The object of this study is a clustering algorithm using various technologies.

This paper compares clustering algorithms that are more commonly used to analyze urban road network trajectories, the growth curve model, with the elbow method and the x-means algorithm. Experiments were conducted with various volumes of big data to determine calculation time, accuracy, and ways to increase calculation time. These results can be used to manage traffic jams in congested areas and city streets. Considering the widespread use of clustering algorithms for solving various problems, this study proposes to introduce GCM, SPGK methods for monitoring and analyzing the state of congestion on city roads. The work was carried out in the following steps: research and selection of methods based on efficiency and time, implementation of parallel computing technologies to improve computation speed, demonstration of the selected method based on collected data from a real city with visualization of the results. The growth curve model algorithm has been proven to be almost 5 times more effective than the elbow method and the x-means algorithm. The time allocated for data processing has been calculated. An increase in the volume of processed data showed an almost stable execution time t=3 s for the GCM algorithm for data with a volume of up to almost 2,000 units. The effectiveness of SPGK-means was shown for different values of the number of points. Models of the Chengdu transport network obtained using a clustering algorithm with maximum grid density of neighborhoods are presented. There are some deviations between the grid and the road network due to the large grid size. This error is explained by an error of up to one between the points and the real grid.

The results obtained clearly show how optimization of congested roads can be influenced. They provide information to obtain data on available routes, which allows you to analyze the road network individually and as a whole

Keywords: congestion point, grid K-means clustering algorithm, trajectory clustering, parallel computing

UDC 004.048
DOI: 10.15587/1729-4061.2024.298274

IDENTIFICATION OF AN ALGORITHM FOR THE ANALYSIS AND STUDY OF URBAN ROAD NETWORK TRAJECTORIES

Zhanerke Temirbekova
PhD*
Lyazat Naizabayeva
Doctor of Technical Sciences, Professor
Department of Computer Science
International IT University
Manas str., 34/1, Almaty,
Republic of Kazakhstan, 050040
Gulzat Turken*
Zukhra Abdiakhmetova
Corresponding author
PhD*
E-mail: zukhra.abdiakhmetova@gmail.com
Maxatbek Satymbekov
Acting Associate Professor*
*Department of Computer Science
Al-Farabi Kazakh National University
Al-Farabi ave., 71, Almaty,
Republic of Kazakhstan, 050040

Received date 29.01.2024
Accepted date 03.04.2024
Published date 30.04.2024

How to Cite: *Temirbekova, Z., Naizabayeva, L., Turken, G., Abdiakhmetova, Z., Satymbekov, M. (2024). Identification of an algorithm for the analysis and study of urban road network trajectories. Eastern-European Journal of Enterprise Technologies, 2 (3 (128)), 14–27. https://doi.org/10.15587/1729-4061.2024.298274*

1. Introduction

Today, traffic congestion management is of great importance to the whole world. Congestion not only reduces the efficiency of transport infrastructure, but also wastes time and resources, pollutes the environment and increases stress levels among people. Therefore, improving traffic flow management and developing more efficient public transport systems and road infrastructure is essential for sustainable development and the well-being of society. Global efforts to manage congestion can lead to improved mobility, reduced pothole emissions and reduced environmental impacts.

With the development of urbanization, traffic congestion has become an increasingly serious urban problem. The study of traffic congestion through trajectory clustering has become a hot topic in modern research. In order to relieve traffic pressure and solve the problem of traffic congestion, local governments and related departments have introduced many policies. These methods include measures such as transforming roads, establishing pedestrian viaducts, and restricting traffic. However, if the source and cause of congestion in a congested area can be known, then further strategic improvements can be effectively targeted at these congested areas to achieve the purpose of alleviating traffic pressure and solving the problem of traffic congestion. So how to discover these sources of congestion and traffic congestion, that is, traffic congestion points, provides reliable conditions for solving the problem of traffic congestion. At the same time, discovering traffic congestion points and formulating reliable solutions based on actual traffic conditions has become an effective way to solve traffic congestion.

In the current traffic congestion research, researchers start from the information contained in the trajectory data such as density, speed, driving direction, time, coordinate position, etc., through trajectory clustering and analysis of changes in related semantic data. It is necessary to find out and analyze the situation with traffic jams in order to determine the traffic jam zone. This and all the problems listed above determine the relevance of the scientific topic under consideration and the

need for further research. However, in the previous research in this paper, there is no relevant research work that starts from the trajectory density and focuses on discovering the source of traffic congestion to provide a reliable basis for solving traffic congestion. Therefore, this paper will try to start with the study of the historical trajectories of urban roads and vehicles and use existing technical means to develop and implement an algorithm for determining traffic jam points.

Therefore, studies on the development of trajectory clustering algorithms for urban road networks based on grid density are scientifically relevant.

2. Literature review and problem statement

The authors of the paper [1] present the extraction of the trajectory data of the starting point and destination point (OD) of the taxi from the trajectories of a large number of taxi trips and used these OD locations to discover the areas of interest to taxi drivers. The authors consider optimization by dividing areas into small parts, but there were unresolved issues related to combining these little parts. The authors of [2] propose a new way to detect people's points of interest, by digging into people's areas of stay and discovering people's areas of interest. Using this data, they identified densely populated areas of the city. However, such a feature alone does not make any contribution to determining traffic jams. The researchers [3] extracted taxi travel information from GPS data, and used the driving distance, driving time, and average speed of the taxi in the on-load and off-load states to study human travel behavior. However, recommendations for preventing traffic jams and optimizing traffic do not contain mathematically based evidence or solutions.

Studying the trajectory characteristics and behavioral habits of moving objects through time clustering is a very effective clustering method in the trajectory clustering algorithm. The authors of [4] propose a recursive optimization method that updates the manipulation trajectory parameters determined by the main time segment variable (PTVs) in the multistage feature batch processing process. Therefore, this approach is not particularly used by many researchers. Calculating the average value of a time series set based on this distance (which is required for the clustering method) has become a challenging problem. The paper [5] proposed three time series fuzzy clustering methods based on DTW distance. And [6] proposed four multivariate time series clustering models and four variants in time behavior management. The data used in these methods are not standardized in structure and have not been filtered, therefore the corresponding studies seem impractical.

Many studies have recognized the importance of local patterns of moving objects. To this end, many new trajectory segmentation algorithms are proposed, and some trajectories are grouped. [7] provides a partition and group framework for trajectory clustering. In view of the problem of frequent pattern mining of indoor trajectories, a meshing method based on vertical projection distance and a frequent pattern mining algorithm for trajectories based on fuzzy grid sequence are proposed [8]. The fine details of segmentation are presented, and the processes of separation and grouping are described. However, the generalization is made in general terms, which does not define each specific case individually. In order to study the congestion characteristics of the same road network in the city, [9] proposed a method for determining congestion clusters, and to what extent these clusters can represent the congestion level of the entire road network. The results obtained show the possibility of their use for traffic forecasting and estimation, however, these results were obtained only for one type of Munich network and perhaps for other traffic patterns the results will be very different than those shown in this work. Considering that the current GPS positioning accuracy cannot reach the lane level, [10] proposed a traffic congestion detection method based on the GPS trajectory of a taxi when turning. Congestion and other types of interference are detected at a detailed level; however, it is important to note that in cases where the Internet or other traffic monitoring devices are disconnected, the approach proposed in this work will not be of much benefit. The paper [11] uses a weighted exponential moving average of measured GPS speed to estimate traffic congestion. The main advantage of this approach is the simplicity and clarity of the system, however, the main and biggest drawback of this work is taking into account human perceptions, which is not an objective assessment in the analysis of such important information as traffic congestion; such information is more subjective. [12] describes a method based on adaptability. In this work, periodic GIS and GPS data are used for accounting. All traffic flow states are ultimately assessed over sliding time periods. This work showed good results, but again, without using the network, this system will be inoperative. An improved method for predicting the state of urban traffic using time-varying and spatial change information should be based on a three-dimensional traffic flow model. In the literature, in order to improve the estimation accuracy, the average speed of the road section is calculated from multi-source traffic data to estimate the traffic state [13]. This work successfully combines 3 models for organizing and analyzing traffic, however, the complexity and sometimes the imposition of complex mathematical formulas complicates the understanding of this approach. [14] used various attributes such as density, speed, inflow, and past state to estimate the congestion status of the area. Test calculations were performed for such large cities as Shanghai and Beijing and excellent results were obtained for data analysis. However, for a more detailed analysis, important attributes such as time and weather are missing, which have been shown and taken into account in other works. [15] shows a new method for parameter calibration and taxi penetration through the dual-fluid model (TFM) of urban traffic, which only uses GPS data to detect urban traffic congestion.

The review of scientific works suggests that it is advisable to conduct research to determine an algorithm that is efficient in terms of time and data processing for the analysis and study of urban road networks and data on traffic trajectories.

There are many trajectory clustering algorithms available, but there is a great need to reduce the time spent on calculations in real time. The reviewed literature sources do not analyze the effectiveness of methods for calculating loaded points. Available sources do not provide detailed information comparing the model with real city data.

3. The aim and objectives of the study

The aim of the study is to identify an algorithm that is efficient in terms of time and data processing for the analysis and study of urban road networks and trajectory data.

To achieve this aim, the following objectives are accomplished:

– to study algorithms for clustering trajectories of the urban road network;

– to calculate the time intended for data processing;

– to study the effectiveness of the method for calculating congestion points;

– using the example of the city of Chengdu, to build a model of urban road networks and trajectory data.

4. Materials and methods

4. 1. The object and hypothesis of the study

The object of this study is the process of identifying congested areas using an algorithm for calculating the number of clusters based on grid centroids.

The hypothesis of this study was defined as follows: it is possible to quickly and efficiently identify congested areas of city roads, regardless of the size of the area. For this, the following methods and algorithms were used:

– Growth Curve Model is a statistical technique used to analyze longitudinal data, commonly employed in fields such as biology, economics, and psychology. It aims to understand the growth trajectory of a variable over time. In essence, it assumes that the observed data points are the result of an underlying continuous process of growth or change.

– Elbow Method is a heuristic used for determining the optimal number of clusters in a dataset for clustering algorithms, particularly in K-means clustering. Clustering algorithms aim to partition data points into groups or clusters based on their similarities.

– X-Means Algorithm is an extension of the K-means clustering algorithm that automatically determines the optimal number of clusters. Traditional K-means requires the user to specify the number of clusters in advance, which can be challenging when the true number of clusters is unknown.

– SPGK-Means is a variant of the traditional K-means clustering algorithm that operates on Spark Parallel clusters rather than Euclidean clusters. In standard K-means, clusters are defined by their centroids, which are the means of the data points assigned to each cluster. However, in high-dimensional spaces, Euclidean distances can become less meaningful due to the curse of dimensionality.

– RPKM4 is a clustering algorithm designed to address some of the limitations of traditional K-means clustering, such as sensitivity to initialization and outliers.

– K-Means++ is an initialization technique for the K-means clustering algorithm, designed to improve the quality and convergence speed of the resulting clusters. Traditional K-means relies on random initialization of cluster centroids, which can lead to suboptimal clustering solutions, particularly when clusters are poorly initialized.

By comparing and combining technologies, a comprehensive result is obtained that is efficient in terms of execution time and number of calculations. It is believed that the combinations of two different technologies used can be expanded to even larger volumes of data.

4. 2. Algorithm for determining the number of clusters in data sets based on the grid centroid

The proposed algorithm is based on the elbow method and clustering algorithm. The core idea of the elbow method is that the K-means clustering algorithm has different *SSE* values for different k values. When the k value is less than the number of actual clusters, the *SSE* value of the k value differs greatly from the standard *SSE* value (the sum of squared errors of the number of actual clusters). When k is greater than or equal to the number of actual sample clusters, the *SSE* value converges rapidly and is close to the standard *SSE* value. Therefore, by taking different k values for many times, we can get the corresponding *SSE* value relationship graph and the k value corresponding to this elbow is the real cluster number of the data. The core index of the elbow method is *SSE* (sum of squared errors):

$$SSE = \sum_{i=0}^{k} \sum_{p \in c_i} \left| p - m_i \right|^2, \tag{1}$$

where p – point coordinate, m_i – nearest centroid.

Although the elbow method can effectively find the number of clusters in the data set, each k value needs to be run multiple times, and the more times, the more accurate. In general, each k value needs to be run more than 10 times. For K-means clustering based on Euclidean distance, the calculation amount of this method is very large. In order to determine the number of clusters in an unknown dataset, this work uses a grid-based centroid division method to discover the number of clusters in an unknown dataset. The core idea of the algorithm is to divide the entire dataset into multiple large grid regions and calculate the centroids of each grid, combine the centroids that are closer (we think these centroids belong to the same sample), and the final number of centroids is the number of sample clusters. Before giving the algorithm steps, we first give the relevant definitions.

Grid density: in a dataset $D\{d_1, d_2, d_3, ..., d_n\}$, all data points are mapped to a spatial grid $G\{g_1, g_2, g_3, ..., g_n\}$ that can cover all data points. If any grid g_n in G has a grid length of l and data points of n, the grid density is:

$$d_{ensity} = \frac{n}{l^2}. \tag{2}$$

Centroid: in grid G, for any grid g_n, the distance center of all data points $|C_i|$ in the grid is called centroid u_i, and the calculation formula is:

$$u_i = \frac{1}{|C_i|} \sum_{x=1, x \in C_i}^{n} x. \tag{3}$$

Weight distance S:

$$S = \frac{1}{d_{ensity}} \sum_{i,j=0, i \neq j}^{k} \left(u_i - u_j \right)^2. \tag{4}$$

Before the calculation of this algorithm, we think that the spatial boundary distance of data samples is known. If the spatial distance of data samples is unknown, the gridding of data points cannot be carried out.

The core idea of the algorithm is that when the distance between two adjacent centroids is close to one unit, the center position between the centroids is obtained through the weight distance. If the density of the area where the center position is located is less than a certain threshold, the two centroids are the centroids of different clusters; otherwise, they are the centroids of the same cluster.

In order to accurately calculate the centroids of all clusters of the special grid, this algorithm adds a quadtree algorithm to split the grid and then recalculates the centroids of each grid after splitting.

The pseudo code for calculating the number of unknown dataset clusters is as follows (Fig. 1).

Input: Unknown sample number dataset Data
Output: Center of mass and k
1: centroids $c_i \leftarrow g_i$; $g_i \leftarrow D_{ata}$
2: **While** g_i hasn't been marked
3: **foreach** calculate $c_i \leftarrow g_i$
4: **if** $c_i \neq \varnothing$, center$\leftarrow c_i \cup c_{i+1}$
5: **if** center$\neq \varnothing$0 c_i, $c_{i+1} \in$ cluster i
6: **else** quadtree $g_{i\in(1,2,3,4)} \leftarrow g_i$, update $c_i \leftarrow g_i$
7: **else** c_i, $c_{i+1} \notin$ cluster i
8: **endfor**

Fig. 1. Grid centroid-based algorithm for determining the number of clusters in data sets

The steps of the algorithm to determine the number of clusters of an unknown data set are as follows:

1) the data must be fitted to the corresponding large grid and the centroid of each grid is calculated;

2) any unlabeled centroid is then selected and marked. The density of the radius *r* of the region where the centroid is located corresponds to the density threshold. If there is a match, you must go to step 4, otherwise, go to step 3;

3) next, the mesh is split using the quadtree algorithm to obtain four new meshes and recalculate their centroids, and then proceed to step 2;

4) the distance between the center of mass and the adjacent center of mass is calculated and the area of the central position through the weight distance is obtained to determine whether the central position meets the density threshold. If so, the two centers of mass are marked as belonging to the same cluster; otherwise, it is marked as a different cluster centroid;

5) step 2 is repeated until all centroids have been passed;

6) the distance between the weights of centroids marked as the same cluster and the center point is calculated. The number of the obtained central points is equal to the number of clusters. The center is located at the initial cluster center selected using K-means.

4. 3. Grid-based K-means clustering

The grid-based K-means algorithm is superior to the K-means algorithm in clustering with large amounts of data because it reduces the number of cluster points and selects the cluster center points (the selected initial points of clustering are all at the center of the cluster). The grid-based K-means clustering algorithm improves the clustering execution time greatly while guaranteeing good clustering results.

Throughout the clustering process, the entire dataset sample is mapped to a grid and the time to get the grid density is O(d), the time to compute the center point of each grid is O(n), the time to reach the cluster center is O(kn), and the time to reclassify the cluster center is O(n). Assuming the number of iterations is m, the time spent in clustering is O(d+n+m ((k+1)n))). Since the time required for the whole clustering process is related to the number of data samples, the time complexity of other clustering processes is independent of the number of data, only the number of grids. Because the number of grids is much smaller than the number of data samples, the grid-based K-means clustering algorithm is much better than the K-means clustering algorithm in clustering efficiency.

The pseudocode for the grid-based K-means parallel clustering algorithm is as follows (Fig. 2).

Input: Track data collection Data
Output: Cluster centers and grid collections of the cluster centers to which they belong
1: center $g_i \leftarrow g_i$; $g_i \leftarrow D_{ata}$
2: **While** $g_i \neq \varnothing$
3: choose k and centroids c_i
4: **repeat**
5: $c_i \leftarrow g_i$
6: update SSE
7: **until** SSE not change

Fig. 2. Grid-based K-means parallel clustering algorithm

4. 4. Congestion point calculation

Next, it is necessary to calculate congestion points and overloaded traffic areas. Compared with the normal driving traffic section, the traffic in the area where the congested area is located is slow, which leads to an increase in the number of track points in the area and a slow speed. Therefore, the grid density of congested areas is larger, and the density of non-congested areas is smaller. Suppose the density of the grids of the two adjacent areas is G_i, G_{i+1} and there is always $G_{i+1} > G_i$. So we have:

$$k = G_{i+1} - G_i, \tag{5}$$

when k is greater than the threshold value ϑ we set, the area is a suspected source of congestion (congestion point).

$$lk = G_{i+1} - G_i. \tag{6}$$

In equation (5), we use G_i instead, and change the congestion calculation from the density difference to the multiple difference relationship. Therefore, the above formula is changed to:

$$k = \frac{G_{i+1} - G_i}{G_i}. \tag{7}$$

The number of trajectory points of our statistical grid is within a certain period of time (t_i, t_j). If the number of trajectory points per unit time of G_i is x_i, and the number of trajectory points per unit time of G_{i+1} is y_i, then:

$$lk = \frac{\sum_{t_i}^{t_j} y_i - \sum_{t_i}^{t_j} x_i}{\sum_{t_i}^{t_j} x_i}. \tag{8}$$

We assume that the average speed of the vehicle (t_i, t_j) over a period of time is v' and the length of the grid is c. From the above formula, we already know that the number of trajectory points through the regional grid is:

$$s = \sum_{t_i}^{t_j} x_i. \tag{9}$$

The number of trajectory points left through an area grid is $n = c/s$. The total number of trajectories left over a period of time (t_i, t_j) is:

$$s = \frac{c\left(t_j - t_i\right)}{\bar{v}}. \tag{10}$$

Therefore, the average speed of the regional grid G_i and G_{i+1} that we can obtain is $\bar{v}_i$, $\bar{v}_{i+1}$:

$$\bar{v} = \frac{c\left(t_j - t_i\right)}{\sum_{t_i}^{t_j} x_i}, \ \bar{v}_{i+1} = \frac{c\left(t_j - t_i\right)}{\sum_{t_i}^{t_j} y_i}. \tag{11}$$

From the above formula, we can conclude that the more trajectory points in the regional grid, the smaller the average speed. Therefore, we determine whether an area is a traffic jam point and meet the following conditions:

$$k=\frac{\sum_{t_i}^{t_j} y_i-\sum_{t_i}^{t_j} x_i}{\sum_{t_i}^{t_j} x_i}>\vartheta, \overline{v}=\overline{v}_{i+1}-\overline{v}_i>\vartheta'. \quad (12)$$

In a certain section of the road, if there is a traffic jam in a certain place, then the front and rear of the congested section are non-congested sections. Therefore, the distance between two adjacent congestion points within a certain range is the congestion area. Let $A(A_x, A_y)$, $B(B_x, B_y)$ be two adjacent congestion points, then the area between A and B is the congestion area. The calculation formula is:

$$l>\sqrt{\left(A_x-B_x\right)^2+\left(A_y-B_y\right)^2}, \quad (13)$$

where L is the set maximum congestion length, and the value is 2 km. The calculation method of the discovery rate C and the accuracy rate V is given below:

$$c_{disc}=\frac{N_{accu}}{N_{all}}, \quad (14)$$

$$c_{accu}=\frac{N_{accu}}{N_{find}}, \quad (15)$$

where N is the number of discoveries, M is the total number of discoveries, and X is the number of correct discoveries.

4. 5. Application of an algorithm for determining the number of clusters in a data set based on the grid centroid

The K-means clustering algorithm needs to determine the number of clusters clustered in advance, so in an unknown data set, the selection of cluster values has a great impact on the clustering effect. At the same time, due to the random selection of initial points, the K-means clustering algorithm exists. The distribution map of random samples is shown in Fig. 3.

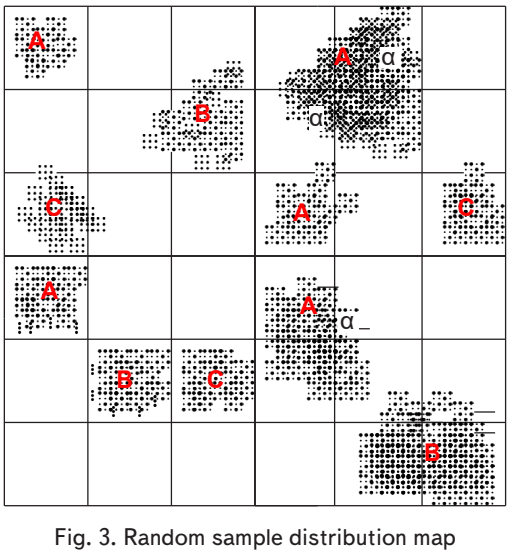

Fig. 3. Random sample distribution map

In the local optimal solution problem, it is necessary to calculate the *SSE* value of the clustered data set multiple times and take its minimum value as the optimal solution, although this value is difficult to prove as the optimal solution. The core index of the elbow method is the sum of squared errors:

$$SSE=\sum_{i=0}^{k}\sum_{p\in c_i}\left|p-m_i\right|^2. \quad (16)$$

Since the current composition of trajectory data is massive, how to quickly process massive amounts of data has become an important obstacle for researchers to mine and analyze the characteristics of trajectory data.

4. 6. Parallel clustering

To conduct the study, a software and hardware tool for processing big data was used. Spark is a software developed by the Algorithms, Machines, and People Lab of the University of California, Berkeley, which can be used to build large-scale, low-latency data analysis applications. Spark is a big data parallel computing framework based on in-memory computing. The following is the big data processing tool as shown in Fig. 4.

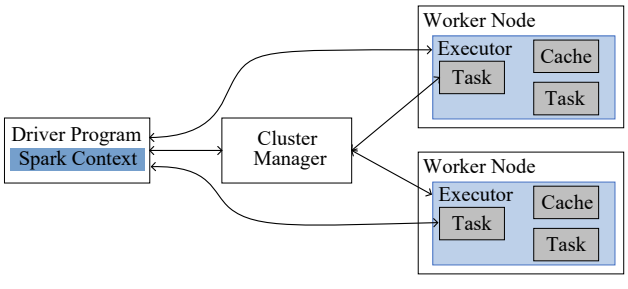

Fig. 4. Spark is a big data processing tool

Further in the work, big data processing methods were used. The algorithm for determining the number of data set clusters based on the grid centroid. First, the initial clustering center point is selected according to the divided class, and then the data sample is divided into the nearest initial center point according to certain distance calculation rules, and then the data points are iteratively reset according to the pre-set heuristic algorithm until the sum of squares of the error is the smallest. So far. The most typical distance-based algorithm is K-means [16]. The algorithm first randomly selects a cluster center point, calculates the distance from all sample data to the cluster center, divides the data into the shortest distance center point, and then recalculates the cluster of each cluster. Typical clustering algorithms based on division are K-means++ described in [17], K-Mediods given in [18] and CLARANS [19], etc.

5. Results of research into traffic congestion management based on big data

5. 1. Research on the urban road network trajectory clustering algorithm

5. 1. 1. Clustering algorithm

The clustering algorithm is an important method of data mining and analysis. Its core idea is to classify the same characteristic data, realize the similar data into a cluster, and separate different data in order to find a specific rule. Through the clustering algorithm, massive data can be gathered and classified purposefully and quickly. At present, the clustering algorithm has been widely used in big data mining analysis, medicine, economy, urban construction, etc. Common clustering

algorithms can be roughly divided into the following four categories according to their clustering methods.

5. 1. 2. Density clustering

There are multiple samples in a dataset, and these different data samples are distributed in different areas of space. The density of the region where the samples are located is large, and the noise data is not even sparse without the spatial density of the region where the data points are located. Therefore, it is very effective to separate each sample of the data set by density clustering. The density clustering algorithm has many methods, among which DBSCAN [19] is one of the most typical density-based clustering methods. The core idea is to arbitrarily select a point of data sample, search for other points within the set area (r=eps) with this point as the center, and if the track points counted in this area meet a set parameter (MinPts), then identify all points in this area as core points. Iterate through all data points that are not accessed until all core areas are found. If there are two high-density points (core points) connected, merge them; if the non-core point is also in the connected area of the core point, add the low-density point (non-core point) to the high-density point (core point). In addition to the DBSCAN algorithm, there are density-based clustering algorithms such as OPTIMS and DENCLUSE [20]. The DBSCAN clustering process is shown in Fig. 5.

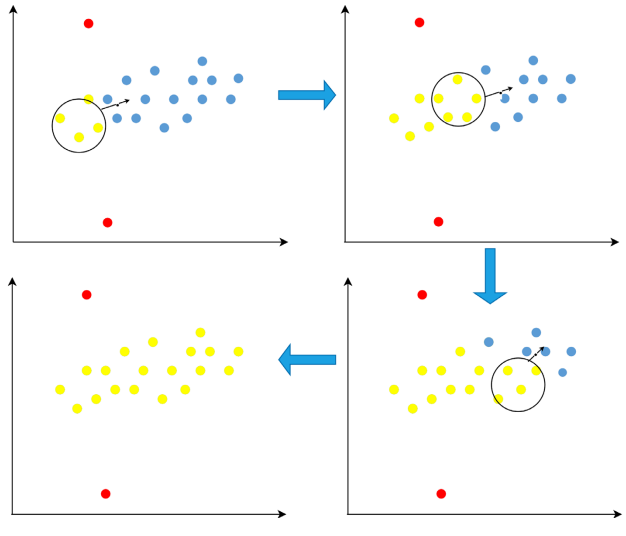

Fig. 5. Diagram of the DBSCAN clustering process

The density-based clustering algorithm has the following advantages due to its own characteristics:

1) clustering is based on the density of the region where the samples are located, so there is no need to determine the number of samples in the data set;

2) because it is based on the attribute clustering of the sample itself, the problem of local optimal solution of clustering effect faced by distance-based clustering is well avoided, and arbitrary clusters can be found;

3) because it is based on local density clustering and has continuity, it can well filter out the interference of noise to the algorithm.

Although the density-based clustering algorithm has the above advantages, it also has the following disadvantages due to the limitations of the algorithm:

1) since the establishment of the center point is random, and two important parameter search ranges (eps) and minimum density values (MinPts) are considered to be set, it is easy to cause human errors, and the accuracy of the algorithm depends on experience;

2) because of the minimum density threshold of the core point, the clustering effect is not ideal for the samples with uneven density distribution.

5. 1. 3. Grid clustering

The grid-based clustering method is a spatial location mapping method, which maps the multidimensional data grid to the cell grid distributed by independent objects through a specific method. The grid-based clustering method uses a multi-resolution network data structure. It takes the mapped data as the cluster object to form a grid data structure, and then all clustering will take the grid as the cluster object. Typical grid clustering algorithms include STING, CLIQUE, Wave Cluster, etc. The grid clustering process is shown in Fig. 6 below:

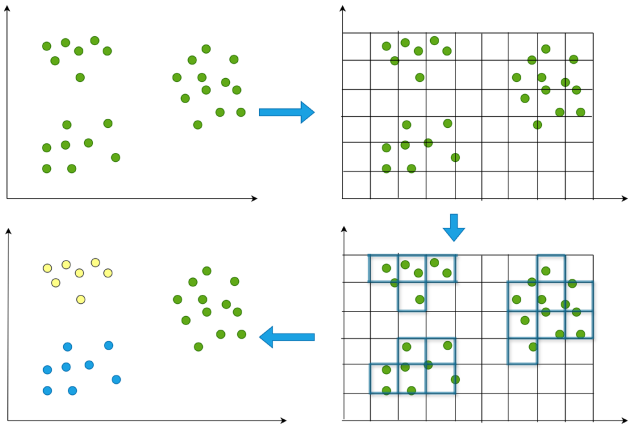

Fig. 6. Schematic diagram of the grid clustering process

The clustering method has the following advantages:

1) it is fast, and the computational complexity is only related to the number of grids and has nothing to do with the size of data;

2) strong scalability for data of different stages;

3) strong anti-interference for noise data.

Disadvantages:

1) the grid division is artificially set and the unit length of the grid is usually large, so the clustering accuracy is poor compared with other clustering algorithms;

2) it is difficult to deal with data samples with uneven density distribution;

3) data sample boundaries cannot be found accurately.

5. 1. 4. Partition-based clustering

The basic principle of clustering based on partition is to divide the data samples to be classified according to different partition requirements, so as to achieve the goal that similar samples are close to each other and different data are far away from each other. For partition-based clustering, first select the initial cluster center point according to the divided class, then divide the data sample to the initial center point of the nearest distance according to certain distance calculation rules, and then perform iterative relocation for the data points according to the pre-determined heuristic algorithms until the sum of squares of errors is the minimum. The most typical distance-based algorithm is K-means. The algorithm firstly selects N cluster centers randomly, calculates the distance from all sample data to the cluster center, divides

the data to the center point of the shortest distance, then recalculates the cluster center of each cluster and calculates the distance from all sample data to the cluster center again, and iterates repeatedly until the change of the cluster center is less than a set threshold. Typical partition-based clustering algorithms include K-means++, K-Mediods and CLARANS. The division and clustering process is shown in Fig. 7.

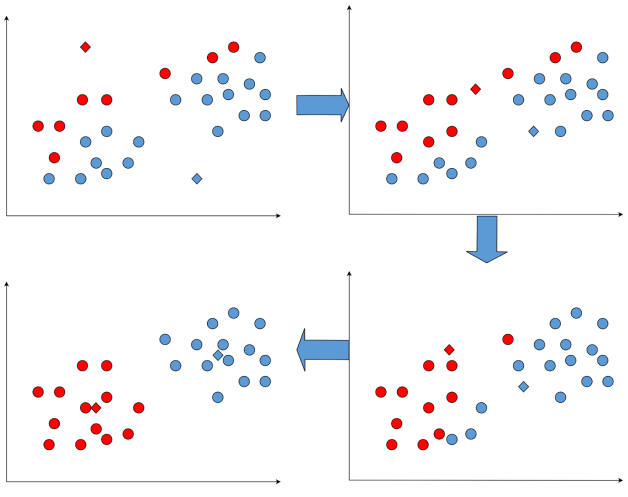

Fig. 7. Schematic diagram of the partition-based clustering process

5. 1. 5. Hierarchical clustering

The idea of hierarchical clustering is to aggregate or segment data samples layer by layer to achieve the purpose of merging or segmentation of data samples. Hierarchical clustering mainly has two types: merged hierarchical clustering and split hierarchical clustering. Merged hierarchical clustering is a bottom-up clustering algorithm. Starting from the lowest level, each time the most similar data sample points are merged to form the upper level clustering. The whole process is until all data samples are merged into one or meet some set conditions. Split hierarchical clustering is a bottom-up method, starting from a whole data sample and only dividing the data layer by layer until all data points are divided into independent units or meet certain segmentation conditions. Classical hierarchical clustering algorithms include BIRCH, CURE, ROCK, etc. Fig. 8 shows the hierarchical clustering process.

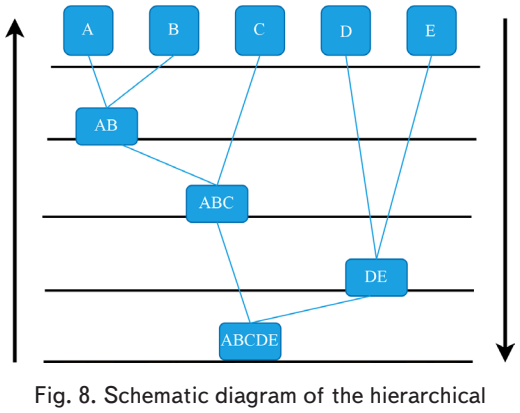

Fig. 8. Schematic diagram of the hierarchical clustering process

The advantages of this clustering algorithm are:

1) distances and rules are easy to define with few restrictions;

2) no need to know the number of samples;

3) you can find the hierarchical relationship of classes;

4) can be clustered into other shapes.

Disadvantages:

1) high computational complexity;

2) singular values can also have a great impact;

3) the algorithm is likely to cluster into chains.

The diagram of the elbow method to determine the clustering value is shown in Fig. 9. From Fig. 9, we can conclude that the clustering value of the elbow position is 4. Therefore, for the clustering of this data set, the best number of clusters should be 4.

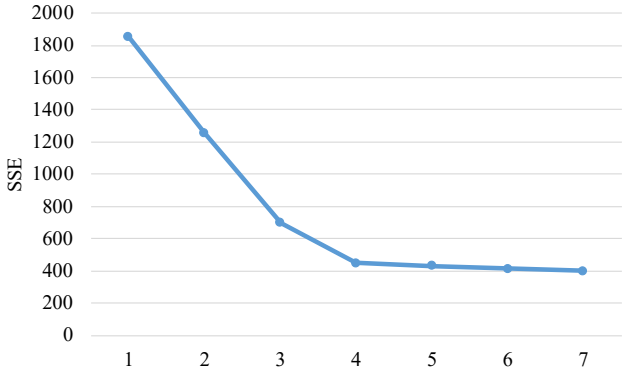

Fig. 9. Schematic diagram of the elbow method

The grid-based K-means algorithm has a better clustering effect than the K-means algorithm under the condition of large amounts of data. The reason is that the algorithm reduces the number of clustering points and the initial clustering points selected at the selected clustering center point are all in the center of the cluster. Under the premise of ensuring a good clustering effect, the grid-based K-means clustering algorithm has greatly improved the clustering execution time. The steps are as follows:

1) divide the grid size according to the sample distance and accuracy requirements;

2) map the data samples into the grid, obtain the center point of each grid, and calculate the density of each grid;

3) select the center of the initial cluster sample and calculate the Euclidean distance from each grid to the center of the cluster;

4) divide the grid according to the cluster center to obtain a new cluster;

5) calculate the weight Euclidean distance from each grid sample point to the center of the cluster, and re-divide the center of the cluster;

6) iterate until the distance between the weights of all grids to the corresponding cluster center is the smallest.

The core idea of the algorithm for clustering trajectories of urban road networks based on grid density is next: The urban road network trajectory clustering algorithm based on grid density first maps the trajectory data into the grid, and the grid size is set to be about 50 meters, the same as the width of the urban road. After that it counts the density of all grids, removes the grids with the smallest density, and arbitrarily selects one of the grids as the starting point (core point) of the cluster; further, it searches for the neighborhood grid of the core point, and selects the grid with the largest density in the neighborhood grid as the core point of the next cluster until all grids are traversed.

The urban road network trajectory clustering algorithm based on grid density is a grid-based clustering algorithm

that can automatically and quickly discover popular traffic sections in the city. In order to discover the congestion points of urban road sections, this work divides popular road sections into segments of length in grids, as shown in Fig. 10.

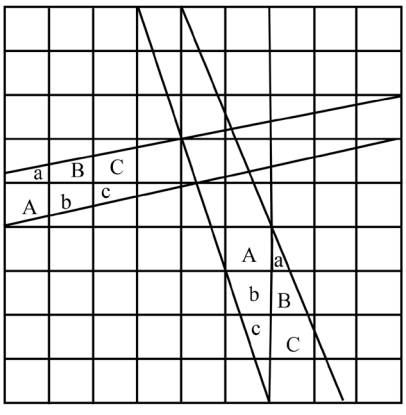

Fig. 10. Schematic diagram of road section division

From Fig. 10, we assume that *A*, *B*, and *C* are the clustering center points of the grid density-based urban road network trajectory clustering algorithm proposed in this paper, and points *a*, *b*, and *c* are the non-clustering center points. In order to obtain the complete area density of the road segment, so as to accurately calculate the difference of the trajectory density between adjacent road segments, it is necessary to assign the grids represented by *a*, *b*, *c* to the cluster center grid. We set the area where *A* is located as area *A*. From the figure, we obviously find that *a* should belong to area *A*. Similarly, *b* belongs to area *B*, and *c* belongs to area *C*. In order to achieve a reasonable allocation of non-cluster center points to cluster center points, this work calculates the angle between the coordinate axes between *A*, *B*, and *C*:

$$\theta = \frac{\sin^{-1^{(y1-y2)}}}{(x_1 - x_2)}, \tag{17}$$

to assign non-core grids *a*, *b*, *c*.

The algorithm for determining the number of data set clusters based on the grid centroid works as follows. In order to verify the effectiveness and efficiency of the data set cluster number determination algorithm based on the grid centroid proposed in this paper, the following experimental scheme is designed for comparison. First of all, in order to verify the effectiveness of this algorithm, the synthetical data set and the data set in the UCI library were used for experiments. Some samples of the data set are shown in Fig. 11.

In this experiment, in order to ensure the accuracy of the experimental results, the Method for Determining the Number of Clusters Based on Grid Centroid (hereinafter referred to as GCM algorithm) is compared with the elbow method and the *x*-means algorithm proposed in the literature [20]. Among them, the elbow method and the *x*-means algorithm are executed more than ten times for each K value and the optimal SSE is selected as the experimental data, and the *K* value is selected from 2. The experimental results are shown in Fig. 12.

From the experimental results in Fig. 11, we can conclude that in the elbow method and the *x*-means algorithm polyline, $k=3$ decreases rapidly, so we can conclude that the number of clusters of the data sample is 3. The GCM algorithm intersects with the elbow method and the polyline of the *x*-means algorithm when the value is equal to 3. Therefore, it can be concluded that the number of clusters of data samples obtained by the GCM algorithm is accurate, which verifies the effectiveness of the GCM algorithm.

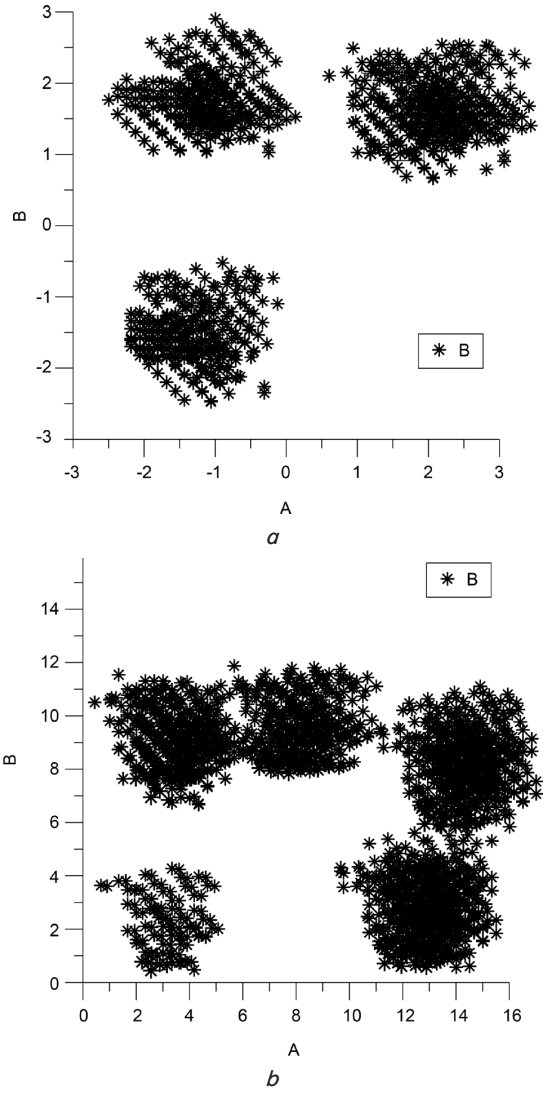

Fig. 11. Partial data sample distribution map: *a* — synthetical dataset; *b* — UCI library dataset

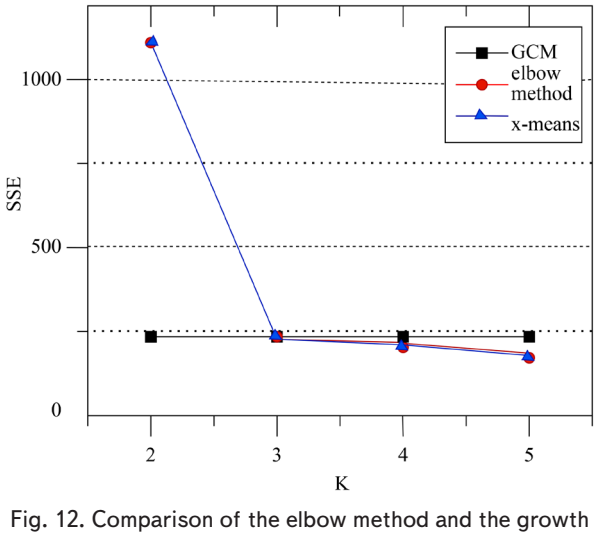

Fig. 12. Comparison of the elbow method and the growth curve model algorithm

5. 2. Calculation of time intended for data processing

A comparison of the methods was also carried out in terms of the time spent for calculations. As a result, the results were obtained according to Fig. 13.

The time allocated for data processing has been calculated. An increase in the volume of processed data showed an almost stable execution time t=3 s for the GCM algorithm for data with a volume of up to almost 2,000 units. While the elbow method and x-means grew all the time as the volume of data increased during runtime. For example, to process data of 2,000 units, x-means spent 30 times, and the elbow method 40 times more time.

Grid-based K-means Parallel Clustering Algorithm works as follows. To prove the effectiveness of the Grid K-means clustering algorithm based on Spark parallel computing (hereinafter referred to as the SPGK-means clustering algorithm), an experiment was conducted.

The trajectory big data provided by Didi Company is used as the experimental data set. We compare the efficiency of the algorithm on a single machine Spark and a Spark cluster and compare the accuracy of K-means++ and the algorithm in this paper through data visualization [21]. Fig. 14, 15 show the experimental comparison between SPGK-means and the RPKM (recursive partition-based K-means) and K-means++ algorithms proposed in the literature [22].

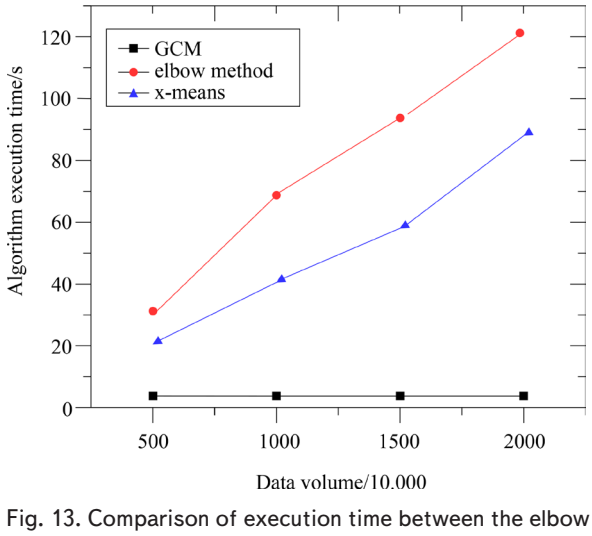

Fig. 13. Comparison of execution time between the elbow method and the growth curve model algorithm

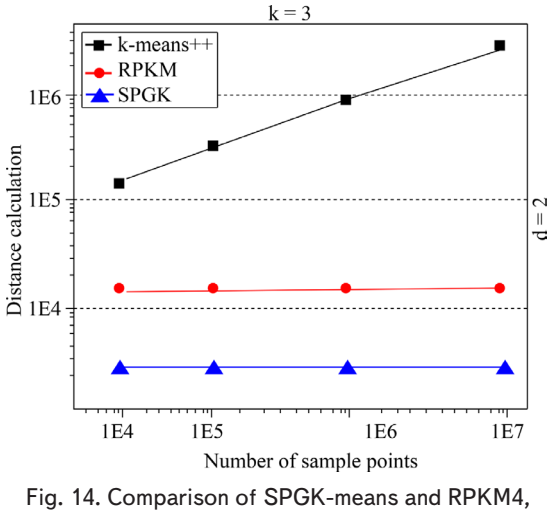

Fig. 14. Comparison of SPGK-means and RPKM4, K-means++ calculation amount

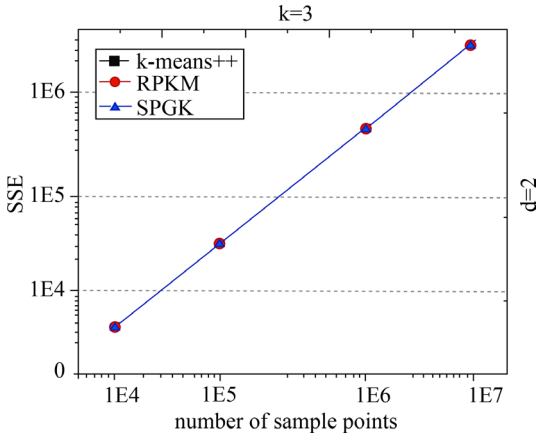

Fig. 15. Comparison of calculated distance and SSE between SPGK-means and RPKM4, K-means++

An experiment was conducted with data from the UCI machine learning database. Each algorithm was run 10 times and averaged.

This experiment showed the effectiveness of SPGK-means for different values of the number of points. RPKM4 consistently calculated 10 times more than SPGK-means, and K-means++ 10 times more than RPKM4 at 10^4, and with an increase in the number of points it also showed an increase in calculations. The use of a parallel cluster reduced the number of calculations by more than 100 times.

However, the SSE values for all methods coincided with an error of 0.0001.

Clustering of Urban Road Network Trajectories Based on Grid Density. Fig. 16, 17 are the Chengdu traffic road network model obtained by the neighborhood grid maximum density clustering algorithm in this paper.

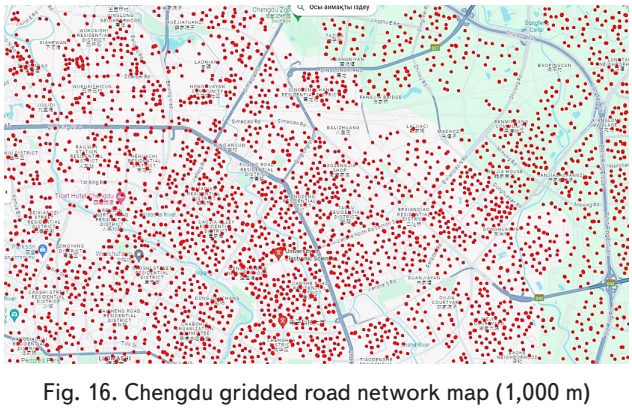

Fig. 16. Chengdu gridded road network map (1,000 m)

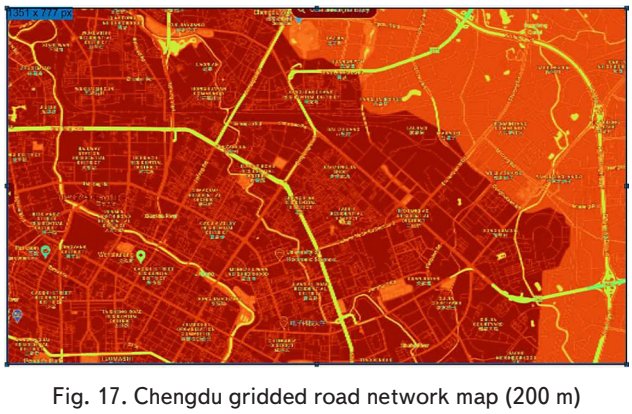

Fig. 17. Chengdu gridded road network map (200 m)

Because the grid is relatively large, the clustering effect is not very good, and there is a certain deviation between the grid and the road network. This is caused by the maximum error of one grid unit between the track point and the original actual coordinate point after gridding. As the meshing becomes smaller, the neighborhood maximum density algorithm finds the road network more and more accurately. Fig. 16 is already close to the track point road network.

We propose a novel approach to calculating congestion points and delineating congested areas within urban road networks. This methodology is crucial for understanding traffic congestion patterns and formulating effective mitigation strategies. An algorithm for clustering the trajectory of an urban road network was investigated by comparing this algorithm with other available ones.

5. 3. Results of experimental calculation of congestion points and areas

The method used in this work is to calculate the density difference between regions through the regional density analysis of track points, so as to obtain the traffic congestion points of the urban road network. The traffic congestion point proposed in this research aims to find the area where the congestion source occurs. Through clustering analysis and mining of vehicle trajectory data, we can not only find the location of the urban traffic network congestion source, but also visualize the mining results, so that we can directly see the specific location of the urban road network congestion source area.

This paper calculates and analyzes the geographical location of the origin of congestion to find out the cause of congestion, which is the core part of this paper's analysis of urban traffic congestion. The relevant definitions of this algorithm are given below.

Traffic flow $|s|$: the number of vehicles passing through a certain area in unit time, which is represented by the number of track points $s\{s_1, s_2, s_3, ..., s_n\}$.

Regional average speed $|v|$: the speed $v\{v_1, v_2, v_3, ..., v_n\}$ through a grid in a period of time (t_i, t_j). Then the average speed in this area is calculated by the following formula:

$$\bar{\vartheta} = \frac{1}{n}\sum_{i=1}^{n} \vartheta_i. \quad (18)$$

Regional density $|d|$: A certain area S within a period of time (t_i, t_j), the ratio of the number of track points left by the grid to the area of the region: $|d|=|s|/S$.

Definition of traffic congestion point: suppose a road S is divided into n sections, $S=\{s_1, s_2, s_3, ..., s_n\}$. There are two adjacent points s_n, s_{n+1}, and the regional track points of s_n, s_{n+1} are $|s_n|$, $|s_{n+1}|$ respectively. If there is $|s_n|>k*|s_{n+1}|$ or $k*|s_n|<|s_{n+1}|$, the adjacent points s_n, s_{n+1} are suspected traffic congestion points.

5. 4. Urban road network trajectory clustering based on grid density

This paper presents a novel method for identifying congestion points and areas within urban environments. In addition to determining congestion points, the study calculates the average speed values for both congested and non-congested areas within the identified congestion point discovery area, utilizing a congestion threshold of 5. The results illustrated in Fig. 18 showcase the average speed values on the ordinate axis, measured in kilometers per hour, with each point along the abscissa representing the congestion value of the corresponding congestion point.

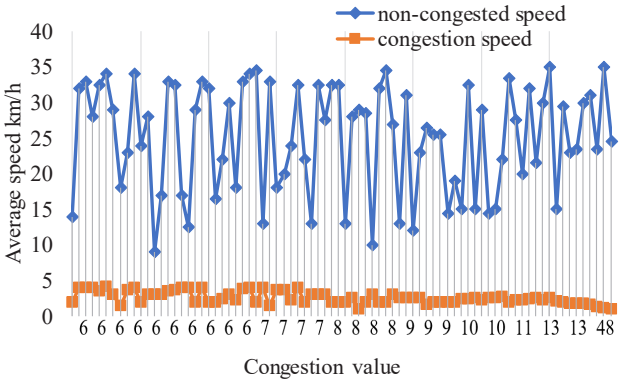

Fig. 18. Average speed in congested and non-congested areas

Notably, the analysis reveals a substantial discrepancy in average speeds between congested and non-congested areas. While the average speed within congested areas remains below 5 km/h, non-congested areas maintain speeds consistently above 20 km/h. This significant contrast underscores the role of adjacent areas as the primary sources of congestion. Consequently, the identified congestion points serve as pivotal indicators for understanding and addressing traffic congestion within urban settings.

As shown in Fig. 19, the traffic network structure of Chengdu can be clearly seen through the trajectory points. Fig. 20, 21 are the traffic network models of Chengdu obtained through the neighborhood grid maximum density clustering algorithm in this paper.

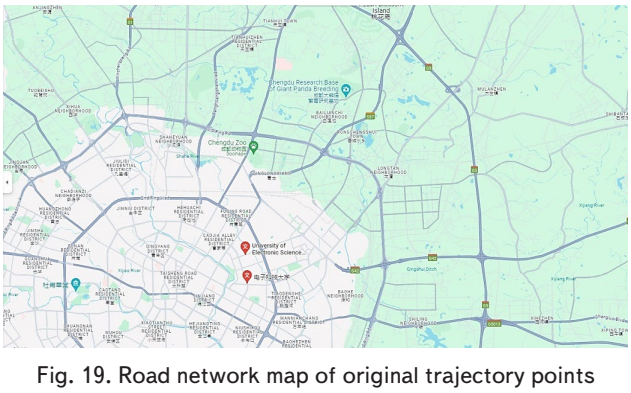

Fig. 19. Road network map of original trajectory points in Chengdu city

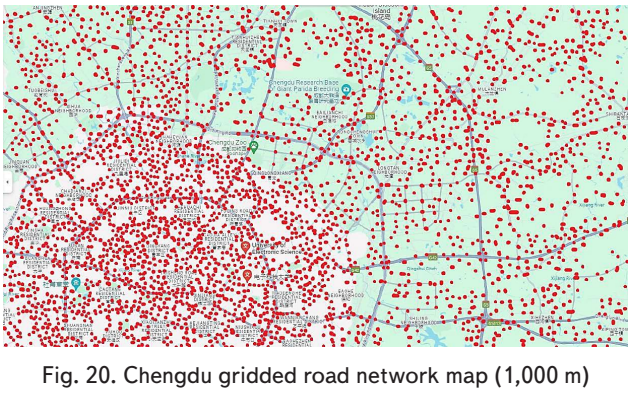

Fig. 20. Chengdu gridded road network map (1,000 m)

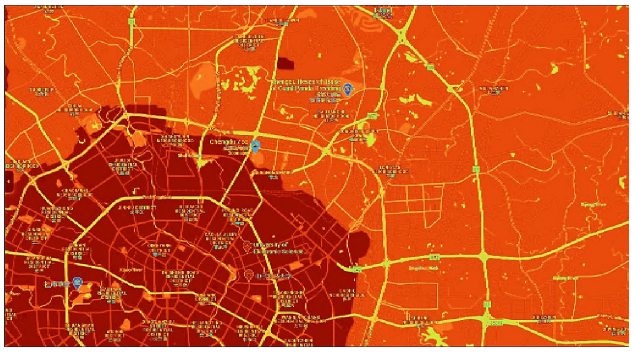

Fig. 21. Chengdu gridded road network map (200 m)

Compared with Fig. 19–21, the neighborhood grid maximum density algorithm can accurately discover the urban road network model. Fig. 20 shows some deviation between the grid and the road network due to the large size of the grid and poor clustering performance. This is caused by an error of up to one grid unit between the trajectory points and the original actual coordinate points after meshing. With the reduction of grid division, the neighborhood maximum density algorithm is becoming increasingly accurate in discovering road networks. Fig. 21 has basically approached the trajectory point road network.

In order to study the congestion situation in cities, this paper proposes an algorithm for discovering traffic congestion based on actual traffic conditions, which is used to discover the areas where urban congestion occurs. This algorithm divides each road into equidistant segments and calculates the ratio of trajectory density difference between adjacent segments. If this value is greater than a certain threshold, the area may be a traffic congestion area. At the same time, we also calculated the average velocity difference of the adjacent segments for verification. However, the presence of traffic lights can affect the accuracy of the sub algorithm's discovery. Therefore, in order to reduce the discovery error of the algorithm, this paper proposes an intersection discovery algorithm. Fig. 22 shows the algorithm recognition diagram for intersections and traffic signal intersections proposed in this paper.

From Fig. 22, we can see that there are many traffic congestion points due to traffic lights and intersections. However, due to the inherent driving characteristics of intersections, even if congestion conditions are met, the area cannot be simply identified as a traffic congestion point. In order to study the actual traffic situation at these intersections, this paper incorporates a calculation method for intersection congestion. From the comparison of the left and right figures in Fig. 22, we can see that this algorithm can effectively identify and eliminate interference from traffic lights and intersections.

The following four figures (Fig. 23, 24) are the visualization display results of the final traffic congestion area discovery algorithm in this paper.

Fig. 23 shows the calculation results for congestion values of 3 and 5, respectively. From Fig. 23, we can see that the congestion points in Chengdu are concentrated on various main roads. Among them, congestion points are relatively concentrated near the roundabout overpass and the bus and train station, while other congestion points are scattered in various popular areas of the city, such as crowded places like People's Park, Kowloon Square, and People's Hospital.

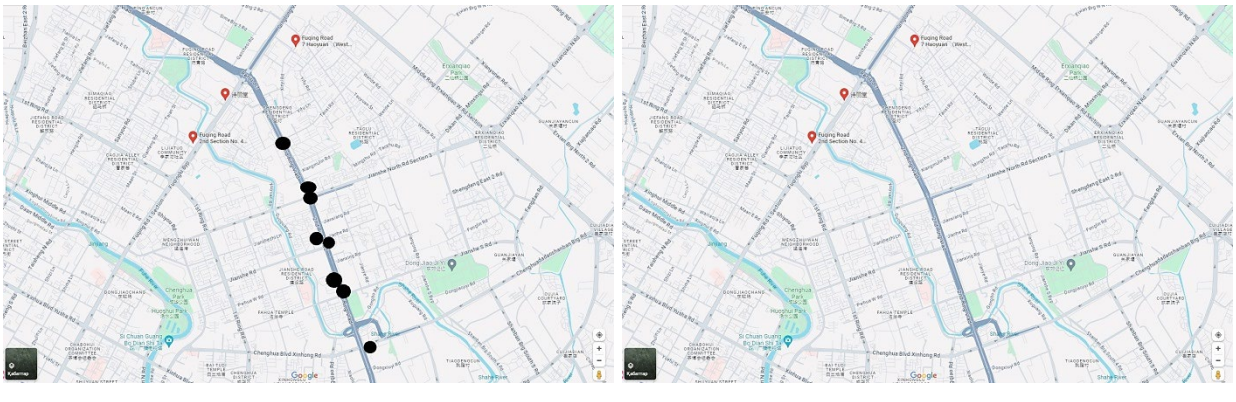

Fig. 22. Identification diagram of intersections and signalized intersections

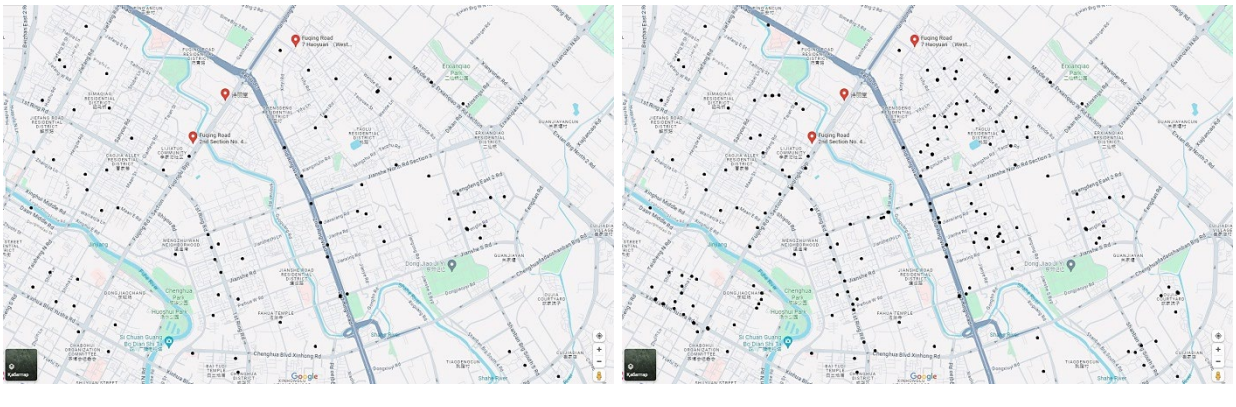

Fig. 23. Congestion points with congestion values of 3 and 5, respectively

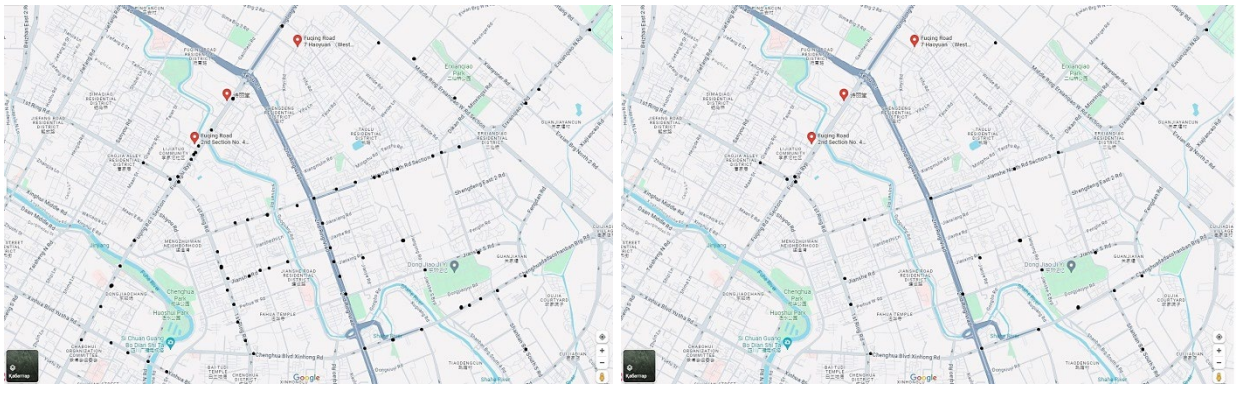

Fig. 24. Congestion points with congestion values of 8 and 10, respectively

In order to verify the accuracy and reliability of this algorithm, this paper has also been validated based on the factual traffic information provided by the Chengdu transportation website. Congestion near the Ring Road Interchange is caused by the need for vehicles to enter and exit the city in high stages of commuting, and the interchange is located at the intersection of vehicles on the main road around the city, which is at the center of traffic and is prone to congestion. Congestion is prone to occur at bus and train stations due to the high flow of cars and people, resulting in traffic congestion. In addition, this paper also found that congestion is also prone to occur in certain locations, for example, congestion often occurs in the streets near schools and hospitals. For example, in the Shazhou Street section near the Chengdu China Railway Second Bureau Hospital, congestion is prone to occur. This paper found that the congestion point is located near the outpatient department of the hospital, and pedestrians passing by are more likely to cause traffic congestion. There are also congestion points near the People's Hospital of Sichuan Province. After analysis, this paper found that the road in this section (West Second Section of the First Ring Road) is funnel-shaped, with one large and one small at both ends of the road, and there are traffic lights at the end of the road, which leads to traffic congestion and the inability to evacuate in a timely manner. The right figure in Fig. 23 shows the traffic congestion points when the congestion value is set to 5. Compared with the left figure in Fig. 23, there is no significant change in this figure. Some congestion concentration points are still near the roundabout interchange, urban main roads, and bus and train passenger stations, with only a few congestion points removed. Fig. 24 shows traffic congestion points with congestion values of 8 and 10, respectively. Fig. 24 fully demonstrates the heavy traffic congestion points in Chengdu, which are prone to congestion in certain areas such as the roundabout overpass, the main railway station road, and other popular road sections.

Therefore, it can be concluded that adjacent regions are the source regions of congestion, which is the congestion point defined in this paper.

6. Discussion of the results of the study of the effectiveness and accuracy of the proposed method

Urban traffic congestion areas can be obtained in many ways, and the most important means at present is to find them by monitoring the traffic flow. Although the accuracy of this method is relatively high, it is difficult to achieve due to the large deployment scope, short monitoring road section, high cost and other reasons. In the aspect of trajectory traffic, mining urban traffic congestion by studying the trajectory clustering analysis of vehicles is a hot spot of mobile trajectory research. Clustering analysis and mining of moving trajectories are carried out, and urban traffic congestion areas are obtained by comparing them with real urban traffic congestion characteristics.

Visualization of calculations carried out using clustering algorithms is presented in Fig. 5–8. The relevance and importance of the analysis of these methods are especially demonstrated in the works [4–8, 10, 13]. It is especially important to note the work of the growth curve model method (Fig. 13), in comparison with the elbow method and the *x*-means algorithm. This algorithm showed $SSE=240$ at values $K=2, 3, 4, 5$, while the elbow method and the *x*-means algorithms at $K=2$ had SSE more than 1,150, at $K=3$ all 3 methods intersected, and at large K values, although the two above methods showed low *SSE* values, their values did not exceed 100 units from the growth curve model indicator.

Time calculation completed (Fig. 12) intended for data processing. An increase in the volume of processed data showed an almost stable execution time $t=3$ s for the GCM algorithm for data with a volume of up to almost 2,000 units. While the elbow method and *x*-means grew all the time as the volume of data increased during runtime. For example, to process data of 2,000 units, *x*-means spent 30 times, and the elbow method 40 times more time.

This experiment (Fig. 12) showed the effectiveness of SPGK-means for different values of the number of points. RPKM4 consistently calculated 10 times more than SPGK-means, and K-means++ 10 times more than RPKM4 at 104, and with an increase in the number of points it also showed an increase in calculations. The use of a parallel cluster reduced the number of calculations by more than 100 times.

Models (Fig. 16, 17) of the Chengdu transport network obtained using a clustering algorithm with the maximum density of the neighborhood grid are presented. There are some deviations (Fig. 19–21) between the grid and the road network due to the large grid size. This error is explained by an error of up to one between the points and the real grid. Only even finer crushing leads to an improvement in the result.

The literature on traffic congestion detection [2] based on cluster analysis of the entire trajectory provides an effective method: to group similar trajectories in a certain period into clusters of each road using clustering methods. Second, for each cluster, the length and angle of all tracks in the group are averaged as representative tracks, i.e. road mode.

Third, we create a feature vector for each road according to the historical traffic conditions of each road and the nature of neighboring roads.

The regional analysis literature [19] based on trajectory points also provides an effective detection method. By dividing the entire urban area into several grids, the density of inflow and outflow paths in each grid, as well as the speed in the grid, are calculated to determine whether the grid area is congested. This method can estimate and predict congestion around the clock, but it cannot judge whether congestion is occurring based only on unit time and average speed. The literature [20] also uses a grid division method to more accurately estimate the jam area. The author divides the detection area into multiple grids and uses multiple weights such as density and speed to estimate traffic congestion and improve detection performance. This work does not take into account the influence of intersections and traffic lights on the algorithm, and there are also shortcomings in traffic congestion estimation and analysis.

The advantage of this study can be considered the experimental, mathematical validity in the calculation of clusters and the location of the calculation grid. Despite its strengths, our study has certain limitations that must be acknowledged. Firstly, the reliance on simulated data may introduce inaccuracies, highlighting the need for validation using real-world data. Additionally, the effectiveness of our methodology may vary depending on factors such as data availability, local traffic conditions, and the quality of road infrastructure. These limitations must be considered when applying our methodology in practice or conducting further theoretical research.

One notable shortcoming of our study is the lack of consideration for external factors such as weather conditions, accidents, and road construction, which can influence traffic congestion. Future research could explore the integration of additional variables to enhance the robustness of our methodology. Additionally, further validation and refinement of the methodology are warranted to address existing limitations and improve its applicability in diverse contexts.

7. Conclusions

1. Algorithms for clustering trajectories of the urban road network were studied. The effectiveness of the Growth Curve Model algorithm has been proven in comparison with the Elbow Method and X-Means Algorithm. When analyzing *SSE* values at various *K* values (2, 3, 4, 5), the Growth Curve Model algorithm demonstrated $SSE=240$. While the Elbow Method and X-Means Algorithm at $K=2$ showed SSE over 1,150. At $K=3$ all three methods matched, however at higher values of *K*, although the Elbow Method and X-Means Algorithm also showed low SSE values, they did not exceed 100 units from the Growth Curve Model algorithm indicator.

2. Calculations were made of the time spent on data processing. As the volume of data processed increased, the execution time of the GCM algorithm remained almost stable, amounting to about 3 seconds for data up to almost 2,000 units. In contrast, the execution time of the Elbow Method and X-Means Algorithm increased with increasing data volume. For example, to process data of 2,000 units, the X-Means Algorithm spent 30 times more time, and the Elbow Method 40 times more.

3. The experiment showed the effectiveness of the SPGK-Means algorithm for various values of the number of points. RPKM4 consistently took 10 times longer than SPGK-Means, and K-means++ took 10 times longer than RPKM4 at 104 points. As the number of points increased, K-means++ also showed an increase in computational cost. The use of a parallel cluster reduced the number of calculations by more than 100 times.

4. Models of the Chengdu transportation network obtained using a clustering algorithm with maximum grid density of neighborhoods were presented. There were some discrepancies between the grid and the road network due to the large grid size. This error is explained by an error of up to 1 between the points and the real grid. Only finer partitioning led to improved results.

Conflict of interest

The authors declare that they have no conflicts of interest in relation to the current study, whether financial, personal, authorship, or otherwise, that could affect the study and the results reported in this paper.

Financing

The study was performed without financial support.

Data availability

All data are available in the main text of the manuscript.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Lu, M., Liang, J., Wang, Z., Yuan, X. (2016). Exploring OD patterns of interested region based on taxi trajectories. Journal of Visualization, 19 (4), 811–821. https://doi.org/10.1007/s12650-016-0357-7
2. Li, T., Wu, J., Dang, A., Liao, L., Xu, M. (2019). Emission pattern mining based on taxi trajectory data in Beijing. Journal of Cleaner Production, 206, 688–700. https://doi.org/10.1016/j.jclepro.2018.09.051
3. Tang, J., Liu, F., Wang, Y., Wang, H. (2015). Uncovering urban human mobility from large scale taxi GPS data. Physica A: Statistical Mechanics and Its Applications, 438, 140–153. https://doi.org/10.1016/j.physa.2015.06.032
4. Liu, X., Luan, X., Liu, F. (2018). Optimizing manipulated trajectory based on principal time-segmented variables for batch processes. Chemometrics and Intelligent Laboratory Systems, 181, 45–51. https://doi.org/10.1016/j.chemolab.2018.08.010

5. Izakian, H., Pedrycz, W., Jamal, I. (2015). Fuzzy clustering of time series data using dynamic time warping distance. Engineering Applications of Artificial Intelligence, 39, 235–244. https://doi.org/10.1016/j.engappai.2014.12.015
6. D'Urso, P., De Giovanni, L., Massari, R. (2018). Robust fuzzy clustering of multivariate time trajectories. International Journal of Approximate Reasoning, 99, 12–38. https://doi.org/10.1016/j.ijar.2018.05.002
7. Lee, J.-G., Han, J., Whang, K.-Y. (2007). Trajectory clustering. Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data. https://doi.org/10.1145/1247480.1247546
8. Wang, L., Hu, K., Ku, T., Yan, X. (2013). Mining frequent trajectory pattern based on vague space partition. Knowledge-Based Systems, 50, 100–111. https://doi.org/10.1016/j.knosys.2013.06.002
9. Rempe, F., Huber, G., Bogenberger, K. (2016). Spatio-Temporal Congestion Patterns in Urban Traffic Networks. Transportation Research Procedia, 15, 513–524. https://doi.org/10.1016/j.trpro.2016.06.043
10. Kan, Z., Tang, L., Kwan, M.-P., Ren, C., Liu, D., Li, Q. (2019). Traffic congestion analysis at the turn level using Taxis' GPS trajectory data. Computers, Environment and Urban Systems, 74, 229–243. https://doi.org/10.1016/j.compenvurbsys.2018.11.007
11. Pattara-atikom, W., Pongpaibool, P., Thajchayapong, S. (2006). Estimating Road Traffic Congestion using Vehicle Velocity. 2006 6th International Conference on ITS Telecommunications. https://doi.org/10.1109/itst.2006.288722
12. Shi, W., Kong, Q.-J., Liu, Y. (2008). A GPS/GIS Integrated System for Urban Traffic Flow Analysis. 2008 11th International IEEE Conference on Intelligent Transportation Systems. https://doi.org/10.1109/itsc.2008.4732569
13. Kong, Q. J., Li, Z., Chen, Y., Liu, Y. (2009). An Approach to Urban Traffic State Estimation by Fusing Multisource Information. IEEE Transactions on Intelligent Transportation Systems, 10 (3), 499–511. https://doi.org/10.1109/tits.2009.2026308
14. Yang, Y., Xu, Y., Han, J., Wang, E., Chen, W., Yue, L. (2017). Efficient traffic congestion estimation using multiple spatio-temporal properties. Neurocomputing, 267, 344–353. https://doi.org/10.1016/j.neucom.2017.06.017
15. Lu, S., Knoop, V. L., Keyvan-Ekbatani, M. (2018). Using taxi GPS data for macroscopic traffic monitoring in large scale urban networks: calibration and MFD derivation. Transportation Research Procedia, 34, 243–250. https://doi.org/10.1016/j.trpro.2018.11.038
16. Hartigan, J. A., Wong, M. A. (1979). Algorithm AS 136: A K-Means Clustering Algorithm. Applied Statistics, 28 (1), 100. https://doi.org/10.2307/2346830
17. Arthur, D., Vassilvitskii, S. (2007). K-means++: the advantages of careful seeding. SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 1027–1035.
18. Cordeiro de Amorim, R., Mirkin, B. (2012). Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. Pattern Recognition, 45 (3), 1061–1075. https://doi.org/10.1016/j.patcog.2011.08.012
19. Ng, R. T., Han, J. (2002). CLARANS: a method for clustering objects for spatial data mining. IEEE Transactions on Knowledge and Data Engineering, 14(5), 1003–1016. https://doi.org/10.1109/tkde.2002.1033770
20. Shahbaba, M., Beheshti, S. (2012). Improving X-means clustering with MNDL. 2012 11th International Conference on Information Science, Signal Processing and Their Applications (ISSPA). https://doi.org/10.1109/isspa.2012.6310493
21. Abdiakhmetova, Z. M. (2017). Wavelet data processing in the problems of allocation in recovery well logging. Journal of Theoretical and Applied Information Technology, 95 (5). Available at: https://www.kaznu.kz/content/files/news/folder23320/2017%20%D0%A1%D0%BA%D0%BE%D0%BF%D1%83%D1%81%207Vol95No5.pdf
22. Turken, G., Pey, V., Abdiakhmetova, Z., Temirbekova, Z. (2023). Research on Creating a Data Warehouse Based on E-Commerce. 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST). https://doi.org/10.1109/sist58284.2023.10223542