Received: 22 May 2021 | Accepted: 3 August 2021

DOI: 10.1049/itr2.12109

IET Intelligent Transport Systems

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

A highly efficient framework for outlier detection in urban traffic flow

Xing Wang[1,2] | Ruihao Zeng[3] | Fumin Zou[4] | Faliang Huang[5] | Biao Jin[1,2]

[1] College of Computer and Cyber Security, Fujian Normal University, Fuzhou, China

[2] Digital Fujian Institute of Big Data Security Technology, Fuzhou, China

[3] International College of Chinese Studies, Fujian Normal University, Fuzhou, China

[4] Fujian Key Laboratory of Automotive Electronic and Electrical Drive Technology, Fujian University of Technology, Fuzhou, China

[5] School of Computer and Information Engineering, Nanning Normal University, Nanning, China

Correspondence
Ruihao Zeng, International College of Chinese Studies, Fujian Normal University, Fuzhou 350117, China.
Email: xmrhzeng2000@163.com

Funding information
the Natural Science Foundation of China, Grant/Award Number: No. 61962038; the Foreign Cooperation Project of Fujian Provincial Department of Science and Technology, Grant/Award Number: No. 2020I0014; the Guangxi Bagui Teams for innovation and Research, Grant/Award Number: No. 201979; the Startup Project of Doctoral Research of Fujian Normal University

Abstract

The outliers in traffic flow represent the anomalies or emergencies in the road. The detection and research of outliers will help to reveal the mechanism of such events. Aiming at the problem of outlier detection in urban traffic flow, this paper innovatively proposes a highly efficient traffic outlier detection framework based on the study of road traffic flow patterns. The main research works are as follows: (1) data pre-processing, the road traffic flow matrix of the roads is calculated based on the collected GPS data, the non-negative matrix factorisation algorithm is chosen to reduce the dimension of the matrix. (2) Road traffic flow pattern extraction, the fuzzy C-means clustering algorithm with the Optimal k-cluster centre (K-FCM) is adopted to cluster the roads with the same road traffic flow pattern. (3) Outlier detection model training and evaluation, kernel density estimation is introduced to fit the probability density of roads traffic flow matrices which are used to train the back propagation neural network based on particle swarm optimisation to obtain the outlier detection and evaluation model, and a threshold is introduced to optimise the precision and recall of the model. The experimental results show that: the average precision and recall of the proposed method in this paper are 95.38% and 96.23%, respectively, and the average detection time is 28.4 seconds. The method has high accuracy, high efficiency and good practical significance.

1 | INTRODUCTION

Urban traffic flow analysis has been an important research direction in urban traffic system. With the rapid development of GPS technology and the wide application of in-vehicle sensors, it has become possible to use GPS data to analyse the problems in urban traffic system. Outlier detection (OD) in traffic flow series is one of the most important applications in traffic flow analysis. How to detect outliers quickly and accurately is of great significance to provide effective decision support for traffic management departments.

Outlier detection has been an extremely important topic in statistics and data mining. Outliers are usually considered to be the values that differ significantly from the rest of the observations according to Zimek et al. [1], thus raising suspicions that these values are generated due to different mechanisms. With this definition, a better accomplishment of outlier detection can both reduce the error of model fitting and inspect the capability of model to describe data in different situations. Such a definition also provides a valuable reference for the meaning of outliers in actual events.

In traffic flow data, outliers often have more practical significance, which refers to traffic flow anomalies. Most of the traffic outliers are due to unexpected traffic accidents, traffic control, abnormal weather, major events, etc. For example, Chang'an Street is generally very smooth after 11:00 p.m. every night, but

the congestion distance is more than 800 m near 12:00 p.m. one night. It is what we consider as 'traffic flow outlier'. However, it is also regarded as an outlier when the traffic flow of a certain section is different from that of the same period in the past, even if the traffic flow of that road section is smooth. For example, the roads near Dongzhimen Station in Beijing are often very congested due to the fact that Dongzhimen Station is the interchange subway station, but one day the traffic flows are smooth, which is believed to be a practical manifestation of traffic flow outlier. Djenouri et al. [2] also demonstrated the importance of outlier detection in traffic problems through a real-case study on urban traffic flow data.

According to the analyses of Bhowmick [3] and Djenouri [4] on outlier detection problem in actual traffic events, what may impact the detection accuracy is not only the fitness of the algorithm itself, but also the unique data patterns in the urban traffic problem, such as the low sampling frequency of GPS points, the complexity of urban roads (e.g. overlapping viaducts in plan), the accuracy of road matching algorithm or even the definition of outlier. In view of the possible problems, we propose a highly efficient framework for outlier detection in urban traffic flow. The Framework is shown as in Figure 1.

In Figure 1, our method consists of three parts. In the first part, for the original GPS points, after data cleaning and map matching, we calculate the road traffic flow matrix and obtain the road traffic flow pattern matrix by non-negative matrix factorisation (NMF) algorithm dimension reduction. In the second part, the optimised K-FCM algorithm is used to cluster the roads with the same traffic flow pattern to obtain the neighbour road sections. In the third part, we use kernel density estimation to fit the Rf and Nr matrix, which obtain the PRf and PNr matrix. PRf and PNr are the input layer of the back propagation neural network based on particle swarm optimisation (PSO-BP) and Os is the output layer. Then, we optimise and evaluate the outlier detection model.

The contributions of our work can be summarised below:

i. We innovatively propose an outlier detection framework. Firstly, the idea of distance and density is introduced to extract road sections with similar traffic patterns. Then the statistical methods are used to calculate the probability density of traffic flow. Finally, the neural network is adopted to train model which is used to analyse the actual traffic flow to obtain outliers. Meanwhile, we evaluate our method and the experimental results show that the proposed method has high accuracy and high robustness as well as the strong practical significance.
ii. Considering both the Euclidean distance between sample points and the actual distance between road sections, we use K-nearest neighbour (KNN) algorithm to optimise the fuzzy C-means algorithm, and the optimised algorithm (K-FCM) has better experimental effect.
iii. We introduce the PSO algorithm to optimise the gradient descent update algorithm of BP neural network to avoid falling into local optimal solution. The optimised neural network algorithm (PSO-BP) has high efficiency as well as good experimental effect.

The remainder of this paper is organised as follows: Section 2 shows the related works. Section 3 illustrates related definitions. Section 4 describes our outlier detection method clearly. Then we evaluate our method by experiments in Section 5. Section 6 concludes this paper.

2 | RELATED WORK

According to the researches of Djenouri [5] and Domingues [6] on traffic outliers, we can divide the vast majority of current research directions into three main categories. The first is to identify traffic outliers using statistical models. The second is to use distance measures and neighbourhoods for local density estimation to derive the outliers. The third is to extract the correlation between traffic flows through pattern analysis to find outliers.

2.1 | Statistical-based approaches

The statistical-based OD approaches mainly follow the following steps. After obtaining the data in n-dimensional space, the data are generally down-scaled using algorithms such as principal component analysis, after which different data clusters are obtained using statistical models, where those with the largest amount of data are considered to be clusters of inliers while those outside the clusters of inliers are considered to be outliers. For example, Ngan et al. [7] proposed an OD method based on Dirichlet process mixture model (DPMM), which solves the shortcoming of previous OD methods that cannot distinguish the spatial-temporal similarity of road sections near traffic signals by mapping traffic signals to covariance signal descriptors, reaching 96.67% resolution in cross-validation. Kingan et al. [8] used least squares regression and residual standard deviation technique for possible outliers in data processing of traffic flow prediction to identify abnormal traffic by a fixed threshold. Turochy et al. [9] proposed a method to detect traffic congestion using Multivariate Statistical Quality Control with multiple variables such as average speed and the occupancy rate. They examined the distribution of sample points using the F-distribution and calculated the scores of samples by covariance. Finally, they derived outliers by comparing the scores with thresholds. The BoostSelect algorithm proposed by Campos et al. [10] improved on the Greedy ensemble and SelectV algorithms by using combinatorial statistics for the selection of outliers. There is also an outlier detection algorithm [11] that combined statistical models with Hidden Markov Models that can effectively respond to the influence relationships between road sections.

2.2 | Density or distance-based approaches

As a common method, this type of methods mostly used distance space or local outlier factor (LOF) to label outliers. For example, Dang et al. [12] proposed KNN algorithm and density threshold to detect outliers, which can be considered as

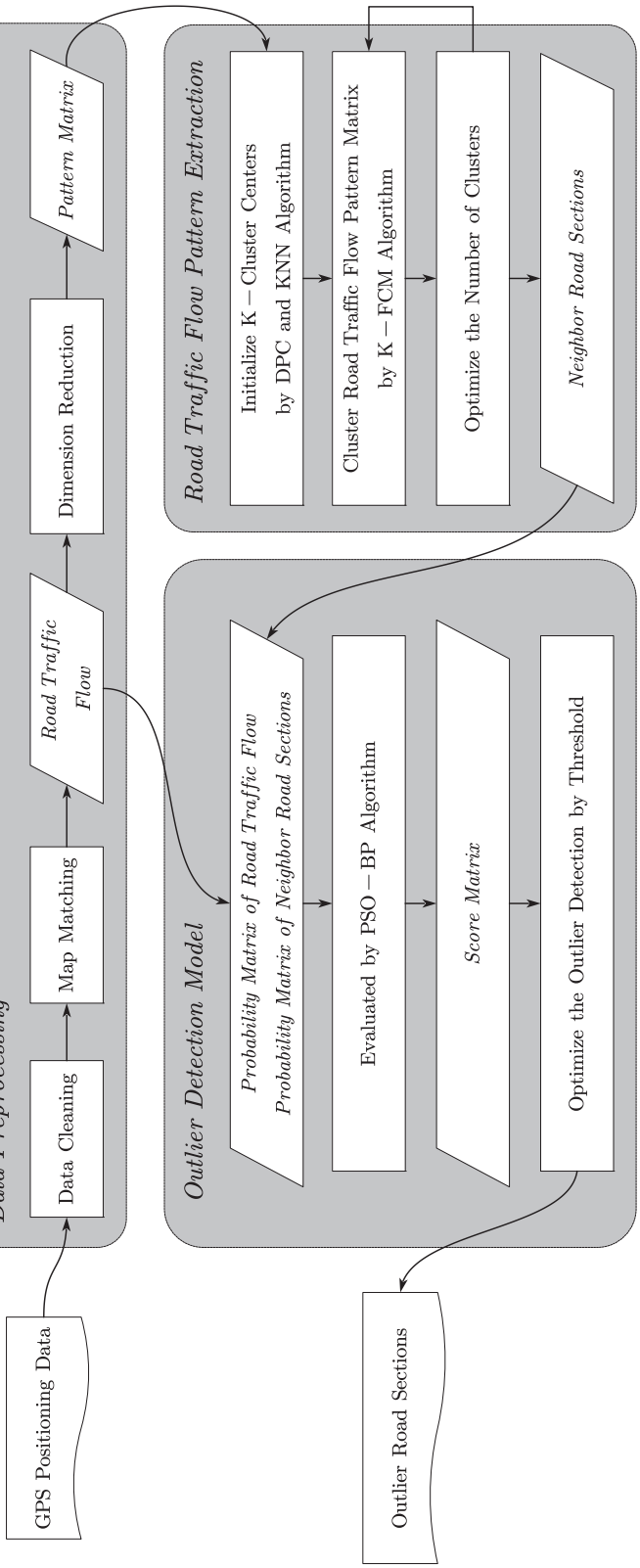


FIGURE 1 The framework of our method

outliers when the density of sample points and their neighbours is less than the specified density threshold. A similar approach was also used to detect outliers by Ramaswamy et al. [13]. This type of methods had general validity after unsupervised outlier tests by Campos et al. [14]. Although Tang et al. [15] used boundary density of LOF to process the traffic video in Hong Kong, they were eventually able to improve the recognition accuracy of OD at intersections to 96%. Munoz-Organero et al. [16] then created a sliding window based on velocity and acceleration using information such as traffic signals. Then they compared the Mahalanobis distance between the flow vector and the central flow vector of sliding window as a basis for determining whether a point is an outlier. The sliding window is also combined with the micro-clustering outlier detection algorithm by Kontaki et al. [17] to complete the screening of outliers with high accuracy. Similar density or distance-based methods have also been used by many other scholars. For example, Huang et al. [18] proposed an effective subspace dimension algorithm to learn the features of outliers by calculating the maximum subspace distance and covariance matrix. Pu et al. [19] used spatial and temporal label propagation method to detect outliers by focusing on the changes of neighbouring roads along with road distance and density. Meanwhile, for example, the efficient scalable neighbourhood detection retention method proposed by Schubert et al. [20], the approximate neighbours for density estimates used by Kirner et al. [21] and neighbourhood-approximated algorithm proposed by Duggimpudi et al. [22] can serve the purpose of detecting outliers.

2.3 | Pattern-based approaches

Considering the causes of outliers as different mechanisms also provides a detection idea for OD, that is, detecting patterns that cause outliers and comparing them with regular pattern. The pattern-based approach usually collects vehicle movement data other than GPS location data, such as vehicle speed, and traffic congestion. This approach is able to mine a regular pattern from various traffic variables in the city. Data that do not fit the regular pattern or fit the abnormal patterns can be classified as outliers. For example, Li et al. [23] compared temporal information of data set with historical similarity to detect outliers. Campos et al. [24] used events to establish node and connection relationships to obtain the characteristics of outliers by creating event linkage graphs. Sun et al. [25] effectively detected the outliers of spatiotemporal data by re-describing the traffic state with the innovative firefly algorithm-based spatiotemporal outlier detection method (IFA-STODM). Systematic methods are more prominent in detecting patterns of outliers, for example, Cao et al. [26] used the present Internet of Things to build a deTects Outlier Patterns (TOP) system with context-driven search algorithm for the efficient capture of outliers' patterns. The mutual information and generalised entropy-based feature selection technique proposed by Bhuyan et al. [27] which can select the subsets of outliers is the embodiment of information theory applied to systematic outlier detection methods. Detection for patterns requires not only various data on traffic, but driver's data can be incorporated into the computational system. For example, Blázquez et al. [28] used the driver's heart rate to determine outliers by calculating the time interval between heart rate peaks through the support vector machine.

In summary, we analyse several typical representative types of research work, as shown in Table 1.

Through the analysis of the related research work, we find that the current research on urban traffic flow outlier detection mainly has the following defects.

i. Statistical methods are not only susceptible to the degree of probability model fit, but also ignore the spatial relationship of each road (e.g. once a road is congested, its neighbouring roads are also likely to be congested).
ii. Most of the methods for examining the distance of sample points are based on Euclidean distance, which may affect the detection results and low the robustness in practical problems by considering Euclidean distance alone. Also, those methods based on distance or density alone tend to ignore the temporal factor.
iii. The method of using multidimensional information to determine traffic patterns can restore the spatial and temporal state of the road in a more realistic way. However, this method will undoubtedly require more data and computing power than other methods that only need to collect the traffic flow of the road.

Different from the above methods, we use the GPS position information of the vehicles to carry out our research work. Based on the collected GPS data, we calculate the road traffic flow pattern of the roads and then cluster the roads with the same road traffic flow pattern, train the back propagation neural network based on PSO (PSO-BP) to obtain the outlier detection and evaluation model, optimise the precision and recall of the model.

3 | PROBLEM DEFINITION

In this section, we first introduce the definitions used in this study and then describe the method of outlier detection.

Definition 1 (Road Traffic Flow Matrix, abbreviated as Rf***)*** Rf is a statistical matrix based on the collected data with timestamps on road sections. For 1 day, Rf can be represented as a $127,049 \times 24$ numerical matrix which is devoted as

$$Rf = \begin{bmatrix} Rf_{11} & Rf_{12} & \cdots & Rf_{1n} \\ Rf_{21} & Rf_{22} & \cdots & Rf_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Rf_{m1} & Rf_{m2} & \cdots & Rf_{mn} \end{bmatrix} \quad (1)$$

In matrix (1), $m = 127,049$ (road sections) and $n = 24$ (hours). The matrix represents the total number of vehicles passing through each marked road section in each unit time

TABLE 1 Comparison of relevant researches

Literature	Approaches	Algorithm	Data	Spatial-temporal detection
[7]	statistical-based	DPMM	GPS data	temporal
[11]	statistical-based	PMM-CHMM	GPS data	temporal
[16]	density or distance-based	DBN	infrastructural elements & GPS data	spatial and temporal
[19]	density or distance-based	STLP-OD	GPS data	spatial and temporal
[22]	density or distance-based	ST-BDBCAN	GPS data	spatial
[25]	pattern-based	IFA-STODM	traffic signals & GPS data	spatial and temporal
[27]	pattern-based	SVM	heart rate & GPS data	spatial and temporal

(hour) period of 1 day. For example, $Rf_{11} = 1$ means one vehicle passed through the road section 1 between 0 and 1 o'clock.

***Definition 2 (Neighbour Road Section Matrix, abbreviated as** Nr)* *Nr* is also a matrix based on the collected data with timestamps on road sections. For 1 day, a unit of *Nr* matrix can be represented as an $m \times 24$ numerical matrix which is devoted as

$$Nr = \begin{bmatrix} Nr_{11} & Nr_{12} & \cdots & Nr_{1n} \\ Nr_{21} & Nr_{22} & \cdots & Nr_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Nr_{m1} & Nr_{m2} & \cdots & Nr_{mn} \end{bmatrix} \tag{2}$$

In matrix (2), m represents the numbers of neighbour road sections with the same road traffic flow pattern and $n = 24$. The matrix represents the total number of vehicles passing through each neighbour road section with the same road traffic flow pattern in each unit time (hour) period of 1 day. For example, $Nr_{11} = 1$ means one vehicle passing through the neighbour road section 1 with the same road traffic flow pattern between 0 and 1 o'clock. Here, we name the road sections with the same road traffic flow pattern as neighbour road sections.

***Definition 3 (Outlier Section Matrix, abbreviated as** Os)* The *Os* Matrix is the statistical data of the road section with timestamps. A unit of *Os* can be denoted as a $127,049 \times 24$ numerical matrix of $\{0 \mid 1\}$ which is collected by traffic control authorities from social media or traffic accidents attended by traffic police. A unit of *Os* is denoted as

$$Os = \begin{bmatrix} Os_{11} & Os_{12} & \cdots & Os_{1n} \\ Os_{21} & Os_{22} & \cdots & Os_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ Os_{m1} & Os_{m2} & \cdots & Os_{mn} \end{bmatrix} \tag{3}$$

The matrix (3) indicates the condition of outlier in each unit time (hour) for each road section in a day. For example, $Os_{11} = 1$ means the first road section in the first time period of the day is identified as an outlier. Then $Os_{11} = 0$ means that road section is considered as normal road for that time period.

Definition 4 (Probability Matrix) The probability matrix here contains probability matrix of road traffic flow within T days (probability of Rf, PRf^T) data and probability matrix of neighbour road sections (probability of Nr, PNr) data. Both of them are probability matrix of road traffic flow with timestamps, which can be denoted as a numerical matrix in the [0,1] interval. The probabilities are calculated by Kernel Density Estimation (KDE) [29] which are defined as follow.

$$PRf^T(Rf_{it}^d) = \frac{1}{Th_1} \sum_{j=1}^{T} \Phi\left(\frac{Rf_{it}^d - Rf_{it}^j}{h_1}\right), \tag{4}$$

$$PNr(Nr_{it}^d) = \frac{1}{Nh_2} \sum_{j=1}^{N} \Phi\left(\frac{Nr_{it}^d - Nr_{jt}^d}{h_2}\right), \tag{5}$$

where Rf_{it}^d and Nr_{it}^d are the traffic volume of ith road section on day d at time period t. N is the number of neighbour road sections in the cluster of ith road section. The kernel $\mathbf{\Phi}$ of KDE is set as Gaussian model with the bandwidth $h_1 = \frac{5\sigma}{T}$ and $h_2 = \frac{5\sigma}{N}$ while σ is represents the standard deviation of Rf and Nr.

***Definition 5 (Score Matrix, abbreviated as** SM)* The *SM* is the output matrix of the anomaly detection and evaluation model. A unit can be denoted as a numerical matrix of $127,049 \times 24$. The value of SM_{ij} is $[-2, 2]$. The larger the value is, the more likely the road section is to be a burst anomaly road section. A unit of *SM* is denoted as

$$SM = \begin{bmatrix} SM_{11} & SM_{12} & \cdots & SM_{1n} \\ SM_{21} & SM_{22} & \cdots & SM_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ SM_{m1} & SM_{m2} & \cdots & SM_{mn} \end{bmatrix} \tag{6}$$

The neural network is trained by using the road traffic flow Rf and the neighbour road section Nr as the input layer of and *Os* as the output layer. In the actual outlier detection, when we input the real time road traffic flow data, the actual output *SM* can be obtained. We can get the outlier road sections by rounding *SM*.

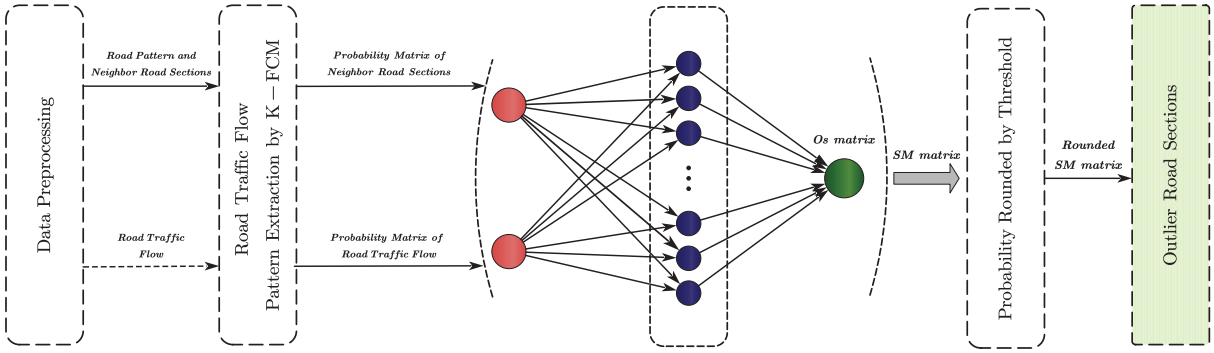


FIGURE 2 The main process of our method

4 | PROPOSED METHOD

In this section, we will introduce our method of outlier detection in detail. The main process is shown as in Figure 2. It mainly includes three parts: data pre-processing, road traffic flow pattern extraction, and outlier detection model training and evaluation.

4.1 | Data pre-processing

In this section, we carry out the basic data cleaning, map matching and data dimension reduction for the original GPS data to provide the basis for the following steps.

4.1.1 | Data cleaning and map matching

Due to equipment abnormalities, human improper operation and other accidents, we establish several principles for data cleaning to ensure the quality of the data. For example, the points beyond the latitude and longitude range of the Fifth Ring Road District in Beijing should be eliminated. For multiple points with the same timestamp recorded on the same track, only the first point can be reserved, parking point and waiting point can be cleared.

After data cleaning, considering the low-frequency sampling characteristics of GPS point, we use interactive voting map matching (IVMM) algorithm [30] to match GPS points with the most appropriate road section marked with a unique road number.

In our original data, there are 127,049 road sections that meet the requirements in the Fifth Ring district in Beijing. The identification numbers of road sections are integers between 1 and 127,049.

4.1.2 | Road traffic flow matrix of road section

After the completion of road section matching, we set the time interval as 15 minutes. The number of vehicles in the same road section of the single day within unit time (15 minutes) is counted and the road traffic flow matrix is obtained. The road traffic flow matrix is a $127,049 \times 96$ size matrix, representing the traffic flow of 127,049 different road sections in every 15 minutes (that is 96 time intervals) of each day.

Considering that most traffic congestion times in Beijing exceed 15 minutes, we add up the number of traffic every four 15 minutes to get the number of traffic every 60 minutes in order to improve the detection accuracy, that is, adjusting the unit time from the original 15 to 60 minutes (1 hour). Therefore, the road traffic flow matrix is shown as a $127,049 \times 24$ size matrix.

4.1.3 | Non-negative matrix factorisation

In order to reduce the time complexity of evaluation model learning, we consider reducing the dimension of the Rf matrix, the NMF algorithm [31, 32] is used for data dimension reduction. For any given non-negative matrix V, the algorithm can find a non-negative matrix W and a non-negative matrix H to satisfy that

$$V_{n\times m} \approx W_{n\times r} \times H_{r\times m} \tag{7}$$

The function of W and H in $V_{n\times m} \approx W_{n\times r} \times H_{r\times m}$ is relative. The W is defined as the coefficient matrix and the H is defined as the base matrix.

When H is constant, we can use three values in each row to represent the linear combination of the three basic traffic patterns, which reflects the greatest significance of using NMF algorithm in this paper. The original history traffic flow matrix V can be replaced by a coefficient matrix W, which will greatly reduce the column dimension of the matrix and will not lose the necessary information as well as no meaningless negative numbers in the results. As a result, it only needs to calculate the coefficient matrix, which greatly reduces the computational complexity and time cost.

4.2 | Road traffic flow pattern extraction

In this section, we give the detailed process of road traffic flow pattern extraction aiming to find road sections with the

same road traffic flow pattern among 127,049 road sections, which supply the input data for the subsequent anomaly detection model.

As the matrices are still characterised by high dimensions and irregular data form, we improve the fuzzy C-means (FCM) algorithm [33] to extract the road traffic flow pattern. The improved algorithm has low time complexity and is suitable for clustering analysis of high-dimensional data.

4.2.1 | Road traffic flow pattern extraction based on K-FCM algorithm

The FCM algorithm is a soft partition algorithm derived from the C-means algorithm [34]. The basic idea is to maximise the similarity between sample points divided into the same cluster. The biggest difference between the FCM algorithm and the C-means algorithm is that the degree of membership is introduced to quantify the fuzzy similarity of each sample point. It can improve the quality of clustering and solve the problem of clustering boundary sample points as well as reduce the drawbacks caused by the rigid division of C-means algorithm. However, FCM algorithm also inevitably has some problems, such as the randomness of cluster centre points selection and the lack of considering the local information and neighbourhood relations of the sample points, which easily conduces to the wrong cluster centre selection or converging to the local optimum.

In view of the above possible problems, the literature [35] optimises the selection of the initial cluster centre point and the literature [36, 37] optimises the FCM algorithm through the density peak algorithm. Here, we propose a clustering algorithm which named the Optimal K cluster centre FCM clustering algorithm (K-FCM). The "K" value is optimised based on the combination of KNN algorithm and clustering by fast Search and find of density peaks (DPC) algorithm, which represents the local density representation method of DPC algorithm optimised by the KNN algorithm.

4.2.2 | Optimisation of cluster centre set

The DPC algorithm can spontaneously discover the centre of the data cluster and efficiently achieve clustering for arbitrary shape data sets. As the cutoff distance d_c in the algorithm affects the local density ρ_i, the nearest neighbour information in the KNN algorithm is introduced to represent the local density of the sample points. The specific steps are as follows:

STEP 1 Integrate the effects of geographic distance and pattern change. We select neighbour roads based on geographical proximity and similarity of traffic patterns, which is

$$\begin{cases} D(i,j) = (1-\omega)\dfrac{D_g(i,j)}{\zeta} + \omega D_t^T(i,j) \\ D_t^T(i,j) = \sqrt{\sum_{k=1}^{n} \left| V_{ik} - V_{jk} \right|^2} \end{cases} \tag{8}$$

In the above equation, the reduced-dimensional matrix V is used to calculate the neighbour road sections. ω is the proportionality factor used to balance the weights of two road sections' distances. $D_g(i,j)$ is the actual distance between the centre points of the two road sections. $D_t^T(i,j)$ represents the difference of flow patterns between two road sections calculated using Euclidean distance where V_{ik} is the kth coefficient of ith road section in matrix V. In order to make the two distances comparable, the parameter ζ is introduced here to adjust the scale of $D_g(i,j)$ which takes the value of 1000.

STEP 2 Use the nearest neighbour information to calculate the local density:

$$\rho_i = \sum_{j \in KNN_i} e^{-D(i,j)}. \tag{9}$$

STEP 3 Calculate the number of samples points whose Euclidean distance between the sample points x_i and the sample points x_j are less than their cutoff distance:

$$\delta_i = \min_{j:\rho_j > \rho_i}(D(i,j)). \tag{10}$$

STEP 4 Calculate the value of the comprehensive variable γ_i which indicates the possibility that the sample points x_i is the peak density point (the greater the γ_i value, the greater probability that the point x_i is the peak density point):

$$\gamma_i = \rho_i \times \delta_i \tag{11}$$

STEP 5 Sort the sample points in descending order according to the γ_i value. Select the first c sample points which is the initial cluster centre set $C_i^{(0)}$.

4.2.3 | Clustering of data

In the above process, we solve the problem of random selection of K cluster centres in FCM algorithm. Then, the K-FCM algorithm can be used to cluster the sample data. The algorithm process is shown in Table 2.

Here, the road sections with the same road traffic flow pattern are clustered by the proposed K-FCM clustering algorithm. NMF algorithm is used to reduce the dimension of Rf matrix into coefficient matrix as sample data to obtain the initial cluster centre set.

We use the K-FCM algorithm to cluster all the road sections, and road sections in the same cluster are neighbour road sections for each other. Accordingly, the Nr matrix is obtained.

4.2.4 | Optimisation of the number of clusters

The quality of the clustering effect is not only affected by the applicability of the algorithm itself, but also by the given number of clusters in the initial stage of clustering. To achieve better

TABLE 2 K-FCM algorithm

Algorithm 1 K-FCM algorithm

Input: $\boldsymbol{D}$-set of sample data, $\boldsymbol{D}^j$-jth element of set $\boldsymbol{D}$, c-number of centres in cluster centre set

1: **for** each element $\boldsymbol{D}^j \in \boldsymbol{D}$ **do**
2: Set ω and compute $D(i, j)$ according to Equation (4.2.2)
3: Compute ρ_j from $D(i, j)$, according to Equation (4.2.2)
4: Compute δ_j from $\rho_j$and $D(i, j)$, according to Equation (4.2.2)
5: Compute γ_j from ρ_j andδ_j, $\gamma_j = \rho_j \times \delta_j$, according to Equation (4.2.2)
6: **end for**
7: Decrease $\boldsymbol{D}^j$ from γ_j and obtain $\boldsymbol{S}$, $\boldsymbol{S}$ = set of decreased $\boldsymbol{D}^j$
8: **while** each integer $j \in [1, c]$ from $j = 1$ **do**
9: Assign $\boldsymbol{C}_i^{(0)}$ as the initial cluster centre set, $\boldsymbol{C}_{i(j)}^{(0)} = \boldsymbol{S}_{i(j)}$
10: **end while**
11: Initialise the number of iterations t, $t = 1$
12: Compute the membership matrix $\mathcal{U}_{(ij)}^{(t)}$ according to FCM
13: Compute the objective function $\mathcal{J}^{(t)}$ according to FCM
14: **for** each t **do**
15: Update the cluster centre set $\boldsymbol{C}_i^{(t+1)}$ according to FCM
16: Update the membership matrix $\mathcal{U}_{(ij)}^{(t+1)}$ according to FCM
17: Update the objective function $\mathcal{J}^{(t+1)}$ according to FCM
18: **if** $\mathcal{J}^{(t+1)} - \mathcal{J}^{(t)} > 0$ **then**
19: $t+ = 1$
20: **else**
21: **break;**
22: **end if**
23: **end for**

Output : $\boldsymbol{C}_i^{(t)}$-targeted centre set, $\mathcal{U}_{(ij)}^{(t)}$-targeted membership

clustering results, here, we use the elbow method to find the optimal number k of the clusters.

In the elbow method, the objective function is defined as

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2. \tag{12}$$

Among the function, SSE is the sum of the squared errors (SSE) which is the clustering error of all samples and measures the quality of clustering. C_i is the ith cluster, p is the sample point in C_i and m_i is the centroid of C_i.

Obviously, with the increase of k, the number of sample clusters increases and the division of sample is more detailed. The results show that the polymerisation quality is better and the SSE of objective function value becomes smaller.

When k is less than the optimum cluster number, the increasing of k will greatly increases the polymerisation degree of each cluster, which leads to the faster decline rate of SSE. When k reaches the optimal cluster number, the effect on the polymerisation degree obtained by increasing k will rapidly decrease, making the descent rate of SSE decrease sharply. Then, the SSE tends to stable with the continuous increase of k value. Therefore, the relationship between SSE and k is elbow shaped. The k value corresponding to elbow is the optimal clustering number of sample data.

4.3 | Anomaly detection based on BP neural network

In this section, we establish an evaluation model based on PRf, PNr and Os matrices.

4.3.1 | Outlier detection and evaluation model

PRf and PNr reflect the occurrence probability of the value of Rf and Nr in the time dimension. The lower the probability is, the less likely the current traffic flow value will appear on the Rf and Nr matrix, that is, the more likely there will be traffic outliers. However, it is not known whether there is a linear relationship between PRf matrix and PNr matrix, back-propagation neural network (BPNN) learning algorithm is used for further evaluation and error correction.

BPNN algorithm [38] has the advantages of strong data compatibility and simple operation process while it still falls into a local optimal solution due to the error function selection. Therefore, the PSO algorithm is introduced to optimise the weights and thresholds of the BPNN algorithm. PSO algorithm is a global search optimisation algorithm based on the concepts of "evolution" and "population", which is helpful to solve the problem of optimal solution in complex space.

PRf and PNr matrices are taken as the input layer, and the Os matrix is taken as the output layer. We firstly initialise the particles and populations in the solvable space, express the characteristics of the particles by position, speed, and fitness, the fitness value obtained by BPNN represents the quality of particles, the particles are calculated every time while they change in the solution space. By comparing the new fitness of particles with individual extreme values and group extreme values to achieve the goal of optimisation, we use the PSO algorithm to optimise the initial connection weight and threshold in BPNN while the optimal solution is assigned to BPNN for prediction. The algorithm process is shown in Table 3.

According to the trained neural network, the real-time road traffic flow data is input for detection. We will get an SM matrix. The SM matrix is a $127,049 \times 24$ probability matrix and each element represents the possibility of outliers of the road sections within a unit time (hour).

Consider that the value of SM_{ij} is not strictly limited to $[0, 1]$, and most of them are decimal. In the subsequent experiments, we introduce a threshold to round the SM matrix so as to intuitively get the outlier road sections from the SM matrix. The rounded results are analysed through specific experimental data. Please refer to Section 5 for detailed experimental procedures and results.

TABLE 3 PSO-BP algorithm

Algorithm 2 PSO-BP algorithm

Input: Rf- probability matrix of road historical traffic flow, PNr-probability matrix of neighbour road section, $t_{(i)}$-targeted number of iterations of PSO, $f_{(i)}$-targeted adaptability of PSO, n-number of hidden layers in BPNN

Determine the topology of BPNN from the number of hidden layers n, $n = 11$
Initialise weights and thresholds of PSO
Initialise velocity and position of particles in PSO
Input PRf and PNr matrix
Assign the number of iterations in PSO t_{PSO}, $t_{PSO} = 1$
for each t_{PSO} **do**
    Compute the fitness population f_{PSO}
    Update velocity and position of particles
    Update individual extremum and global extremum position of particles
    if $t_{PSO} \geq t_{(i)}$ or $f_{PSO} \geq f_{(i)}$ **then**
        Obtain the best weights and thresholds of PSO
        Assign the number of iterations in BPNN t_{BP}, $t_{BP} = 1$
        for each t_{BP} **do**
            Compute the error
            Update weights and thresholds of BPNN
            if meets the output goal of BPNN **then**
                break;
            else
                $t_{BP} += 1$
            end if
        end for
        break;
    else
        $t_{PSO} += 1$
    end if
end if

Output : SM-score matrix

TABLE 4 Statistics of data set

Trajectories	Data duration	1 to 30 November 2018
	No. of taxis	12,712
	No. of effective days	30
	No. of Avg. sampling intervals (s)	60
Roads	No. of road sections	180,350
	No. of road nodes	132,273
Outlier reports	Data duration	21 to 30 November 2018
	Avg. reports of day	21

5 | EXPERIMENT

To test the accuracy and advanced nature of the method, a series of comparative experiments are conducted using the same data set and experimental environment.

5.1 | Data set and experimental configuration

For verifying the feasibility of our method. Beijing, which has a relatively complex domestic traffic conditions, is selected as the data experimental city. The GPS positioning information of 12,712 taxis travelling within the Fifth Ring district of Beijing for 30 days is used to detect the anomalistic traffic conditions in Beijing. Each GPS point has nine fields, which are vehicle ID, sampling time, longitude, latitude, speed, direction and vehicle status.

According to statistics, during the 30 days from 1 to 30 November 2018, there are 33,651,069 GPS points collected from 12,712 taxis in Beijing. The number of GPS points collected by each vehicle ranges from 1 to 19,285, with an average of 26,349. The order of $r = 6$ in NMF algorithm is determined. The naming of the data set is based on the date. For example, the data on 21 November is named as $BJ21$.

The IVMM algorithm is used to match each GPS point to the corresponding road segment. Also, we use the road segmentation standard in the international open street map (OSM) for road segmentation, with a total of 180,350 road segments and 132,273 road nodes.

Taking the GPS points on 21 November as an example, the part of single day density distribution of some GPS points is shown in Figure 3.

In the data set, the average sampling time is 60 seconds. To make the time description better fit the traffic characteristics of modern cities, we divide the day from 0:00 to 24:00 into 24 unit time sections equally. For each road segment, we count the number of vehicles that pass by in each unit time and consider it as one element of the Rf matrix. Therefore, one unit of time interval is 1 hour.

In summary, the data set is described in Table 4.

To calculate the parameters of the method, we determine the approximate range of the parameters in the algorithm using a priori experiments. After that we determine the values of the distance parameter and the number of clusters using univariate experiments.

In parallel experiments, we use pre-processed traffic flow data and Os data of Beijing for 21, 23 and 29 November 2018 to verify the result of the method.

In the robustness check of the method, we compare the detection performance of the algorithm before and after optimisation and examine the effect of different time splits on the detection results.

In comparative experiment, the data of 29 November 2018 is used to compare with different methods.

The experimental environment is Windows 10.1903 operating system. The hardware configuration is Inter(R) i7-9750H processor with 32GB memory and T2000 graphics card with 4GB.

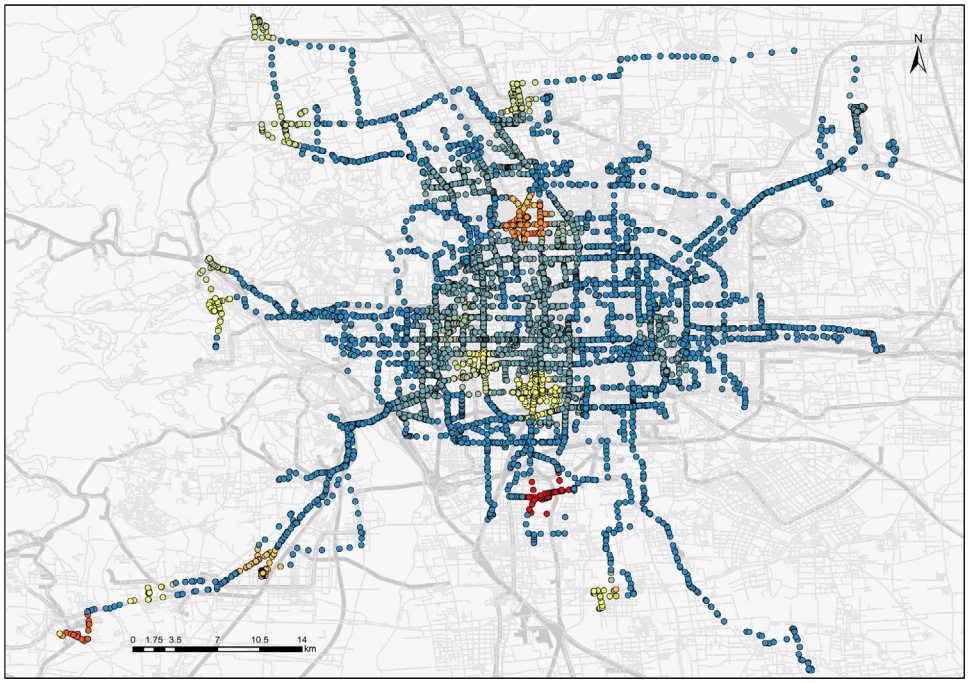


FIGURE 3 Density distribution of part of GPS points in core road area

5.2 | Evaluation metrics

As the probability of rounding elements in the SM matrix obtained by the outlier detection model is too small, a positive threshold ϵ is introduced for error correction. When the score value is in $[1-\epsilon, 2]$, the road section is regarded as an outlier road section. To qualitatively judge the improvement of the accuracy with the introduction of threshold value, the precision rate (Precision, P), the recall rate (Recall, R) and the average value (Average, F_1) are introduced. The outlier road section is positive and the normal section is negative.

$$R = \frac{TP}{TP + FN}, \tag{13}$$

$$P = \frac{TP}{TP + FP}, \tag{14}$$

$$F_1 = \frac{2(P \times R)}{P + R}. \tag{15}$$

5.3 | Effect of parameters setting on method performance

STEP 1 Considering that we set the distance of neighbour road sections as two parts in the K-FCM algorithm which refers to the actual distance and the Euclidean distance, the proportion of the two distances in the total distance necessarily affects how well the clustering algorithm classifies the sample points and thus affects the detection for outlier points. After performing a rough pre-order experiment, we set the number of clusters in the model as $c = 300$ and the threshold of SM matrix as $\epsilon = 0.17$ while we select the test data set as $BJ29$. For $\omega \in [0, 1]$, we set up an effect detection every 0.1 of ω which includes the precision P, recall R, F_1 measure and detection time. The results are shown in Table 5.

TABLE 5 Result with different ω

ω	P	R	F_1	Detection time (s)
0	40.86%	80.83%	54.30%	15.7
0.1	41.20%	65.53%	50.60%	17.5
0.2	43.07%	65.38%	51.93%	18.4
0.3	48.87%	81.22%	61.04%	16.2
0.4	68.74%	81.50%	74.58%	19.1
0.5	72.12%	94.77%	81.91%	27.9
0.6	74.56%	92.46%	82.55%	28.2
0.7	81.70%	93.85%	87.35%	26.1
0.8	84.92%	89.62%	87.21%	27.3
0.9	74.56%	92.76%	82.67%	26.6
1.0	72.53%	82.29%	77.11%	27.8

According to the experimental results, we can find that the optimal detection can be achieved when ω is taken as 0.7 or 0.8 in the case of a single variable. When $\omega = 1$, that is, only the actual distance of the road section is taken as the distance between sample points, the detection results are the

TABLE 6 Experiment results of parameter c

c	Avg SSE	Avg CHI	Avg SC	Avg P	Avg R	Avg training time (h)
287	367.231	3149.25	0.679	81.92%	82.84%	1.756
288	361.526	3258.85	0.679	81.92%	82.84%	1.76
289	356.412	3284.56	0.679	83.84%	85.13%	1.85
290	347.851	3233.89	0.681	83.84%	85.13%	1.889
291	341.839	3236.78	0.681	83.84%	87.95%	1.88
292	335.239	3256.76	0.681	87.83%	87.95%	1.87
293	330.291	3276.66	0.683	87.83%	88.56%	1.89
294	326.038	3308.76	0.683	87.83%	88.56%	1.84
295	322.073	3294.49	0.683	91.34%	93.19%	1.93
296	318.151	3377.34	0.685	91.34%	94.96%	1.89
297	314.458	3358.48	0.685	94.24%	94.96%	1.92
298	310.081	3368.96	0.687	94.24%	94.96%	1.97
299	308.123	3402.12	0.688	95.38%	96.23%	1.92
300	306.973	3405.75	0.689	95.38%	96.23%	1.96
301	305.197	3363.47	0.689	88.64%	96.23%	1.95
302	304.827	3337.49	0.688	88.64%	91.98%	1.93
303	303.198	3325.53	0.688	87.83%	91.98%	1.92
304	302.013	3323.13	0.691	87.83%	91.98%	2.23
305	301.187	3381.90	0.691	81.94%	84.58%	2.31
306	301.932	3337.47	0.692	81.94%	84.58%	2.39

worst but the detection time consumed is the shortest. When ω gradually converges to 1, the detection effect decreases after reaching a peak in the middle section while the detection time stabilises after increasing to 25 seconds. As the conclusion, we set $\omega = 0.8$ as the final percentage of the Euclidean distance in the distance formula of K-FCM algorithm.

STEP 2 Here, we verify the influence of parameter adjustment on the experimental results and test our optimised algorithm on the final experimental results. We conduct two groups of experiments, respectively, describe the experiments effects with the metrics sum of the SSE, Calinski–Harabasz Index (CHI), Silhouette Coefficient (SC), P, R and model training time under the experimental conditions of $\omega = 0.8$ and $\epsilon = 0.18$. Considering the time complexity and the accuracy of the results, we still set $r = 6$ in the NMF algorithm and use the "*BJ*29" data for experiments.

In the pre-experiment, we use the K-FCM and PSO-BP algorithm proposed in this paper to test the parameter c in Table 2. Here we set c to an integer of [287,306] for comparison experiments. The average experiment results are shown in Table 6.

The results show that with the increase of c value, the SSE value decreases, and the change rate gradually decreases, especially the change of SSE is extremely slight when $c > 297$. The two values of CHI and SC can reflect the clustering effect to a certain extent. It can be found that when $c > 299$, CHI and SC reach the highest value and fluctuate slightly in a certain range. P obtains the best value when $c = 299$. The trend of model training time basically increases with the increase of c, but the change rate will increase sharply when $c > 304$. In general, when c is obtained [299,300], the overall effect of the experiment is better, which further reflects that the value of c has a more direct impact on this experiment.

TABLE 7 Result of outlier detection

Name	Data size	P	R	F_1	ϵ	Detection time (s)
*BJ*21	127,049	97.63%	98.01%	97.82%	0.178	31.2
*BJ*23	127,049	95.64%	96.94%	96.29%	0.178	27.3
*BJ*29	127,049	95.38%	96.23%	95.81%	0.180	28.4

5.4 | Performance analysis in parallel experiments

Here, we carry out experiments to verify the performance of the proposed method.

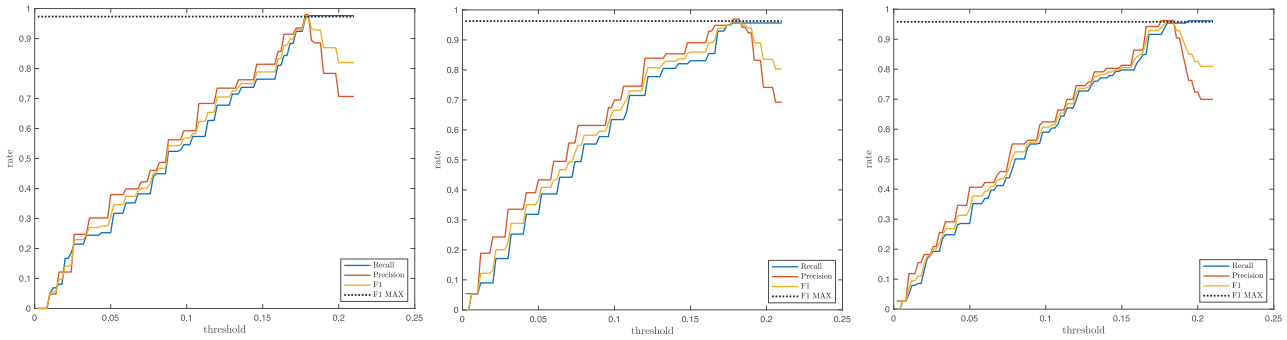
Setting the value interval of ϵ as [0,0.5] and increasing the value by 0.02, finally, the three-value trend charts of P, R and F_1 in 3 days are obtained, as shown in Figure 4.

We can find that with the increase of threshold, R will become larger. When the threshold increases, the number of sections identified as outliers will increase, while the corresponding sections not considered as outliers will be more likely to be detected as outliers. Unlike R, P is not a function of monotone trends. With the increase threshold value, the number of sections mistakenly detected as outliers increases accordingly, which leads to the rapid decline of P after reaching the extreme value. The change pattern of P and R changes leads to an extreme point of F_1. Therefore, the standard threshold of the day is the ϵ value when F_1 reaches the extreme point. The corresponding values of P, R, F_1, ϵ and the total detection time are shown in Table 7.

As shown in Table 6, the average value of P reaches 96.22%, the average value of R reaches 97.06%, the average value of F_1 reaches 96.64% and the average detection time is 28.97 seconds.

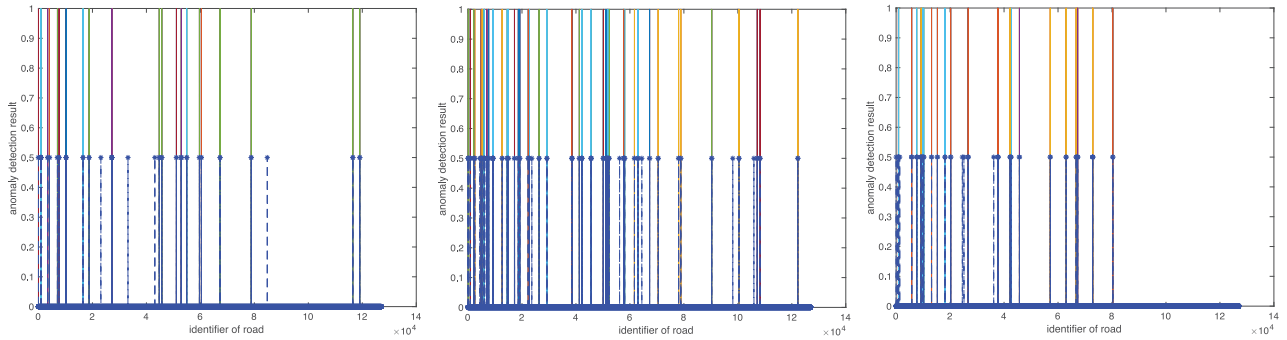
The following results are the corresponding map of anomaly road detection. For the convenience of comparison, the detection results after threshold rounding are divided into half, as shown in Figure 5.

The outlier detection results show the similarities and differences between the results obtained by this method and the actual results. Among them, each half-valued point represents a road section that is detected as an outlier, and each point with a value of 1 represents a real outlier road section. If the coordinates are the same, it means that the road section corresponding to the abscissa is correctly detected as an outlier road section. On the contrary, it means that the road section corresponding to the abscissa has been wrongly detected or not detected. According to the detection results, the following heat map of outlier detection can be obtained by using the OSM endpoint identification (the marked point at this time is the centre point



(a) Day 21 (b) Day 23 (c) Day 29

FIGURE 4 The value tendency of P, R F_1



(a) Day 21 (b) Day 23 (c) Day 29

FIGURE 5 Outlier detection results

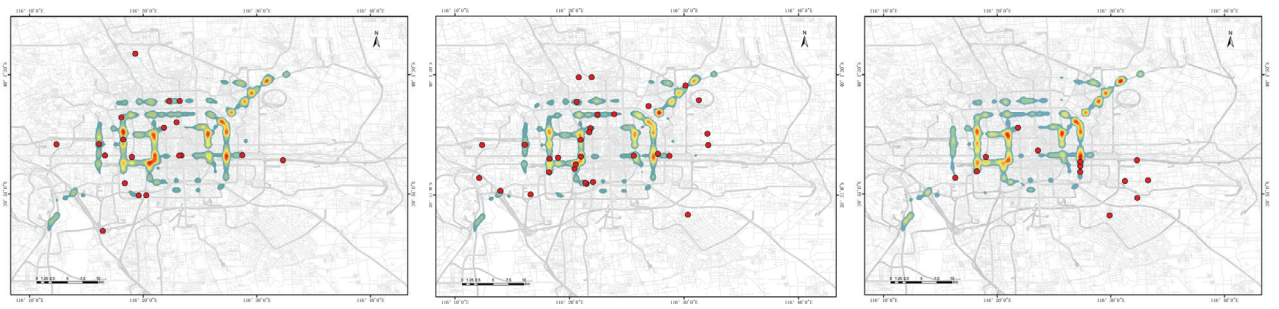
of the detected outlier road section), as shown in Figure 6. The red points represent the outliers.

5.5 | Analysis of algorithm robustness

Considering that we choose the time interval based on the average congestion time in Beijing and the empirical values of time intervals in most studies, here we check the robustness of the algorithm for different time intervals. We still choose the *BJ*21, *BJ*23 and *BJ*29 data sets and set the same values of ω, c and ϵ as in the previous experiments, considering 15, 30, 45 and 60 minutes time intervals, respectively, for the experiments. The average results are shown in Table 8.

TABLE 8 Comparison of time intervals

Time interval (min)	Avg P	Avg R	Avg training time (h)
15	49.12%	64.28%	4.36
30	81.37%	85.78%	2.98
45	88.12%	91.29%	2.32
60	96.22%	97.06%	1.97



(a) Day 21 (b) Day 23 (c) Day 29

FIGURE 6 Actual effect of outliers detection in road

TABLE 9 Comparison of optimised algorithm

F	Avg *P*	Avg *R*	Avg training time (h)
FCM+BP	63.89%	69.53%	2.79
K-FCM+BP	74.48%	79.29%	2.24
FCM+PSO-BP	78.12%	82.88%	2.04
K-FCM+PSO-BP	95.38%	94.09%	1.92

Based on the experimental results, we can know that the detection accuracy and computing time are improved after a time interval greater than 30 minutes. When the time interval is too small, the detection accuracy of the algorithm is at a lower value and the computing time increases significantly. Such results are also more in line with the road congestion time in urban areas, which further reflect the great robustness of our method.

In another group of experiments, for the K-FCM and PSO-BPNN algorithms, we carry out experiments to verify the algorithms before and after optimisation. The average values are shown in Table 8.

The results in Table 9 show that the optimisation of FCM algorithm can have less time cost, while the optimisation of BP algorithm can have better precision and recall. Our method combines two optimisation algorithms to get the optimal values of precision, recall rate and model training time. Therefore, the results are greatly improved compared with the experiment without algorithm optimisation. Therefore, the proposed method has a significant effect on the experimental results.

5.6 | Algorithm comparison in comparative experiments

To evaluate the performance of the proposed method quantitatively, two different types of the latest representative traffic outlier detection methods are selected based on the comprehensive analysis of relevant research. The data set *BJ*29 is used to conduct comparative experiments in the same experimental environment. The first method is a statistical-based approach [11] that also studied Beijing traffic data as we do. They introduce a Poisson mixture model (PMM) coupled hidden Markov model (CHMM) outlier detection method to detect traffic anomalies. It is worth mentioning that they also use the location data of taxis driving around the city to represent the traffic situation in the city. They even investigated the real causes of the outliers. The density-based spatial and temporal label propagation OD (STLP-OD) framework [19] introduced the CHMM model to enhance the impact of outlier candidates while utilising the basic label propagation algorithm. Detection performance with the help of orbiting data relay system (ODRS) can be significantly improved. The average experimental results are shown in Table 10.

We draw the following conclusions according to the experimental results: (1) compared with the existing method, the proposed method has higher detection accuracy. (2) The detection time efficiency of the proposed method is not optimal, and further improvement of neural network model is expected to achieve shorter detection time in the future.

TABLE 10 Comparison of related algorithm

Algorithm	*P*	*R*	F_1	Detection time (s)
K-FCM+PSO-BP	95.38%	96.23%	95.81%	28.4
PMM-CHMM	79.34%	87.10%	83.04%	7.5
STLP-OD+ODRS$^+$	89.63%	93.24%	91.34%	38.2

Here, it should be noted that we have not done comparative experiments with pattern-based methods, because most of these methods need not only GPS positioning data, but also more sensor data. For example, in the application of using support vector machine (SVM) and machine learning algorithms [27], the authors also needed the driver's heartbeat data. However, it is certain that this type of approaches will be much larger in data volume than methods that only require GPS location data while the computation time is mostly longer.

In summary, the proposed method in this paper has good performance in detection accuracy and time efficiency. More importantly, the method provides a more realistic representation of the actual significance of outlier in traffic events, which is important for understanding and utilising traffic outlier data.

6 | CONCLUSION

This paper studies the problem of traffic outlier detection and propose a highly efficient framework. In terms of simply counting road traffic probabilities, our method uses the densities and distances of road sections to further characterise road conditions in the spatial dimension. Compared to cluster screening based on distance and density alone, we use the optimised clustering algorithm to screen out road sections with similar road patterns while effectively circumventing the drawback of requiring multidimensional data in the pattern-based approaches. Finally, we make reasonable use of the optimised neural network for learning and efficiently detecting traffic outliers with the precision of 95.38% and recall of 96.23%, contributing to the understanding and utilisation of outliers in the field of transportation.

CONFLICTS OF INTEREST

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

AUTHOR CONTRIBUTIONS

The author Ruihao Zeng and Xing Wang contributed equally to this work. Xing Wang and Ruihao Zeng completed the writing of the thesis, Fumin Zou conducted the guidance of the thesis, Faliang Huang conducted part of the experiment of the thesis, and Biao Jin conducted part of the experiment and grammar modification of the thesis.

ACKNOWLEDGEMENTS

This research was funded by the Natural Science Foundation of China (Grant No. 61962038), the Foreign Cooperation Project of Fujian Provincial Department of Science and Technology (Grant No. 2020I0014), the Startup Project of Doctoral Research of Fujian Normal University, and in part by the Guangxi Bagui Teams for Innovation and Research (Grant No. 201979). We also thank all the partners who take part in the research.

ORCID

Xing Wang https://orcid.org/0000-0003-1144-0995
Ruihao Zeng https://orcid.org/0000-0003-2927-1018

REFERENCES

1. Zimek, A., Filzmoser, P.: There and back again: Outlier detection between statistical reasoning and data mining algorithms. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 8(6), e1280 (2018)
2. Djenouri, Y., Zimek, A. & Chiarandini, M. : Outlier detection in urban traffic flow distributions. In 2018 IEEE International Conference on Data Mining (ICDM), pp. 935–940. IEEE, Piscataway, NJ (2018)
3. Bhowmick, K., Narvekar, M.: Trajectory outlier detection for traffic events: a survey. In: Bhalla, S., Bhateja, V., Chandavale, A., Hiwale, A., Satapathy, S. (eds.) Intelligent Computing and Information and Communication, pp. 37–46. Springer, Singapore (2018)
4. Djenouri, Y., et al.: A survey on urban traffic anomalies detection algorithms. IEEE Access 7, 12192–12205 (2019)
5. Djenouri, Y. & Zimek, A. : Outlier detection in urban traffic data. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, pp. 1–12. IEEE, Piscataway, NJ (2018)
6. Domingues, R., et al.: A comparative evaluation of outlier detection algorithms: experiments and analyses. Pattern Recognit. 74, 406–421 (2018)
7. Ngan, H.Y., Yung, N.H., Yeh, A.G.: Outlier detection in traffic data based on the Dirichlet process mixture model. IET Intell. Transp. Syst. 9(7), 773–781 (2015)
8. Kingan, R.J., Westhuis, T.B.: Robust regression methods for traffic growth forecasting. Transp. Res. Rec. 1957(1), 51–55 (2006)
9. Turochy, R.E. & Smith, B.L. : Applying quality control to traffic condition monitoring. In Proceedings of the 2000 IEEE Intelligent Transportation Systems (Cat. No. 00TH8493), pp. 15–20. IEEE, Piscataway, NJ (2000)
10. Campos, G. O., Zimek, A., Meira, W.: An unsupervised boosting strategy for outlier detection ensembles. In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 564–576. Springer, Cham (2018)
11. Yujun, C., et al.: Spatial-temporal traffic outlier detection by coupling road level of service. IET Intell. Transp. Syst. 13(6), 1016–1022 (2019)
12. Dang, T.T., Ngan, H.Y., Liu, W.: Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. In 2015 IEEE International Conference on Digital Signal Processing (DSP), pp. 507–510. IEEE, Piscataway, NJ (2015)
13. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 427–438. ACM, New York, NY (2000)
14. Campos, G.O., et al.: On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min. Knowl. Discov. 30(4), 891–927 (2016)
15. Tang, J., Ngan, H.Y.: Traffic outlier detection by density-based bounded local outlier factors. Inf. Technol. Ind. 4(1), 6 (2016)
16. Munoz-Organero, M., Ruiz-Blaquez, R., Sánchez-Fernández, L.: Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. Comput. Environ. Urban Syst. 68, 1–8 (2018)
17. Kontaki, M., et al.: Efficient and flexible algorithms for monitoring distance-based outliers over data streams. Inf. Syst. 55, 37–53 (2016)
18. Huang, T., Sethu, H., Kandasamy, N.: A new approach to dimensionality reduction for anomaly detection in data traffic. IEEE Trans. Netw. Serv. Manag. 13(3), 651–665 (2016)
19. Pu, J., et al.: STLP-OD: spatial and temporal label propagation for traffic outlier detection. IEEE Access 7, 63036–63044 (2019)
20. Schubert, E., Zimek, A., Kriegel, H. P.: Fast and scalable outlier detection with approximate nearest neighbor ensembles. In International Conference on Database Systems for Advanced Applications, pp. 19–36. Springer, Cham (2015)
21. Kirner, E., Schubert, E., Zimek, A.: Good and bad neighborhood approximations for outlier detection ensembles. In International Conference on Similarity Search and Applications, pp. 173–187. Springer, Cham (2017)
22. Duggimpudi, M.B., et al.: Spatio-temporal outlier detection algorithms based on computing behavioral outlierness factor. Data Knowl. Eng. 122, 1–24 (2019)
23. Li, X., et al.: Temporal outlier detection in vehicle traffic data. In 2009 IEEE 25th International Conference on Data Engineering, pp. 1319–1322. IEEE, Piscataway, NJ (2009)
24. Campos, G.O., Meira Jr, W. & Zimek, A. : Outlier detection in graphs: On the impact of multiple graph models. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, pp. 1–12. ACM, New York, NY (2018)
25. Sun, D., et al.: ST TD outlier detection. IET Intell. Transp. Syst. 11(4), 203–211 (2017)
26. Cao, L., et al.: Efficient discovery of sequence outlier patterns. Proc. VLDB Endow. 12(8), 920–932 (2019)
27. Bhuyan, M.H., Bhattacharyya, D.K., Kalita, J.K.: A multi-step outlier-based anomaly detection approach to network-wide traffic. Inf. Sci. 348, 243–271 (2016)
28. Blázquez, R.R., Organero, M.M., Fernández, L.S.: Evaluation of outlier detection algorithms for traffic congestion assessment in smart city traffic data from vehicle sensors. Int. J. Heavy Veh. Syst. 25(3-4), 308–321 (2018)
29. Ngan, H.Y., Yung, N.H., Yeh, A.G.: A comparative study of outlier detection for large-scale traffic data by one-class SVM and kernel density estimation. In Image Processing: Machine Vision Applications VIII, pp. 94050I. International Society for Optics and Photonics (2015)
30. Yuan, J. et al.: An interactive-voting based map matching algorithm. In 2010 Eleventh international conference on mobile data management, pp. 43–52. IEEE, Piscataway, NJ (2010)
31. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(6755), 788–791 (1999)
32. Févotte, C., Idier, J.: Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput. 23(9), 2421–2456 (2011)
33. Khalilia, M. A., et al.: Improvements to the relational fuzzy c-means clustering algorithm. Pattern Recognit. 47(12), 3920–3930 (2014)
34. Jain, A. K.: Data clustering: 50 years beyond K-means. Pattern Recognit. Lett. 31(8), 651–666 (2010)
35. Wu, Z., Wu, Z., Zhang, J.: An improved FCM algorithm with adaptive weights based on SA-PSO. Neural Comput. Appl. 28(10), 3113–3118 (2017)
36. Ding, S., , et al.: An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood. Knowl.-Based Syst. 133, 294–313 (2017)
37. Lei, T., , et al.: Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering. IEEE Trans. Fuzzy Syst. 26(5), 3027–3041 (2018)
38. Li, J. C., et al.: A link prediction method for heterogeneous networks based on BP neural network. Phys. A 495, 1–17 (2018)

How to cite this article: Wang, X., et al. A highly efficient framework for outlier detection in urban traffic flow. IET Intell. Transp. Syst. 2021;1–14. https://doi.org/10.1049/itr2.12109