features are both far away from their spatio-temporal neighbors, and similarly for anomalies #1, #2, and #3 in Fig. 2 (b).

However, anomalies #3, #4 in Fig. 2 (a), #4, #5 in Fig. 2 (b) and #1, #2 in Fig. 2 (c) were detected as *class B anomalies*, because only their spatial features are far away from their temporal neighbors. In general, class A anomaly has more serious impact on traffics than class B anomaly does, while class B anomaly is more difficult to identify. Fig. 3 illustrates the processes of detecting class B anomalies, with steps 12 to 25 in Algorithm 1. Fig. 4 presents the performance on all the datasets under different settings of parameter δ . With the increase of δ , a higher accuracy and lower false alarm rates were achieved for all datasets. However, when δ rises above a threshold (e.g., $\delta > 2.8$ for Bus Route 66), the detection rate decreases, while it remains high when δ is below the threshold.

The detected anomalies shown in Table 2 are all coincided with the known traffic anomalous events, which are elaborated as follows:

Known event 1: A sedan bumped a car at Shachong East Road in the late afternoon of 14 August 2016, the driver of the sedan escaped after the accident resulting in serious traffic congestion⁴. It was raining at that time and this event only affected services for Bus Route 50.

Known event 2: A severe car crash (an SUV and a truck) occurred on the West No.2 Ring Road in the morning of 18 September 2016. Two men died on site and one got injured⁵. This event imposed impacts on Bus Route 66 and Route 50 bus services.

Known event 3: Two cars crashed at a bus station near the Guizhou Cancer Hospital (West Beijing Road) around the noon on 26 November 2016. A pedestrian died⁶. This event affected Bus Route 66 service.

Known event 4: An SUV crashed an electric motorcycle on the North Wenchang Avenue in the morning of 14 December 2016. Two riders on the electric motorcycle got injured while trapping under the vehicle⁷. Only Bus Route 18 service was influenced by this crash.

⁴http://www.gywb.cn/content/2016-08/16/content_5188212.htm

⁵http://www.sohu.com/a/114567218_398062

⁶https://m.sohu.com/n/474230721/?wscrid=53843_3&_smuid=BnKG38irJV6gorGDwjyzS0&mv=2

⁷<http://gz.sina.com.cn/news/sh/2016-12-15/detail-ifxytqav9265554.shtml>

4.3. Results about feature visualization and anomaly insight analysis (answering RQ2)

Fig. 5 (a), (e) or (i) depicts the CT of a real-world trajectory in Bus Route 66, 50 or 18, respectively. It is evident from Fig. 5 (a) that the bus trajectory starts at the color of yellow ; the color changes gradually and finally gets to blue  when the bus is approaching to the terminal stop. The horizontal axis indicates the temporal feature (N_{CT} , 1 unit equals 10 seconds, each row contains 100 units).

The CTM of anomalous trajectory is obtained by fusion of the color trajectory (CT) with the GIS map (via Eq. (22) and Eq. (23)). Here we illustrate an anomalous trajectory by taking the anomaly #1 in Bus Route 66 as example. As shown in Fig. 6 (a), subfigure (i) is the CTM of #1, and  denotes the actual event site. By contrasting the CT of anomaly #1 (i.e., subfigure (ii)) and non-anomalies (i.e., subfigure (iii) and (iv)), we can have an intuitive perspective that the anomaly might have occurred around light yellow , because the part with such color is very different from those of the non-anomalies. However, when it proceeds to the color of grey , the rest part of CT turns to be similar to those of non-anomalies. It means that the anomaly happened at the locations highlighted between locations  and  in Fig. 6 (a), which is in line with the real location () of event 2.

Apart from location detection, our method also provided insights to understand implications of the car crash on the road by highlighting the road section between  and  (at the left bottom of Fig. 6 (a)). Similarly, Fig. 6 (b) visually illustrates another example of anomaly #4 in Bus Route 50 that happened between the color of bright red  and dark red , which also coincides with the real site () of event 1.

4.4. Online detection results about anomalous patterns (answering RQ3)

We conducted online detection simulation experiments for all trajectories in test sets. The online anomaly report is carried out every 3 minutes. The parameters N_C , δ , r and η set for ON-ATPD are the same as those used by OFF-ATPD. Parameter n is tested with different values from 1 to 3. Table 3 shows the performance of our proposed online detection algorithm. All of the known anomalies are detected correctly. In particular with the increase of parameter n , higher accuracies and lower false alarm rates can be achieved on all datasets. On average for each detection, the method needs about 3 seconds of computational time for each detection in Bus Route 66, while needs about 7 seconds

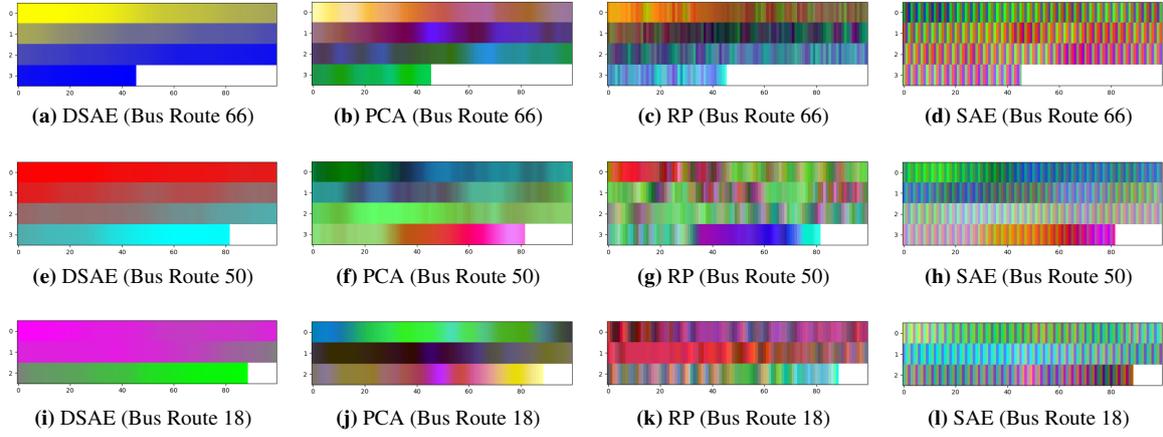


Fig. 5. Examples of color trajectories generated by DSAE, PCA, RP or SAE. Each method generates similar visualization patterns on all of these datasets. The CT visualizations generated by our proposed DSAE model are the smoothest.

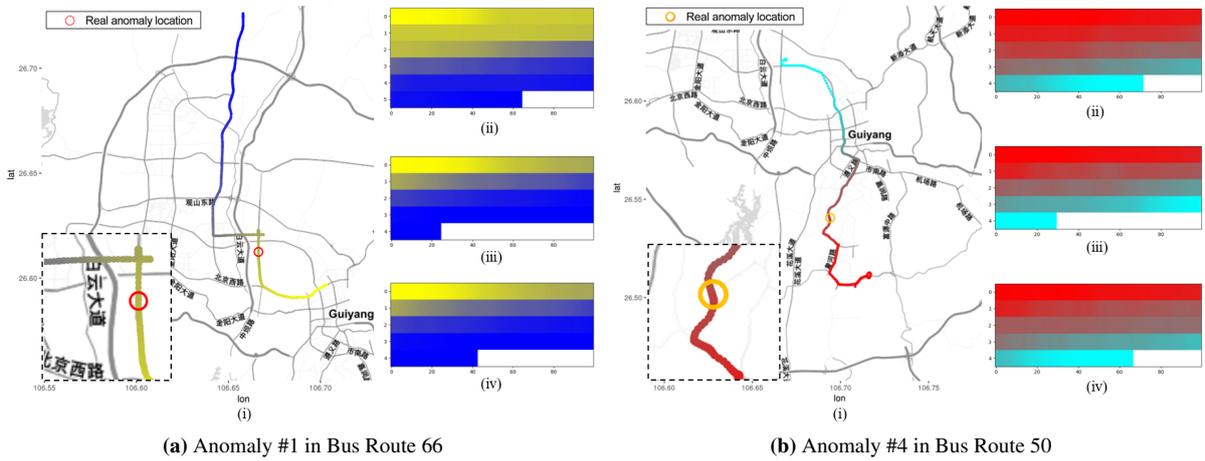


Fig. 6. Insight analyses for anomaly #1 in Bus Route 66 and anomaly #4 in Bus Route 50. (i) CTM of the anomalous trajectory. (ii) CT of the anomalous trajectory. (iii) CT of a non-anomalous trajectory. (iv) CT of another non-anomalous trajectory.

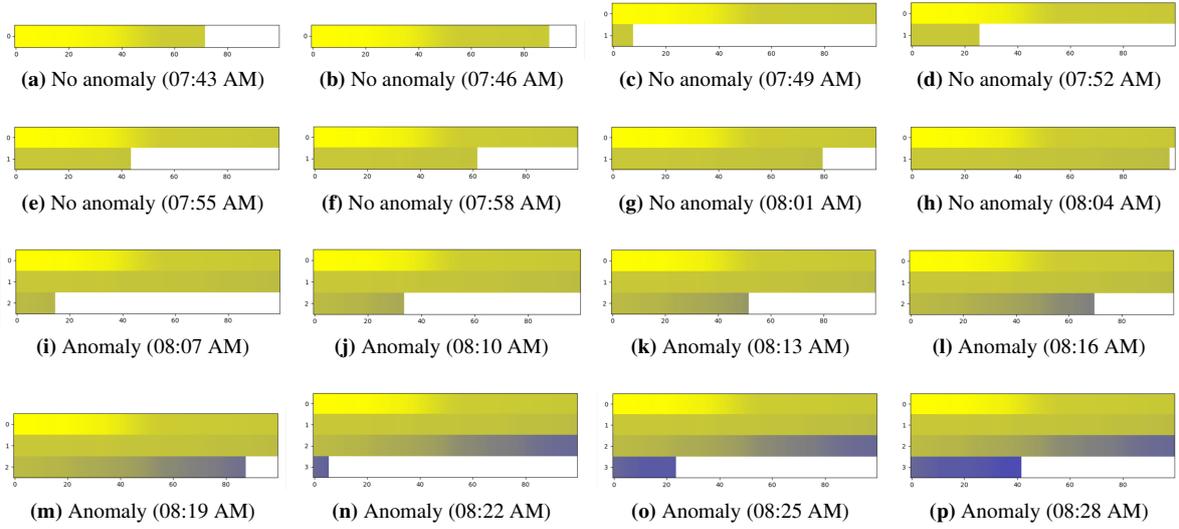


Fig. 7. An illustrative example of online traffic anomaly detection process in Bus Route 66.

580 and 12 seconds in Bus Route 18 and Route 50 detection, respectively.

Table 3. Performance of the proposed online anomaly detection method (ON-ATPD)

Route	Parameter	Acc (%)	DR (%)	FAR (%)	Time
66	n=1	94.51	100	5.63	3.2s
	n=2	95.12	100	5.00	3.2s
	n=3	97.56	100	2.50	3.2s
50	n=1	91.92	100	8.19	12.1s
	n=2	92.76	100	7.34	12.0s
	n=3	95.26	100	4.80	12.3s
18	n=1	74.40	100	25.77	6.5s
	n=2	76.45	100	23.71	6.5s
	n=3	87.71	100	12.37	6.6s

Notes: Time is the mean computational time for one detection. Our experiments were conducted on the server with Intel Xeon Gold 6150 of 2.7GHz.

585 Fig. 7 illustrates some sequential steps of the online detection of anomaly #1 in the Route 66 dataset, where the real-time color trajectories with an interval of 3 minutes are displayed at corresponding timestamp. A detection result of ‘Anomaly’ or ‘No anomaly’ indicates whether there exists any anomaly for the current trajectory. For anomaly #1, the detection system is alarmed around 08:07 AM with an anomaly reporting, when the bus is located at the color ■. Comparing with the real anomaly location shown in Fig. 6 (a), the detected site at color ■ is quite close to the real anomaly location ■.
590
620

4.5. Comparisons with baseline methods (answering RQ4)

595 4.5.1. Feature extraction and visualization

Our deep learning-based feature extraction method DSAE is compared with other popular baseline methods including PCA, random projection (RP) and sparse auto-encoder (SAE) to understand the quality of our color trajectories (CT). From Fig. 5, it is apparent that our DSAE-based model can generate the smoothest color distributed trajectories. In Fig. 5 (a)(e) and (i), with the trajectory moves on, it gradually changes from one color to another distinct color. While the trajectories by the rest baselines sometimes switch back to the previous color at certain parts of the CT. This conflict will make it difficult for anomaly insight analyses when they are overlapped on the GIS map. Furthermore, on the spatio-temporal planes derived by the above baseline methods, none of them can get better detection performance than the DSAE-based method for all the datasets (Fig. 8). The distribution of some known anomalies (especially the class B anomalies) yields a similar pattern with that of non-anomalies (#3, #4 in Bus Route 66 by PCA and SAE, #4, #5 in Bus Route 50 by RP and SAE, #1, #2 in Bus Route 18 by PCA, RP and SAE), which makes difficulties to clearly distinguish between anomalies and non-anomalies. Moreover, many of the known non-anomalies are obviously mapped as isolated outlier points (labeled ○ in Fig. 8), which do not exhibit the characteristics of the expected patterns.
610
615
620

We also calculated the AMSD values (see Section

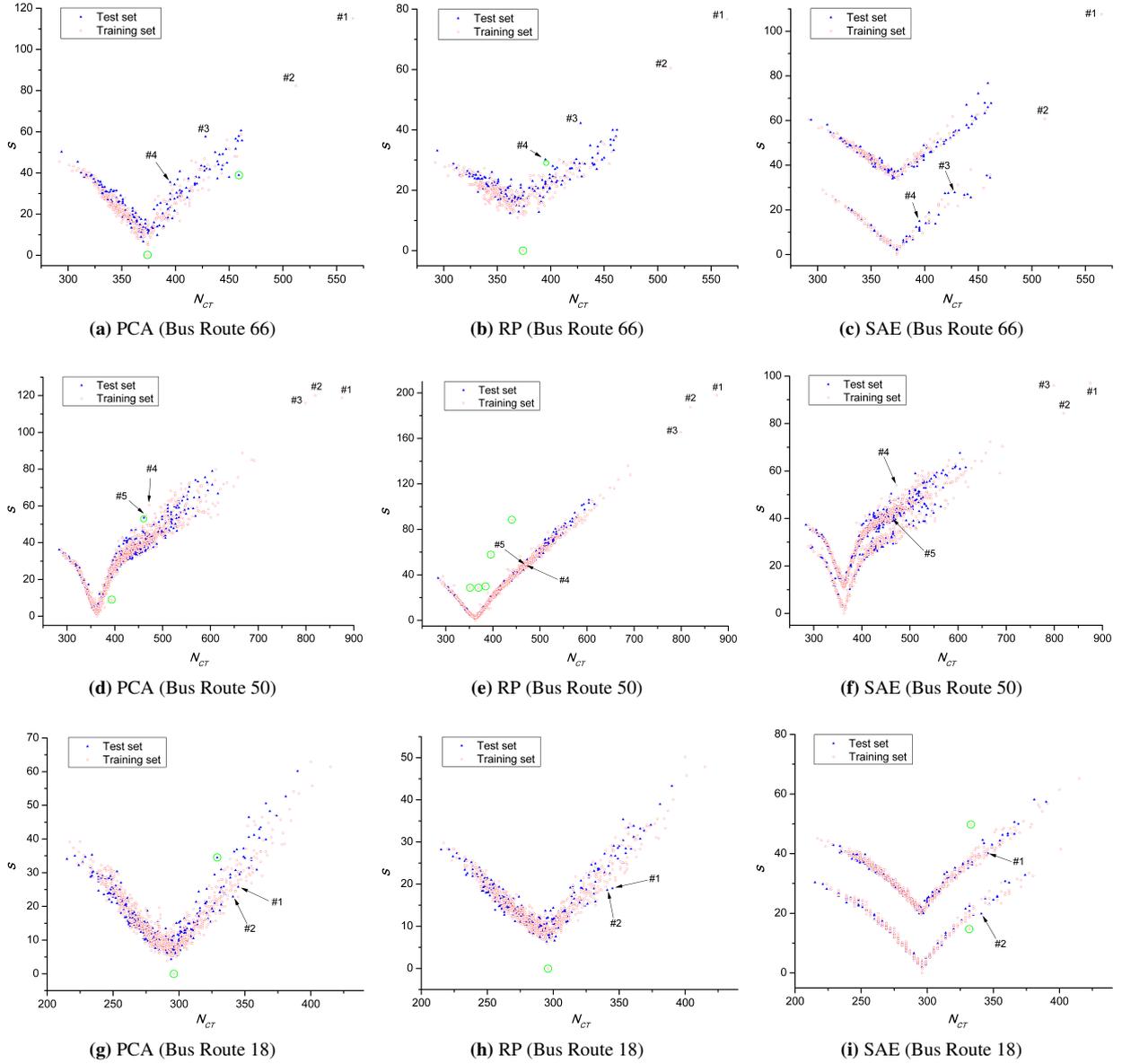


Fig. 8. Spatio-temporal planes derived from the baseline methods PCA, RP and SAE on different dataset. Objects with \circ represent those points which can be regarded as anomalies with high confidence, while they are not anomalies in reality. Moreover, the distribution of some known class B anomalies yields similar patterns with these non-anomalies, which makes them difficult to identify.

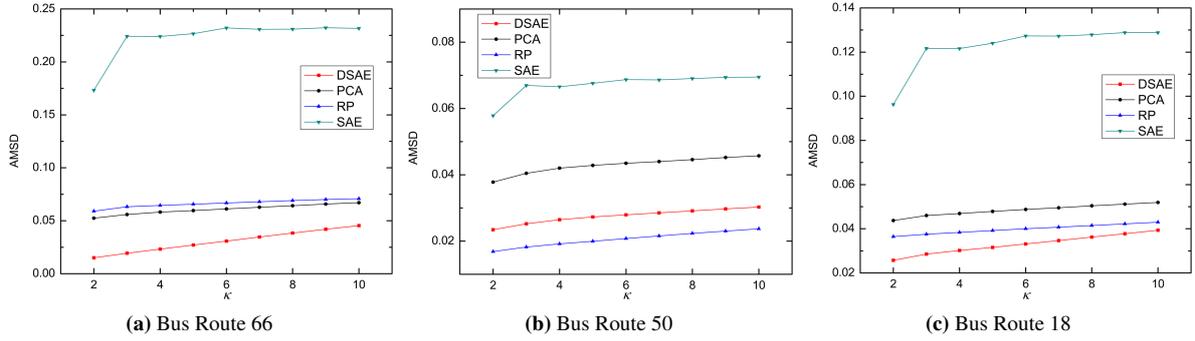


Fig. 9. Comparison of concentration performance on the training sets.

4.1.3) for all of the non-anomalies, under every window size κ from 2 to 10. DSAE-based model achieved the best performance on the datasets of Bus Route 66 and Route 18, as shown in Fig. 9. RP obtained fairly good performance on Bus Route 50; however, its performance in anomaly detection is sensitive as it made false predictions on quite a number of points in Bus Route 50. SAE performed the worst on all of the datasets.

4.5.2. Comparison on anomalous traffic patterns detected by our offline approach

We compare the anomaly detection performance by our offline detection approach (OFF-ATPD) with those by the commonly used methods in outlier/anomaly detection [46], including classification-based methods (one-class SVM (OneSVM) [25, 26], binary SVM (BiSVM) and LSTM network), a clustering-based method (HDBSCAN clustering [47]) and a nearest-neighbor-based method (kNN). The same features extracted via DSAE are used for these baseline methods. Our approach is implemented by Python and TensorFlow, the code of our algorithms is publicly available in GitHub repositories⁸. OneSVM and BiSVM use the ‘e1071’ package in R. LSTM network is implemented by the ‘rnn’ package in R. While baselines of HDBSCAN and kNN use the R packages of ‘dbscan’ and ‘FNN’, respectively. The output probabilities of our approach to calculate AUC are linearly scaled by the similarity (i.e., $S(\epsilon_i)$ in Algorithm 1). While the output probabilities via OneSVM and BiSVM are estimated by Platt scaling [48]. The performances are shown in Table 4. Because there is no positive sample in the training set of Bus Route 18, the supervised learning methods

of BiSVM, LSTM and kNN are not applicable. Overall, OFF-ATPD achieved better performances with high accuracies, the 100% detection rates, low false alarm rates and high AUC scores on all of these datasets. OneSVM is also a competitive method that detected all anomalies correctly; however, its high false alarm rates (61.25%, 47.74% and 52.23%) make it less efficient. BiSVM and HDBSCAN also demonstrated low false alarm rates and high AUC scores; nevertheless, they are unable to identify all the anomalies accurately. None of the rest machine learning baseline methods could detect all of the anomalies correctly. One reason is probably that the real-world datasets for traffic anomaly detection as we utilized in this experiment are extremely imbalanced. Machine learning on imbalanced datasets might produce unsatisfactory classifiers [49, 50]. Instead of taking machine learning ideas for pattern recognition, our developed algorithm explores the ideas of spatio-temporal neighborhood and Boxplot rules to identify anomalous traffic patterns in class A task and in class B task, respectively. Because these anomalous patterns have distinct spatial and temporal characteristics, our approach can achieve better performance on imbalanced data than the baseline machine learning approaches.

5. Conclusion and future work

This work has developed novel methods for online and offline detections of anomalous traffic patterns from bus trajectory datasets. Our methods have explored deep learning ideas to extract novel features and the methods can make good visualization of the features as well. Based on the spatial and temporal characteristics of the anomalies, we have termed *class A anomaly* and *class B anomaly* to better address the discrepancy issues between these diversified anomalous patterns. The key idea of our algorithm is to use the Boxplot rule or

⁸<https://github.com/Xiaocai-Zhang/Anomalous-Traffic-Patterns-Detection>

Table 4. Performance comparison on the test sets with the baseline methods

Route	Metric	OFF-ATPD	OneSVM	BiSVM	LSTM	HDBSCAN	kNN
66	Acc (%)	99.39	40.24	98.78	98.17	98.17	98.78
	DR (%)	100	100	50.00	25.00	75.00	50.00
	FAR (%)	0.63	61.25	0	0	1.25	0
	AUC (%)	99.61	75.63	99.53	75.00	97.19	75.00
50	Acc (%)	98.33	52.92	99.44	98.89	99.44	99.44
	DR (%)	100	100	60.00	20.00	60.00	60.00
	FAR (%)	1.69	47.74	0	0	0	0
	AUC (%)	99.66	79.52	100	60.00	97.12	80.00
18	Acc (%)	96.93	48.12	—	—	99.32	—
	DR (%)	100	100	—	—	0	—
	FAR (%)	3.09	52.23	—	—	0	—
	AUC (%)	97.25	75.60	—	—	99.14	—

Notes: Supervised learning method BiSVM, LSTM or kNN cannot be applied to Bus Route 18 dataset since there is no positive sample in the training set. AUC is computed by the ‘sklearn’ package in Python.

690 the nearest neighborhood for different detection tasks
of anomalous patterns. Our methods are also able to
conduct insights analysis on the locations of anomalies
as well as on the traffic influences to the road caused
by the corresponding anomalies. We developed an on-
695 line detection method extending from the offline method
for a real-time detection. Comprehensive experiments
on 3 real-world bus route datasets confirmed the effec-
tiveness and superiority of our deep feature extraction
method, offline and online detection approaches while
700 comparing with the baseline methods PCA, RP, SAE,
one-class SVM, binary SVM, LSTM, HDBSCAN and
kNN.

Future infrastructure plans for some cities have
adopted the ‘Bus Lane’ strategy for main roads during
705 certain periods to improve the reliability and efficiency
of bus services. In that case, our approach may not be
efficient to detect the incident-related anomaly, as the
situation that some incidents affecting other vehicles on
the road might not affect buses. However, from the per-
710 spective of bus service operation or management, that
situation does not make much sense, since those anoma-
lies that impose little impact on bus service will not be
taken into account for decision making. In the future,
we plan to study how to improve the reliability of on-
715 line traffic anomaly detection algorithm and to test our
methods on more datasets. Besides, we will explore
the possibility of our methods on other trajectory data
sources, such as the city-wide taxis or trains trajectory
data.

720 CRediT authorship contribution statement

Xiaocai Zhang: Conceptualization, Data curation,
Methodology, Software, Visualisation, Writing - orig-

inal draft. **Yi Zheng:** Software, Data curation, Visual-
isation. **Zhixun Zhao:** Methodology, Writing - review
& editing. **Yuansheng Liu:** Methodology, Writing -
review & editing. **Michael Blumenstein:** Supervision,
Writing - review & editing. **Jinyan Li:** Supervision,
Methodology, Project administration, Writing - review
& editing.

730 Declaration of competing interest

The authors declare that they have no known com-
peting financial interests or personal relationships that
could have appeared to influence the work reported in
this paper.

735 Acknowledgment

The authors would like to thank the anonymous re-
viewers for their helpful comments and suggestions to
significantly improve the quality of this paper. This
work is financially supported by China Scholarship
740 Council, China and University of Technology Sydney,
Australia.

References

- [1] Y. Li, T. Guo, R. Xia, W. Xie, Road traffic anomaly detection
based on fuzzy theory, *IEEE Access* 6 (2018) 40281–40288.
- [2] W. Liu, Y. Zheng, S. Chawla, J. Yuan, X. Xing, Discovering
spatio-temporal causal interactions in traffic data streams, in:
Proceedings of the 17th ACM SIGKDD International Confer-
ence on Knowledge Discovery and Data Mining, ACM, 2011,
pp. 1010–1018.
- [3] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, A. Tolba, Lotad:
750 Long-term traffic anomaly detection based on crowdsourced bus
trajectory data, *World Wide Web* 21 (3) (2018) 825–847.

- [4] Y. Xu, X. Ouyang, Y. Cheng, S. Yu, L. Xiong, C.-C. Ng, S. Pranata, S. Shen, J. Xing, Dual-mode vehicle motion pattern learning for high performance road traffic anomaly detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 145–152.
- [5] S. Chawla, Y. Zheng, J. Hu, Inferring the root cause in road traffic anomalies, in: Proceedings of 2012 IEEE International Conference on Data Mining, IEEE, 2012, pp. 141–150.
- [6] L. Pang, S. Chawla, W. Liu, Y. Zheng, On detection of emerging anomalous traffic patterns using gps data, *Data & Knowl. Eng.* 87 (2013) 357–373.
- [7] Z. Wang, M. Lu, X. Yuan, J. Zhang, H. Van De Wetering, Visual traffic jam analysis based on trajectory data, *IEEE Trans. Vis. Comput. Graphics* 19 (12) (2013) 2159–2168.
- [8] L. X. Pang, S. Chawla, W. Liu, Y. Zheng, On mining anomalous patterns in road traffic streams, in: Proceedings of International Conference on Advanced Data Mining and Applications, Springer, 2011, pp. 237–251.
- [9] H. Wang, H. Wen, F. Yi, H. Zhu, L. Sun, Road traffic anomaly detection via collaborative path inference from gps snippets, *Sensors* 17 (3) (2017) 1–21.
- [10] J. Mao, P. Sun, C. Jin, A. Zhou, Outlier detection over distributed trajectory streams, in: Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM, 2018, pp. 64–72.
- [11] D. Zhang, N. Li, Z.-H. Zhou, C. Chen, L. Sun, S. Li, ibat: detecting anomalous taxi trajectories from gps traces, in: Proceedings of the 13th International Conference on Ubiquitous Computing, ACM, 2011, pp. 99–108.
- [12] W. Kuang, S. An, H. Jiang, Detecting traffic anomalies in urban areas using taxi gps data, *Mathematical Problems Eng.* 2015 (2015) 1–14.
- [13] Y. Yu, L. Cao, E. A. Rundensteiner, Q. Wang, Detecting moving object outliers in massive-scale trajectory streams, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 422–431.
- [14] L. Song, R. Wang, D. Xiao, X. Han, Y. Cai, C. Shi, Anomalous trajectory detection using recurrent neural network, in: Proceedings of International Conference on Advanced Data Mining and Applications, Springer, 2018, pp. 263–277.
- [15] H. Wu, W. Sun, B. Zheng, A fast trajectory outlier detection approach via driving behavior modeling, in: Proceedings of the 26th ACM International Conference on Information and Knowledge Management, ACM, 2017, pp. 837–846.
- [16] X. Zhang, Z. Zhao, Y. Zheng, J. Li, Prediction of taxi destinations using a novel data embedding method and ensemble learning, *IEEE Trans. Intell. Transp. Syst.* 21 (1) (2019) 68–78.
- [17] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, *ACM SIGCOMM Computer Commun. Rev.* 34 (4) (2004) 219–230.
- [18] Y. Liu, L. Zhang, Y. Guan, Sketch-based streaming pca algorithm for network-wide traffic anomaly detection, in: Proceedings of the 30th International Conference on Distributed Computing Systems, IEEE, 2010, pp. 807–816.
- [19] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano, T. Pepe, A novel pca-based network anomaly detection, in: Proceedings of 2011 IEEE International Conference on Communications (ICC), IEEE, 2011, pp. 1–5.
- [20] A. Juvonen, T. Hamalainen, An efficient network log anomaly detection system using random projection dimensionality reduction, in: Proceedings of the 6th International Conference on New Technologies, Mobility and Security (NTMS), IEEE, 2014, pp. 1–5.
- [21] R. Fontugne, P. Abry, K. Fukuda, P. Borgnat, J. Mazel, H. Wendt, D. Veitch, Random projection and multiscale wavelet leader based anomaly detection and address identification in internet traffic, in: Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2015, pp. 5530–5534.
- [22] G. Münz, S. Li, G. Carle, Traffic anomaly detection using k-means clustering, in: GI/ITG Workshop MMBnet, 2007, pp. 13–14.
- [23] K. Leung, C. Leckie, Unsupervised anomaly detection in network intrusion detection using clusters, in: Proceedings of the Twenty-eighth Australasian Conference on Computer Science, Australian Computer Society, Inc., 2005, pp. 333–342.
- [24] A. Pawling, N. V. Chawla, G. Madey, Anomaly detection in a mobile communication network, *Computational and Mathematical Organization Theory* 13 (4) (2007) 407–422.
- [25] K.-L. Li, H.-K. Huang, S.-F. Tian, W. Xu, Improving one-class svm for anomaly detection, in: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, IEEE, 2003, pp. 3077–3081.
- [26] Y. Wang, J. Wong, A. Miner, Anomaly intrusion detection using one class svm, in: Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, IEEE, 2004, pp. 358–364.
- [27] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N. V. Chawla, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 1409–1416.
- [28] P. Lv, Y. Yu, Y. Fan, X. Tang, X. Tong, Layer-constrained variational autoencoding kernel density estimation model for anomaly detection, *Knowl.-Based Syst.* 196 (2020) 105753.
- [29] V. Hautamaki, I. Karkkainen, P. Franti, Outlier detection using k-nearest neighbour graph, in: Proceedings of the 17th International Conference on Pattern Recognition, IEEE, 2004, pp. 430–433.
- [30] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection, arXiv preprint arXiv:1607.00148.
- [31] S. Chauhan, L. Vig, Anomaly detection in eeg time signals via deep long short-term memory networks, in: Proceedings of 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2015, pp. 1–7.
- [32] T.-Y. Kim, S.-B. Cho, Web traffic anomaly detection using c-lstm neural networks, *Expert Syst. Appl.* 106 (2018) 66–76.
- [33] D. Barbará, C. Domeniconi, J. P. Rogers, Detecting outliers using transduction and statistical testing, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2006, pp. 55–64.
- [34] L. Fan, L. Xiong, Differentially private anomaly detection with a case study on epidemic outbreak detection, in: Proceedings of IEEE 13th International Conference on Data Mining Workshops, IEEE, 2013, pp. 833–840.
- [35] J. P. Rogers, D. Barbará, C. Domeniconi, Detecting spatio-temporal outliers with kernels and statistical testing, in: Proceedings of the 17th International Conference on Geoinformatics, IEEE, 2009, pp. 1–6.
- [36] M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction, in: Proceedings of the 2nd Workshop on Machine Learning for Sensory Data Analysis, ACM, 2014, pp. 1–8.
- [37] H. Nguyen, W. Liu, P. Rivera, F. Chen, Trafficwatch: Real-time traffic incident detection and monitoring using social media, in: Proceedings of the 20th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2016, pp. 540–551.
- [38] J. A. Barria, S. Thajchayapong, Detection and classification

- of traffic anomalies using microscopic traffic variables, *IEEE Trans. Intell. Transp. Syst.* 12 (3) (2011) 695–704.
- 885 [39] Y. Li, W. Liu, Q. Huang, Traffic anomaly detection based on image descriptor in videos, *Multimedia Tools Appl.* 75 (5) (2016) 2487–2505.
- [40] J. Zhao, Z. Yi, S. Pan, Y. Zhao, Z. Zhao, F. Su, B. Zhuang, Un-supervised traffic anomaly detection using trajectories, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 133–140.
- 890 [41] M. Riveiro, M. Lebram, M. Elmer, Anomaly detection for road traffic: A visual analytics framework, *IEEE Trans. Intell. Transp. Syst.* 18 (8) (2017) 2260–2270.
- 895 [42] T. Feng, H. J. Timmermans, Transportation mode recognition using gps and accelerometer data, *Transp. Res. Part C: Emerg. Technol.* 37 (2013) 118–130.
- [43] Y. Jia, J. Wu, M. Xu, Traffic flow prediction with rainfall impact using a deep learning method, *J. Advanced Transp.* 2017 (2017) 1–11.
- 900 [44] H. Liu, T. Taniguchi, Y. Tanaka, K. Takenaka, T. Bando, Visualization of driving behavior based on hidden feature extraction by using deep learning, *IEEE Trans. Intell. Transp. Syst.* 18 (9) (2017) 2477–2489.
- 905 [45] C.-F. Tsai, C.-Y. Lin, A triangle area based nearest neighbors approach to intrusion detection, *Pattern Recognit.* 43 (1) (2010) 222–229.
- [46] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, *ACM Comput. Surv. (CSUR)* 41 (3) (2009) 15:1–15:58.
- 910 [47] R. J. Campello, D. Moulavi, J. Sander, Density-based clustering based on hierarchical density estimates, in: *Proceedings of 17th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2013, pp. 160–172.
- [48] H.-T. Lin, C.-J. Lin, R. C. Weng, A note on platt’s probabilistic outputs for support vector machines, *Mach. Learn.* 68 (3) (2007) 267–276.
- 915 [49] F. Provost, Machine learning from imbalanced data sets 101, in: *Proceedings of the AAAI’2000 Workshop Imbalanced Data Sets*, AAAI Press, 2000, pp. 1–3.
- 920 [50] T. Zhu, Y. Lin, Y. Liu, Improving interpolation-based oversampling for imbalanced data learning, *Knowl.-Based Syst.* 187 (2020) 104826.