Detection of Anomalous Traffic Patterns and Insight Analysis from Bus Trajectory Data

Xiaocai Zhang[1][0000−0002−3783−6560], Xuan Zhang[1], Sunny Verma[1], Yuansheng Liu[1], Michael Blumenstein[2], and Jinyan Li (✉)[1][0000−0003−1833−7413]

[1] Advanced Analytics Institute, FEIT, University of Technology Sydney, Sydney, Australia
Xiaocai.Zhang@student.uts.edu.au, Jinyan.Li@uts.edu.au

[2] Centre for Artificial Intelligence, School of Computer Science, FEIT, University of Technology Sydney, Sydney, Australia
Michael.Blumenstein@uts.edu.au

Abstract. Detection of anomalous patterns from traffic data is closely related to analysis of traffic accidents, fault detection, flow management, and new infrastructure planning. Existing methods on traffic anomaly detection are modelled on taxi trajectory data and have shortcoming that the data may lose much information about actual road traffic situation, as taxi drivers can select optimal route for themselves to avoid traffic anomalies. We employ bus trajectory data as it reflects real traffic conditions on the road to detect city-wide anomalous traffic patterns and to provide broader range of insights into these anomalies. Taking these considerations, we first propose a feature visualization method by mapping extracted 3-dimensional hidden features to red-green-blue (RGB) color space with a deep sparse autoencoder (DSAE). A color trajectory (CT) is produced by encoding a trajectory with RGB colors. Then, a novel algorithm is devised to detect spatio-temporal outliers with spatial and temporal properties extracted from the CT. We also integrate the CT with the geographic information system (GIS) map to obtain insights for understanding the traffic anomaly locations, and more importantly the road influence affected by the corresponding anomalies. Our proposed method was tested on three real-world bus trajectory data sets to demonstrate the excellent performance of high detection rates and low false alarm rates.

Keywords: Traffic · anomalous pattern · bus trajectory · deep sparse autoencoder.

1 Introduction

Detection of anomalous traffic patterns is to find out the traffic patterns which are not expected but which are helpful for traffic management [10]. Anomalous patterns in moving transportation carriers' trajectories can reflect abnormal traffic streams on the transportation networks [13]. These patterns can be caused by various factors such as traffic accidents, traffic controls, parades, sports events, disasters or other events. To our knowledge, existing trajectory-based traffic anomalies/outliers detection methods are based on taxi trajectory data [3, 7, 13, 16, 19]. However, many accessible trajectory data sources of the bus have not been investigated for traffic anomalies probing. Bus service operates along almost all main roads in metropolitan cities everyday, which facilitates commuters substantially. Moreover, GPS has equipped bus with high-resolution positioning information, which probes the city-wide traffic situations factually.

Models based on bus trajectory data have the following advantages: (*i*) As public transportation carrier, there is not much risk of privacy leaking; (*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, bus trajectory is more independent of the drivers' preference, which can probe the real road traffic conditions. On the contrary, taxi trajectory data may lose much information about traffic situation, since taxi drivers can choose paths for themselves [6, 20]. If a taxi driver gets the traffic information ahead, the driver very likely chooses the optimal route to avoid a foreseeable congestion, which may not get useful information regarding anomalous traffic pattern discovery.

Our **contributions** in this research are summarized as follows:

1. We present a deep neural network architecture to extract deep hidden features and devise a novel algorithm for anomalous traffic patterns detection.
2. We visualize features by mapping them into red-green-blue (RGB) color space and conduct GIS map fusion for getting insights into anomalies regarding of uncovering anomaly locations as well as impacts to road traffic.
3. We perform comprehensive experiments on three real-world data sets to confirm the effectiveness and superiority of the deep neural network architecture on feature extraction, our proposed anomaly detection algorithm and insight of anomaly.

2 Related Work

Anomalous pattern detection from spatio-temporal data is a popular topic in the domain of data mining and knowledge discovery [13]. For anomaly/outlier detection, principal component analysis (PCA) based methods [8], random projection [5], clustering based method [14] and one-class SVM [9] have been widely adopted.

The study of road traffic anomaly detection has also been investigated in many studies, most of which are based on city-wide taxi trajectories. Pang *et al.* [16] have applied likelihood ratio tests which have been commonly used in epidemiological studies to represent traffic patterns. Experiments on real taxi GPS data show the accurate and rapid detection of traffic anomalies. Liu *et al.* [13] constructed anomaly detection model by building a region graph from taxi GPS data, where a node represents a region and the link between every two nodes denotes the traffic flow, and the extreme outliers can be detected from graph links. Chawla *et al.* [3] used PCA to identify traffic anomaly form GPS trajectory data based on their deviation from their respective historical traffic profile. Zhang *et al.* [19] proposed an isolation based anomalous taxi trajectory detection method and get high detection rate. Kuang *et al.* [7] employed wavelet transform and PCA to uncover anomalous traffic events in urban areas using taxi GPS data.

Apart from using taxi trajectory data, there are also a few papers utilizing other kinds of data sources. Nguyen *et al.* [15] developed a real-time system using social media (Twitter) data for traffic incident detection. Li *et al.* [11] introduced a traffic anomaly detection algorithm based on the massive traffic video. Riveiro *et al.* [17] constructed a visual analytics framework that employs large amounts of multidimensional and heterogeneous road traffic data for traffic anomaly detection.

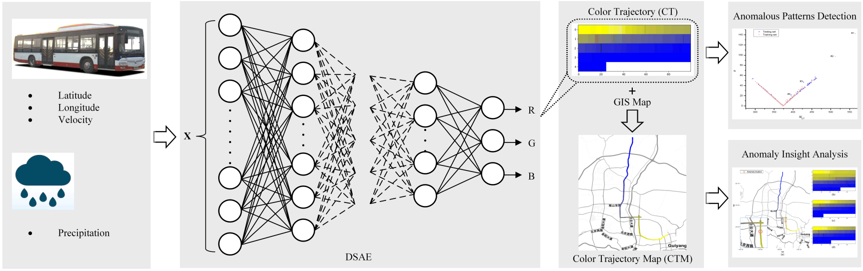


Fig. 1: The workflow of anomalous traffic pattern detection and insight analysis.

3 Proposed Method

Fig. 1 illustrates the overall workflow of our proposed method. First, feature extraction is conducted by a deep learning method; and a color trajectory (CT) is established and visualized for traffic anomalous insight analysis. Then, anomalous patterns are uncovered by extracting spatial and temporal properties and devised anomaly detection algorithm.

3.1 Feature Extraction and Visualization Using Deep Learning

We employ deep sparse autoencoder (DSAE) to extract hidden features, which can also be used for bus trajectory visualization. A GPS trajectory trip **T** is defined as

$$\mathbf{T} = \left((\varphi_1, \lambda_1, v_1)^T, \cdots, (\varphi_N, \lambda_N, v_N)^T\right) \in \mathbb{R}^{D\times N} \tag{1}$$

where φ, λ and v denote latitude and longitude and velocity (unit: km/h), N denotes the number of time series and D denotes the dimensionality. In this study, we have $D = 3$.

We also consider the impact of rainfall on traffic conditions, as the consensus that heavy rainfall can significantly affect traffic flow characteristics and lead to traffic congestion or even accidents. Therefore, we integrate trajectory data with precipitation data r_i (unit: mm/h) to get

$$\mathbf{Z} = \left(\left(\varphi_1, \lambda_1, v_1, r_1\right)^T, \cdots, \left(\varphi_N, \lambda_N, v_N, r_N\right)^T\right) \in \mathbb{R}^{(D+1)\times N} \tag{2}$$

Before feeding the data to DSAE, data normalization and windowing operation are conducted. We set a time window size ω to move z_i by one step along the time axis, therefore, we have the windowed data as the network input **X**

$$x_i = (\varphi_i, \lambda_i, v_i, r_i, \cdots, \varphi_{i+\omega-1}, \lambda_{i+\omega-1}, v_{i+\omega-1}, r_{i+\omega-1})^T \in \mathbb{R}^{4*\omega} \tag{3}$$

$$\mathbf{X} = \left(x_1, x_2, \cdots, x_{N_X-1}, x_{N_X}\right) \in \mathbb{R}^{(4*\omega)\times N_X} \tag{4}$$

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

We feed $\mathbf{X}$ into the DSAE, which is a deep neural network stacked by many sparse autoencoder (SAE). Each single SAE is lay-wise pre-trained before fine-tuning of the whole network. Suppose the visible layer's vector in the lth SAE denotes $\mathbf{v}^{(l)} \in \mathbb{R}^{D_V \times N_X}$, then we can get the hidden layer's vector ($\mathbf{h}^{(l)}$) and reconstruction vector ($\mathbf{r}^{(l)}$).

$$\mathbf{h}^{(l)} = \tanh(\mathbf{W}_{en}^{(l)} \cdot \mathbf{v}^{(l)} + \mathbf{b}_{en}^{(l)}) \in \mathbb{R}^{D_H^{(l)} \times N_X} \tag{5}$$

$$\mathbf{r}^{(l)} = \tanh(\mathbf{W}_{de}^{(l)} \cdot \mathbf{h}^{(l)} + \mathbf{b}_{de}^{(l)}) \in \mathbb{R}^{D_R^{(l)} \times N_X} \tag{6}$$

where $\mathbf{W}_{en}^{(l)}$ and $\mathbf{W}_{de}^{(l)}$ are the weights of lth layer of encoder and decoder. $\mathbf{b}_{en}^{(l)}$ and $\mathbf{b}_{de}^{(l)}$ are the biases of lth layer of encoder and decoder. The reconstruction error is

$$\min L^{(l)} = \tfrac{1}{2} \left\|\mathbf{h}^{(l)} - \mathbf{r}^{(l)}\right\|_2^2 + \alpha \left(\left\|\mathbf{W}_{en}^{(l)}\right\|_2^2 + \left\|\mathbf{W}_{de}^{(l)}\right\|_2^2 \right) + \beta \textstyle\sum_{j=1}^{D_H^{(l)}} \mathrm{KL}\left(\rho || \hat{\rho_j^{(l)}}\right) \tag{7}$$

$$\mathrm{KL}\left(\rho || \hat{h}^{(l)}\right) = \rho \log \frac{\rho}{\hat{\rho_j^{(l)}}} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho_j^{(l)}}} \tag{8}$$

where α, β and ρ are the preset parameters. $\hat{\rho_j^{(l)}}$ is the average activation of units in the lth hidden layer.

We embed a 3-neuron layer as the output layer of DSAE to get 3-dimensional hidden features, corresponding to red, green and blue channel in RGB color space.

$$\mathbf{Y} = (o_r, o_g, o_b)^T \in \mathbb{R}^{3 \times N_Y} \tag{9}$$

where $N_Y = N - \omega + 1$. Then we normalize the red channel o_r into range $[0, 255]$.

$$R = \mathrm{Round}\left(\frac{o_r - \min(o_r)}{\max(o_r) - \min(o_r)} \times 255 \right) \tag{10}$$

In a similar way, we can also get the green channel (G) and blue channel (B).

$$\mathbf{CT} = (R, G, B)^T \in \mathbb{R}^{3 \times N_{CT}} \tag{11}$$

where $N_{CT} = N - \omega + 1$, and we call $\mathbf{CT}$ the *color trajectory* (CT) forementioned.

3.2 Anomalous Patterns Detection

For the ith complete trajectory, we have τ_i called *trajectory representation*.

$$\tau_i = \left(N_{CT_i}, \mathbf{CT}_i\right) = \left(N_{CT_i}, \left(R_i, G_i, B_i\right)^T\right) \tag{12}$$

where N_{CT_i} is a temporal feature that reflects the trajectory duration. A longer duration indicates a higher confidence that traffic anomaly might have occurred. However, if a trajectory duration is located in a normal zone, there also might be traffic anomaly happened.

We choose a trajectory representation τ_k as *exemplar*, which we recommend to choose with small N_{CT}, as it is more impossible to be anomaly. We define $s\left(\tau_i, \tau_k\right)$ as the similarity between $\mathbf{CT}_i$ and $\mathbf{CT}_k$ of exemplar. To calculate the similarity, there is precondition that $N_{CT_i} = N_{CT_k}$. If $N_{CT_k} < N_{CT_i}$, we append $N_{CT_i} - N_{CT_k}$ points of white color (rgb(255, 255, 255)) to $\mathbf{CT}_k$. In that way, we construct a new trajectory representation τ_j to make $N_{CT_i} = N_{CT_j}$, while the temporal feature N_{CT_k} keeps the same.

$$\tau_j = \left(N_{CT_k}, \mathbf{CT}_j\right) = \left(N_{CT_k}, \left(R_j, G_j, B_j\right)^T\right) \tag{13}$$

Similarly, if $N_{CT_k} > N_{CT_i}$, we do the same processing on $\mathbf{CT}_i$. Then, the similarity between $\mathbf{CT}_i$ and $\mathbf{CT}_k$ can be derived by Eq. (14).

$$s\left(\tau_i, \tau_k\right) = \begin{cases} \sum_{n=1}^{N_{CT_i}} \left(\frac{\left(R_i^n - R_k^n\right)^2 + \left(G_i^n - G_k^n\right)^2 + \left(B_i^n - B_k^n\right)^2}{255^2 + 255^2 + 255^2} \right) & \text{if} N_{CT_i} = N_{CT_k} \\ \sum_{n=1}^{N_{CT_i}} \left(\frac{\left(R_i^n - R_j^n\right)^2 + \left(G_i^n - G_j^n\right)^2 + \left(B_i^n - B_j^n\right)^2}{255^2 + 255^2 + 255^2} \right) & \text{if} N_{CT_i} > N_{CT_k} \\ \sum_{n=1}^{N_{CT_k}} \left(\frac{\left(R_k^n - R_m^n\right)^2 + \left(G_k^n - G_m^n\right)^2 + \left(B_k^n - B_m^n\right)^2}{255^2 + 255^2 + 255^2} \right) & \text{if} N_{CT_i} < N_{CT_k} \end{cases} \tag{14}$$

Suppose $d_{ab}^n = \frac{\left(R_a^n - R_b^n\right)^2 + \left(G_a^n - G_b^n\right)^2 + \left(B_a^n - B_b^n\right)^2}{255^2 + 255^2 + 255^2}$, we introduce a small positive threshold ε, if the similarity between two color points is smaller than ε, we regard that they are the same and the similarity equals 0. Therefore, we have Eq. (15) in Eq. (14).

$$d_{ab}^n = \begin{cases} d_{ab}^n & \text{if } d_{ab}^n \geq \varepsilon \\ 0 & \text{if } d_{ab}^n < \varepsilon \end{cases} \tag{15}$$

For the *i*th complete trajectory, we have

$$\epsilon_i = \left(N_{CT_i}, s\left(\tau_i, \tau_k\right)\right) \tag{16}$$

where $s\left(\tau_i, \tau_k\right)$ is a spatial property which reflects the spatial distribution of the moving object. By mapping all ϵ to a two-dimensional space which we refer to as *spatio-temporal plane* here, we could categorize anomalous trajectories into two categories: *class 1 anomaly* and *class 2 anomaly*. They are defined as follows:

Class 1 anomaly: An anomalous trajectory whose associated spatial and temporal feature values are both different from both spatial and temporal feature values of its spatio-temporal neighbors.

Class 2 anomaly: An anomalous trajectory whose associated spatial feature value is very different from the spatial feature values of its temporal neighbors.

Spatio-temporal outliers points can be detected with devised anomalous traffic patterns detection (**ATPD**) algorithm (Algorithm 1), where steps 4 to 13 divide the whole observations into different candidature sets for different detection tasks (class 1 and class 2 anomaly detection) by adopting a threshold N_C. For class 2 anomaly detection, we employ Boxplot rule with a threshold δ to uncover anomalous observations (steps 14 to 26). Simultaneously, class 1 anomaly could be detected by computing Euclidean distance from the nearest neighbor and employing a threshold r (steps 27 to 34).

Algorithm 1: ATPD algorithm

```
Parameters: $N_C, \delta, r, K$;
Input: $\epsilon$;
Output: $C$;
$m \leftarrow 0, n \leftarrow 0$;
for $\epsilon_i \in \epsilon$ do
    if $N_{CT_i} \geq N_C$ then
        $m \leftarrow m + 1$;
        $\epsilon_{C1}(m) \leftarrow N_{CT_i}$;
    else
        $n \leftarrow n + 1$;
        $\epsilon_{C2}(n) \leftarrow N_{CT_i}$;
    end if
end for
for $\epsilon_{C2}(n) \in \epsilon_{C2}$ do
    $TN \leftarrow$ Temporal $K$ neighbors selection $(N_{CT_n}, K)$;
    $S \leftarrow$ Get corresponding similarity set $(TN, \epsilon)$;
    $Q_1, Q_3 \leftarrow$ Compute the first and third quartile $(S)$;
    $IQR \leftarrow Q_3 - Q_1$;
    $U \leftarrow Q_3 + \delta * IQR$;
    $L \leftarrow Q_1 - \delta * IQR$;
    if $s_{C2}(\tau_n, \tau_k) > U$ or $s_{C2}(\tau_n, \tau_k) < L$ then
        $C_{C2}(n) \leftarrow$ True;
    else
        $C_{C2}(n) \leftarrow$ False;
    end if
end for
for $\epsilon_{C1}(m) \in \epsilon_{C1}$ do
    $D \leftarrow$ Compute distance to the nearest spatio-temporal neighbor $(\epsilon_{C1}(m))$;
    if $D > r$ then
        $C_{C1}(m) \leftarrow$ True;
    else
        $C_{C1}(m) \leftarrow$ False;
    end if
end for
$C \leftarrow$ Combine $(C_{C1}, C_{C2})$;
```

3.3 Anomaly Insight Analysis

We combine Eq. (1) and Eq. (11) together to construct *color trajectory map* (CTM) by integrating **CT** with GIS map. Note that $N_{CT} < N$ as the windowing processing introduced. We construct *location vector* l_i and *location matrix* **L**.

$$l_i = (\varphi_i, \lambda_i)^T \in \mathbb{R}^2 \tag{17}$$

$$\mathbf{L} = \left(l_{\left|\frac{w-1}{2}\right|+1}, l_{\left|\frac{w-1}{2}\right|+2}, \cdots, l_{\left|\frac{w-1}{2}\right|+N-w}, l_{\left|\frac{w-1}{2}\right|+N-w+1} \right) \in \mathbb{R}^{2\times N_L} \tag{18}$$

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

We combine the location matrix **L** and **CT** together to get

$$\mathbf{L}' = (\mathbf{L}, \mathbf{CT}) \tag{19}$$

For each L_i', map color with value of $\left(R_i, G_i, B_i\right)^T$ to coordinate $(\varphi_i, \lambda_i)^T$ on the GIS map. So that the CTM of a whole trajectory has been generated.

$$L_i' = \left((\varphi_i, \lambda_i)^T, \left(R_i, G_i, B_i\right)^T\right) \tag{20}$$

Color trajectory (**CT**) and CTM are linked together via rgb value ($(R_i, G_i, B_i)^T$). By comparing the **CT** of an anomalous trajectory and non-anomalous trajectories, the anomalous color with significant difference could be found intuitively. Then, the anomaly happened location (road section) could also be discovered on CTM intuitively.

4 Experiments and Analysis

In this section, we conduct extensive experiments for answering the following questions:

Q1: Is **ATPD** effective while detecting all anomalies i.e. high detection rate?

Q2: Does our developed feature visualization method useful for capturing anomaly locations and traffic impacts with detected anomalies using **ATPD**?

Q3: How does our proposed **ATPD** for feature extraction and anomaly detection performs in comparison to the state-of-the-art anomaly detection methods?

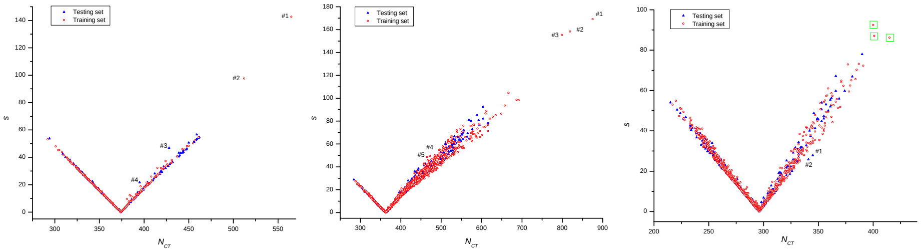
4.1 Experimental Settings

Data Sets: We use the GPS trajectory data of bus Route 66, Route 50 and Route 18 in Guiyang, China. All the data (including the local hourly precipitation data) is available at Guiyang Open Government Data Platform [1]. All the data are divided into training and test set: three months for training, and the following one month for testing. Table 1 gives detail description of data sets.

Table 1: Data sets description

Route No.	Day Type	Whole Sample	Training Sample	Test Sample	Training Set Period	Test Set Period	Input Dimension
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

Parameters: The parameters are set as: $(\omega, \alpha, \beta, \rho, \varepsilon) = (10, 10^{-5}, 10^{-4}, 0.05, 0.01)$ for all the data sets, as suggested by the literature work [12]. The algorithm's parameters are set as: $(N_C, \delta, r, K) = (450, 2.2, 50, 2)$ for Route 66, $(N_C, \delta, r, K) = (500, 1.7, 50, 2)$ for Route 50 and $(N_C, \delta, r, K) = (350, 0.9, 40, 2)$ for Route 18, with the understanding and trials from training set. Besides, we employ a DSAE with four encoding layers with the dimensions of $40 \rightarrow 20 \rightarrow 10 \rightarrow 3$ to get the three-dimensional hidden features.



(a) Route 66 (b) Route 50 (c) Route 18

Fig. 2: Spatio-temporal planes of Route 66, 50 and 18. Objects inside □ are not with high confidence to be regarded as anomalies, as their spatial and temporal features are not far away from their spatio-temporal neighbors.

4.2 Result from Anomalous Patterns Detection (Answering Q1)

To evaluate this criteria we utilize the detection rate (precision) calculated as $TP/(TP+FN)$ and false alarm rate calculated as $FP/(FP+TN)$ [18]. The performance comparisons of our proposed **ATPD** versus popular baselines are shown in Table 3, note that we transfer the anomalous observations in training set to test set in order to enlarge the anomalous sample size for performance testing. The proposed **ATPD** detects all anomalies with low false alarm rate. Visualizations of spatio-temporal planes for Route 66, 50 and 18 are shown in Fig. 2, where points are distributed along the tick (✓) with the pattern that similarity s increases with N_{CT} when $N_{CT} > N_{CT_k}$, while it decreases with N_{CT} when $N_{CT} < N_{CT_k}$. Moreover, at first glance, we can discern those anomalous objects that are not located in the expected zones from the spatio-temporal planes shown in Fig. 2. Also, in Fig. 2 (a), anomalies #1 and #2 are referred as *class 1 anomaly* as their spatial and temporal features are both far away from their spatio-temporal neighbors, and so does anomalies #1, #2, and #3 in Fig. 2 (b). However, we regard anomaly #3, #4 in Fig. 2 (a), #4, #5 in Fig. 2 (b) and #1, #2 in Fig. 2 (c) as *class 2 anomaly*, as only their spatial features are far away from their temporal neighbors. In general, class 1 anomaly has more serious impact on traffic than class 2 anomaly, while class 2 anomaly is more difficult to uncover.

The detected outliers by using our model are all coincide with the known traffic anomalous events (Table 2). They are listed in the following:

Known event 1: A sedan bumped another car at the Shachong East Road in the late afternoon of 14 August 2016, the driver of the sedan ran away after accident resulting in severe traffic congestion. Note that it was raining during this time period and the event only affected services for Route 50.

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

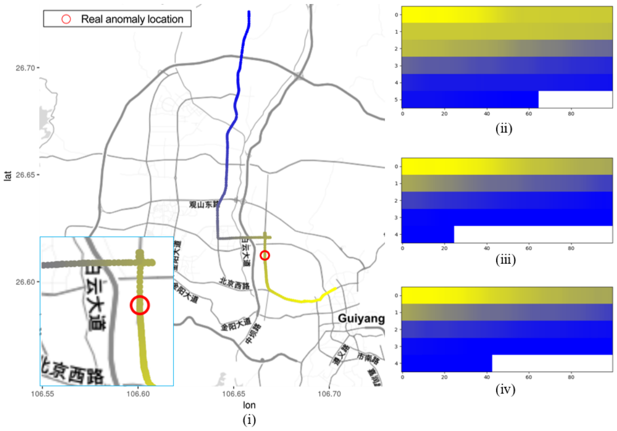
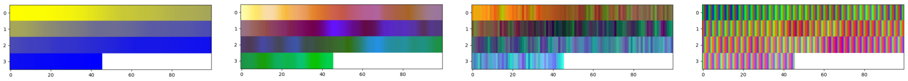


Fig. 3: Insight analysis example of anomaly #1 in Route 66. (i) CTM of anomaly #1. (ii) CT of anomaly #1. (iii) CT of an non-anomaly. (iv) CT of another non-anomaly.

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 died. This event only affected Route 66 service.

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



(a) DSAE (b) PCA (c) RP (d) SAE

Fig. 4: A color trajectory in Route 66 generated by methods of DSAE, PCA, RP and SAE. The CT generated by our DSAE based method is the most smooth and distinct.

4.3 Result from Visualization and Anomaly Insight Analysis (Answering Q2)

Fig. 4 (a) gives an example of the CT from a real-world bus trajectory. For Fig. 4 (a), the bus trajectory starts at color of yellow 🟨, with the bus proceeding to the destination, the color changes gradually and finally terminates at blue 🟦. The horizontal axis indicates the trajectory duration (N_{CT}, 1 unit equals 10 seconds, each row contains 100 units).

After detecting anomalous trajectory we can get the CTM of these trajectories by fusion of the GIS map (Eq. 20). We illustrate an anomalous trajectory by taking the CTM

Table 2: Detected anomalies

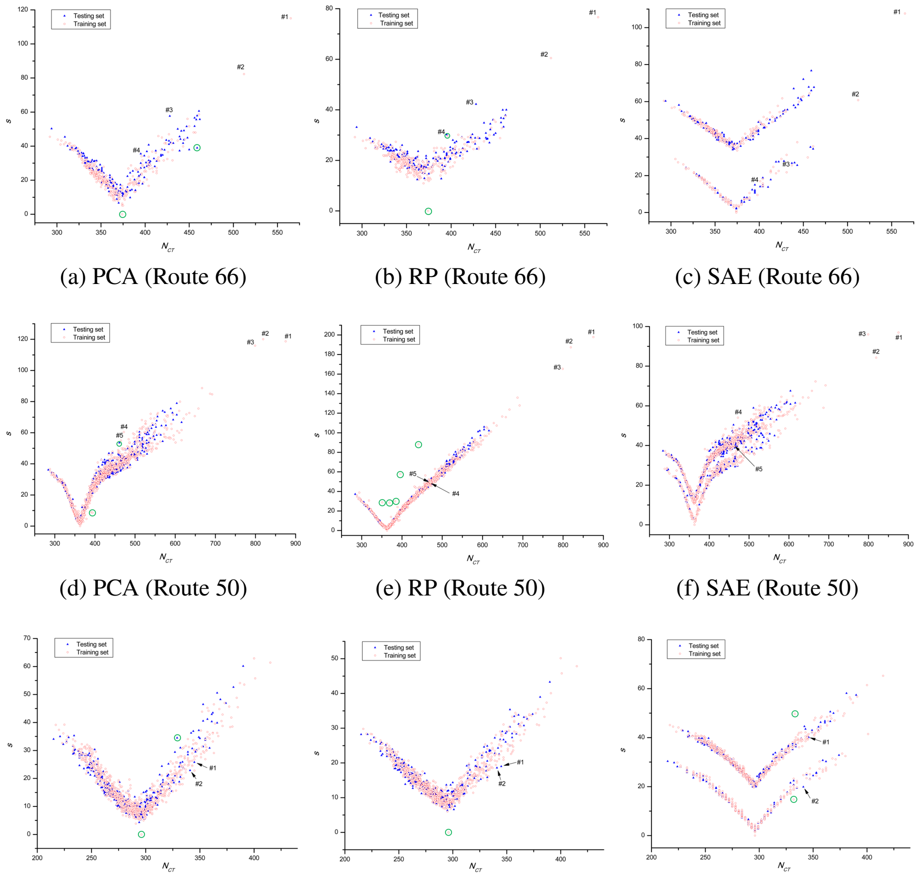
Route	Anomaly	Date	Running Time	Event	Category
66	#1	18 Sept. 2016	07:29-09:00	Event 2	Class 1 anomaly
	#2	18 Sept. 2016	07:00-08:23	Event 2	Class 1 anomaly
	#3	26 Nov. 2016	12:43-13:55	Event 3	Class 2 anomaly
	#4	26 Nov. 2016	12:07-13:09	Event 3	Class 2 anomaly
50	#1	18 Sept. 2016	06:58-09:25	Event 2	Class 1 anomaly
	#2	18 Sept. 2016	07:22-09:40	Event 2	Class 1 anomaly
	#3	18 Sept. 2016	07:34-09:48	Event 2	Class 1 anomaly
	#4	14 Aug. 2016	19:41-21:01	Event 1	Class 2 anomaly
	#5	14 Aug. 2016	17:32-18:51	Event 1	Class 2 anomaly
18	#1	14 Dec. 2016	09:31-10:28	Event 4	Class 2 anomaly
	#2	14 Dec. 2016	09:50-10:48	Event 4	Class 2 anomaly

of anomaly #1 for Route 66 as example as displayed in Fig. 3, where ○ in (i) denotes the actual event site. In contrast the CT of anomaly #1 (Fig. 3 (ii)) and non-anomalies (Fig. 3 (iii) and (iv)) we could have an intuitive perspective that the anomaly might have occurred around light yellow 🟨 color. Since, the trajectory part with such color is quite different from those of non-anomalies. However, when it proceeds to color of grey ■, the rest partial colors turn to be similar to those of non-anomalies. Therefore, we could infer that the anomaly indeed happened at the locations highlighted between locations 🟨 and ■ in Fig. 3, which is in line with the real location (○) of event 2. Apart from location detection, we also provide influential insights to this car crash by highlighting road between 🟨 and ■ (left bottom of Fig. 3).

4.4 Baseline and Comparison (Answering Q3)

Feature Extraction and Visualization: We compare our model with other popular baselines in dimensionality reduction: PCA, random projection (RP) and sparse autoencoder (SAE). In contrast the CTs generated by DSAE (Fig. 4 (a)) and baselines ((Fig. 4) (b) (c) (d))), we can find that DSAE gets a more smooth and distinct color distribution. In Fig. 5, by comparing the spatio-temporal planes derived by above baselines, we can find that none of these baselines can get favorable detection performance than the DSAE-based model on all data sets. Some anomalies (especially for class 2 anomaly) are mixed together with non-anomalies (#3, #4 in Route 66 with PCA and SAE, #4, #5 in Route 50 with RP and SAE, #1, #2 in Route 18 with PCA, RP and SAE), which is quite difficult to detect. Moreover, many non-anomalies can be mis-detected as outliers with high confidence (labeled ○ in Fig. 5).

We define an evaluation index named averaged moving standard deviation (AMSD) to evaluate the concentration of tick shape (✓) distributed data. We employ a window size (κ) for windowing operation along N_{CT} (horizontal axis), compute the sample standard deviation of all normalized $s\left(\tau_{ij}, \tau_k\right)$ (denoted as $\hat{s}\left(\tau_{ij}, \tau_k\right)$) within each κ-sized N_{CT}. Following this, we get the mean standard deviation of all κ-sized N_{CT} for AMSD. We remove those anomalous trajectories from training sets, and calculate the AMSD value of all the non-anomalies derived from DSAE-based model and other baselines. In



(a) PCA (Route 66) (b) RP (Route 66) (c) SAE (Route 66)

(d) PCA (Route 50) (e) RP (Route 50) (f) SAE (Route 50)

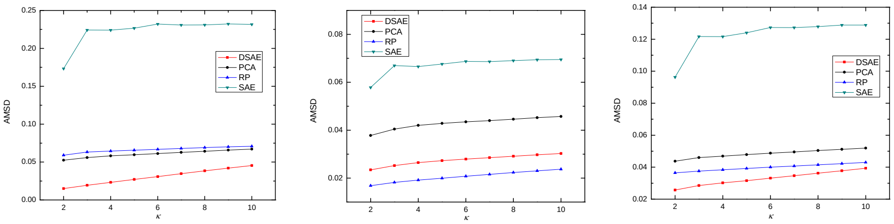
(g) PCA (Route 18) (h) RP (Route 18) (i) SAE (Route 18)

Fig. 5: Spatio-temporal planes from baselines of PCA, RP and SAE.

Fig. 6, our model performs best in data sets of Route 66 and Route 18. RP can get fairly good performance in Route 50, however, its performance in anomaly detection is poorly as it mis-detected quite a number of points in Route 50.

$$AMSD = \frac{1}{m}\sum_{i=1}^{m}\sqrt{\frac{1}{n_i - 1}\sum_{j=1}^{n_i}\left(\hat{s}\left(\tau_{ij}, \tau_k\right) - \bar{s}_i\right)} \qquad (21)$$

Anomaly Detection: We also compare our model (**ATPD**) with other baselines that employed in outlier/anomaly detection: classification based (one-class SVM, long short-term memory (LSTM) networks), clustering based (affinity propagation (AP) clustering) and nearest-neighbor based (kNN) [2]. We also feed the extracted features via DSAE into those baselines. The performances are given in Table 3. Since there is no anomalous



(a) Route 66 (b) Route 50 (c) Route 18

Fig. 6: Concentration performance evaluation in training sets.

Table 3: Comparison of performance in test set with baselines

Route	Detection Rate					False Alarm Rate				
	ATPD	SVM	LSTM	AP	kNN	**ATPD**	SVM	LSTM	AP	kNN
66	100%	0	25.00%	50.00%	50.00%	0.63%	38.12%	0	12.50%	0
50	100%	0	20.00%	60.00%	60.00%	1.69%	51.98%	0	0	0
18	100%	—	—	0	—	3.09%	—	—	0.34%	—

trajectory in Rout 18 training set, supervised learning of SVM, LSTM and kNN cannot be conducted. Overall, **ATPD** can get superior performance, SVM performs poorly since it might have exploited inadequacies of the soft-margin maximization paradigm when handling extremely imbalanced data [4].

5 Conclusion and Future Work

In this paper, we proposed an anomalous traffic patterns detection method for bus trajectory data analysis. We extracted spatial and temporal properties from the color trajectory derived from deep learning method, and then devised an algorithm based on Boxplot rule and nearest neighbor for anomalous patterns detection. We integrated the visualized color trajectory with GIS map to generate color trajectory map, from which we can get intuitive insights into the locations of these anomalies and also the traffic influences to road by the corresponding anomalies. Experiments on three real-world bus route data sets confirmed the effectiveness and superiority of our proposed method compared with baselines of PCA, RP, SAE, SVM, LSTM, AP and kNN. Some cities have adopted 'Bus Lane' strategy for some metropolitan roads in certain periods to improve the reliability and efficiency of bus services. In that case, our approach may not be efficient to detect some city-wide incident-based anomalies, as the situation that some incidents that affect other vehicles on the road might not affect buses. However, from the perspective of bus service operation or management, such situation does not make much sense, since those anomalies that impose little impact on bus service will not be taken into account for decision making. In the future, we plan to study the online approach based on the developed methodology for real-time traffic anomaly detection, which is essential as buses run on almost all major roads of the city area.

References

1. Guiyang open government data platform, `http://www.gyopendata.gov.cn/city/index.htm`, accessed: 2018-02-1
2. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) **41**(3), 15 (2009)
3. Chawla, S., Zheng, Y., Hu, J.: Inferring the root cause in road traffic anomalies. In: Proc. 2012 IEEE 12th Int. Conf. Data Mining. pp. 141–150 (2012)
4. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. (9), 1263–1284 (2008)
5. Juvonen, A., Hamalainen, T.: An efficient network log anomaly detection system using random projection dimensionality reduction. In: Proc. 2014 6th Int. Conf. New Technologies, Mobility and Security (NTMS). pp. 1–5. IEEE (2014)
6. Kong, X., Song, X., Xia, F., Guo, H., Wang, J., Tolba, A.: Lotad: Long-term traffic anomaly detection based on crowdsourced bus trajectory data. World Wide Web **21**(3), 825–847 (2018)
7. Kuang, W., An, S., Jiang, H.: Detecting traffic anomalies in urban areas using taxi gps data. Mathematical Problems in Eng. **2015** (2015)
8. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: ACM SIGCOMM Computer Communication Review. vol. 34, pp. 219–230. ACM (2004)
9. Li, K.L., Huang, H.K., Tian, S.F., Xu, W.: Improving one-class svm for anomaly detection. In: Proc. 2003 Int. Conf. Machine Learning Cybern. vol. 5, pp. 3077–3081. IEEE (2003)
10. Li, Y., Guo, T., Xia, R., Xie, W.: Road traffic anomaly detection based on fuzzy theory. IEEE Access **6**, 40281–40288 (2018)
11. Li, Y., Liu, W., Huang, Q.: Traffic anomaly detection based on image descriptor in videos. Multimedia Tools and Application **75**(5), 2487–2505 (2016)
12. Liu, H., Taniguchi, T., Tanaka, Y., Takenaka, K., Bando, T.: Visualization of driving behavior based on hidden feature extraction by using deep learning. IEEE Trans. Intell. Transp. Syst. **18**(9), 2477–2489 (2017)
13. Liu, W., Zheng, Y., Chawla, S., Yuan, J., Xing, X.: Discovering spatio-temporal causal interactions in traffic data streams. In: Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining. pp. 1010–1018. ACM (2011)
14. Münz, G., Li, S., Carle, G.: Traffic anomaly detection using k-means clustering. In: GI/ITG Workshop MMBnet. pp. 13–14 (2007)
15. Nguyen, H., Liu, W., Rivera, P., Chen, F.: Trafficwatch: Real-time traffic incident detection and monitoring using social media. In: Proc. 20th Pacific-Asia Conf. Knowl. Discovery and Data Mining. pp. 540–551. Springer (2016)
16. Pang, L., Chawla, S., Liu, W., Zheng, Y.: On detection of emerging anomalous traffic patterns using gps data. Data & Knowl. Eng. **87**, 357–373 (2013)
17. Riveiro, M., Lebram, M., Elmer, M.: Anomaly detection for road traffic: A visual analytics framework. IEEE Trans. Intell. Transp. Syst. **18**(8), 2260–2270 (2017)
18. Tsai, C.F., Lin, C.Y.: A triangle area based nearest neighbors approach to intrusion detection. Pattern Recognition **43**(1), 222–229 (2010)
19. Zhang, D., Li, N., Zhou, Z.H., Chen, C., Sun, L., Li, S.: ibat: detecting anomalous taxi trajectories from gps traces. In: Proc. 13th Int. Conf. Ubiquitous Computing. pp. 99–108. ACM (2011)
20. Zhang, X., Zhao, Z., Zheng, Y., Li, J.: Prediction of taxi destinations using a novel data embedding method and ensemble learning. IEEE Trans. Intell. Transp. Syst. (2019)