Expert Systems with Applications 37 (2010) 1169–1178

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A comparison of outlier detection algorithms for ITS data

Shuyan Chen [a,b,*], Wei Wang [a], Henk van Zuylen [b]

[a] Transportation College, Southeast University, 210096 Nanjing, China
[b] Civil Engineering and Geosciences, Delft University of Technology, 2600 GA Delft, The Netherlands

ARTICLE INFO

Keywords:
Outlier detection
Traffic data
Statistics-based
Distance-based
Density-based

ABSTRACT

In order to improve the veracity and reliability of a traffic model built, or to extract important and valuable information from collected traffic data, the technique of outlier mining has been introduced into the traffic engineering domain for detecting and analyzing the outliers in traffic data sets. Three typical outlier algorithms, respectively the statistics-based approach, the distance-based approach, and the density-based local outlier approach, are described with respect to the principle, the characteristics and the time complexity of the algorithms. A comparison among the three algorithms is made through application to intelligent transportation systems (ITS). Two traffic data sets with different dimensions have been used in our experiments carried out, one is travel time data, and the other is traffic flow data. We conducted a number of experiments to recognize outliers hidden in the data sets before building the travel time prediction model and the traffic flow foundation diagram. In addition, some artificial generated outliers are introduced into the traffic flow data to see how well the different algorithms detect them. Three strategies-based on ensemble learning, partition and average LOF have been proposed to develop a better outlier recognizer. The experimental results reveal that these methods of outlier mining are feasible and valid to detect outliers in traffic data sets, and have a good potential for use in the domain of traffic engineering. The comparison and analysis presented in this paper are expected to provide some insights to practitioners who plan to use outlier mining for ITS data.

1. Introduction

Intelligent transport systems (ITS) are becoming more and more widespread as a means of dealing with the problems of transport, offering a range of solutions for transport users, operators and managers. Since the advent of intelligent transportation system, immense amounts of traffic data have been gathered every day, which provide a rich source of traffic information for various traffic management and traveler information applications. Due to various reasons, such as detector faults, transmission distortion, emergent traffic accident or other possible influence factors, the traffic data collected is inevitably corrupted, and often contains data item that do not comply with the general behavior of the data model, which is termed as outlier. In the traffic data context, any observation that is substantially different from the main traffic flow can be defined as an outlier. There could be two types of outliers: (*a*) outliers caused by measurement error or equipment failure (faulty values) and (*b*) outliers reflecting ground truth (e.g., a vehicle traveling at an extremely low or extremely high speed compared with the speeds of the other vehicles) (Park, Turner, & Spiegelman, 2003).

One of the key techniques of intelligent transportation system is the technique of estimating and forecasting the traffic parameters. When the traffic data collected is used to build a model, these outliers are not representative, and can not describe the system behavior effectively. Thus, inclusion of such outliers in the traffic model may lead to misleading results. In order to improve the veracity and reliability of the dynamic traffic information and to ensure the effect of the traffic model, it is essential and necessary to identify the abnormal data and remove them from the data set. This procedure is called data quality control or data cleaning. Turner, Albert, Gajewski, and Eisele (2000) have listed several reasons why quality control procedures are especially critical with ITS data.

As the second kind of outlier is concerned, it may contain much more valuable information than the general data. For instance, when a traffic jam or a traffic incident happens, traffic flow will change suddenly and this will be reflect by outliers. We can detect traffic incident through recognizing outliers. Such an outlier is a valid measurement and may provide useful, important and valuable information. Analyzing these outliers, we can draw out unknown but potentially important patterns.

Finding and analyzing outliers is the essential content of data mining called as outlier mining, which is an interesting and important of task of data mining. The outlier mining technique finds many applications in credit card fraud detection, network intrusion

* Corresponding author. Address: Transportation College, Southeast University, 210096 Nanjing, China.
E-mail addresses: shuyan.chen@tudelft.nl, chenshuyan@seu.edu.cn (S. Chen), H.J.vanZuylen@tudelft.nl (H. van Zuylen).

0957-4174/$ - see front matter © 2009 Published by Elsevier Ltd.
doi:10.1016/j.eswa.2009.06.008

detection, medical treatment analysis, and so on. Since traffic data quality has been noted as an important consideration in ITS, a few of literatures can be found on outlier related studies for transport. Turochy and Smith (2000) wrote one of a few pioneering papers to monitor traffic conditions with multivariate statistical quality control (MSQC) which introduce Hotelling's T^2-statistic as the monitoring statistic for the quality control of ITS data, and take into account the relationships among the traffic variables measured. Park et al. (2003) presented alternative procedures for statistical quality control of ITS data that based on variants of the Mahalanobis distance and empirical cutoff points. The methods classified data as outliers on the basis of comparisons with empirical cutoff points derived from extensive archived data rather than from standard statistical tables. This method was illustrated by ITS data from San Antonio, Austin and Texas. Recently, Ban et al. (2007) applied a local median absolute deviation (MAD) method to remove outliers from raw probe vehicles data to get the "ground-truth" travel times. Their study reveals that the local MAD method is very effective to remove outliers if the length of time window is properly selected. In the study of Kingan and Westhuis (2006), two outlier detection techniques, using residual standard deviation and Cook's distance, respectively, were employed to remove outlier for traffic growth prediction, the historical annual average daily traffic (AADT) values from several thousand sites in the state of New York were used as a case. Omenzetter, Brownjohn, and Moyo (2004) conducted statistical analysis of wavelet coefficient time series to detect outliers, so as to identify abrupt, anomalous and potentially onerous events in the strain data recorded by a multi-sensor structural health monitoring (SHM) systems installed in a major bridge structure. Such events may result, among other causes, from sudden settlement of foundation, ground movement, excessive traffic load or failure of post-tensioning cables.

There are various methods in detection of outliers. To name a few, they are statistics-based, distance-based, density-based, deepness-based, and deviation-based outlier detection method. However, many of the applications of outlier studies in traffic engineering domain were just limited within the statistics-based approach. To our best knowledge, many other typical methods of outlier mining are still seldom applied to the ITS databases, although they are very popular and have been applied successfully in many other areas. The objective of this research is to investigate how their performance is if these outlier methods are used to deal traffic problems. It is aimed to compare the effectiveness of different techniques of outlier detection specifically to remove outliers prior to traffic modeling, or to explore the intrinsic patterns hidden in traffic data. In this paper, we introduced the techniques of outlier mining into traffic engineering domain, applied them to remove outlier before modeling travel time and traffic flow, hoping to provide some insights to practitioners who plan to use outlier mining techniques to the ITS databases.

The rest of this paper is organized as follows: The next section gives an outline of the existing outlier data detection techniques, and introduces three typical outlier mining approaches, especially on analyzing their principle, characteristics as well as their time complexity. Section 3 provides two applications of outlier mining in traffic engineering domain, and the results are compared and discussed. This is followed by the conclusion remarks and future study directions in Section 4.

2. Approaches of outlier detection

An outlier is defined as a data point that does not comply with the general behavior of the data model. The detection of outlier can be described as a process that selects k samples that are considerably dissimilar, exceptional, or inconsistent with respect to the remaining data. Outlier mining actually consists of two sub-problems: firstly, to define what kind of data is deemed to be exceptional in the given data set; and secondly, to find an efficient algorithm to obtain such data (Han & Kamber, 2001). Recently, many outlier detection algorithms have been proposed which can be categorized roughly in several approaches as follows: statistics-based, distance-based, density-based, deepness-based, deviation-based, clustering-based method, and so on (Huang, Lin, Chen, & Fan, 2006). The choice of the method depends on the number of dimension of the data, data type, samples size, algorithms efficiency (time-complexity and space- complexity), and the users' understanding of the problem. Due to the space limitation of this paper, we will just describe three outlier detection algorithms in more detail.

2.1. Statistics-based outlier detection approach

The first and simplest outlier detection technique is the statistics-based method (Han & Kamber, 2001; Huang et al., 2006). Almost all applications of outlier detection to traffic data can be categorized into this approach. The main idea of this approach assumes a distribution or probability model for the given data set (e.g., a normal distribution) and then identifies outliers with respect to the model using a discordancy test. A statistical discordancy test examines two hypotheses, a working hypothesis and an alternative hypothesis. The working hypothesis, $H : x_i \in F,\ i = 1, 2, \ldots, n$, assumes the entire data set of n comes from an initial same distribution model F, while the alternative hypothesis $\overline{H}$ assumes the data comes from another distribution model G, $\overline{H} : x_i \in G,\ i = 1, 2, \ldots, n$.

The working hypothesis is refused or accepted by test statistics under significance lever α. The working hypothesis is retained if there is no statistically significant evidence supporting its rejection. A discordancy test verifies whether an object x_i is significantly large or small in relation to the distribution F (Han & Kamber, 2001). Different test statistics have been proposed for use as a discordancy test, depending on the available knowledge of the data. Take one-dimensional data set following a normal distribution as an example, that is, $X \sim N(\mu, \sigma^2)$, where μ is the mean and σ is the standard variance, then $(X - \mu)/\sigma \sim N(0, 1)$, it has,

$$p\left(\left|\frac{x-\mu}{\sigma}\right| < z_{\alpha/2}\right) = 1 - \alpha \tag{1}$$

A $(1 - \alpha) * 100\%$ confidence interval for a normally distributed sample is,

$$(\mu - \sigma z_{\alpha/2}, \mu + \sigma z_{\alpha/2}) \tag{2}$$

where, $z_{\alpha/2}$ is the critical values for hypothesis testing given significance probability $\alpha/2$, and it can be obtained by calculating the inverse cumulative distribution function.

According to this equation, X lies in $(\mu - \sigma z_{\alpha/2}, \mu + \sigma z_{\alpha/2})$ under significance lever α. In other words, the probability for an object falling outside this range is less than $\alpha * 100\%$, so it is a low probability event which is regarded as impossible to happen in a single examination. Therefore, if one object falls within this interval, it is normal, otherwise, it can be considered as an outlier with some reasons.

Such kind of algorithm is simple and its time complexity is $O(n)$, however, the result is very much dependent on model F chosen since x_i may be an outlier under one model and a perfect valid value under another. Another major drawback is that it is only appropriate for one-dimensional data but not multi-dimensional data.

Detection of outliers in multivariate data is more difficult than among univariate observations because outliers have more room to hide in the bulk of multi-dimensional data, and the analysis needs to account for correlation *among the different variables mea-*

sured. Multivariate outlier identification procedures often depend on a Mahalanobis distance or a related statistic, Hotelling's T^2-statistic. The Mahalanobis distance or Hotelling's T^2-statistic is a weighted (squared) distance from the mean with weights proportional to the inverse of variability. It attempts to measure how far an observation is from the center of the data, taking into account the inherent variability and correlation in the data.

Let $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ be a multivariate observation consisting of p measurement (such as speed, volume, or occupancy), where $i = 1, \ldots, n, n$ is the total number of observations. The underlying assumption for classical multivariate outlier identification procedures with Hotelling's T^2-statistic with F-distribution cutoff values is that the observations independently come from a multivariate normal population, $X \sim N(\mu, \sigma^2)$, where μ is common mean and σ is covariance matrix. Under these assumptions, the squared Mahalanobis distance (D_i^2) is

$$D_i^2 = (x_i - \hat{\mu})\hat{\Sigma}^{-1}(x_i - \hat{\mu}) \tag{3}$$

where, $i = 1, 2, \ldots, n$, $\hat{\mu}$ and $\hat{\Sigma}$ are the sample mean and covariance matrix, respectively.

The Mahalanobis distance can be approximated by an F-distribution $[p(n-l)(n+l)/n(n-p)]F_{p,n-p}$. At the specified significance level α, the null hypothesis that the new observation, x, and the reference samples come from populations with equal means μ cannot be rejected if it meet the following condition of formula. In this manner, a determination can be made as to whether a new observation can be considered as outlier or not for a given significance level defined by α

$$(x - \hat{\mu})\hat{\Sigma}^{-1}(x - \hat{\mu}) \leqslant [p(n-l)(n+l)/n(n-p)]F_{p,n-p}(\alpha) \tag{4}$$

The Mahalanobis distance differs from the Euclidean distance in that it accounts for the relative dispersions and inherent correlations among vector elements. It is a probabilistic distance in the sense that equal distances imply equal likelihoods. The Mahalanobis distance or Hotelling's T^2-statistic is very promising for detecting outliers technique because it is capable of handling multi-dimensional data, and it has been considered in the transportation context recently (Park et al., 2003; Turochy & Smith, 2000; Omenzetter et al., 2004).

Another kind of statistics-based method detecting outlier is via predefined threshold for certain calculated quantity, such as MAD (Ban et al., 2007), it is often a simplicity version of classical statistics-based method, for the threshold is often set by experience or experiment without considering data distribution and significant level.

Generally speaking, statistics-based approach is simple and its time complexity is linear in term of the size of data and dimension, $O(pn)$. However, it requires knowing in advance the distribution of the data and the distribution parameters. This is always difficult, as the real data usually does not comply with any ideal mathematics distribution. Furthermore, it is difficult to deal with periodicity data and categorical data. This restricts its application.

2.2. Distance-based outlier detection approach

Knorr and Ng (1998) and Knorr et al. (2000) originally proposed a new outlier definition based on the concept of distance and present a mining algorithm. An object x in data set X is defined as an outlier with parameters p and d, described as DB(p,d), if a fraction of p of the objects in X lie at a distance greater than d from x. Suppose $M = n^*(1-p)$, where n is the number of data, outlier detection is a process that determine whether the number of the points which are at a distance less than d from x is more than M. If so, x is not an outlier, otherwise x is an outlier.

This definition generalizes the definition of outlier and it is suitable when the data set does not fit any standard distribution model, it can discover outliers effectively, and especially it can be used in multi-dimensional samples. The algorithm runs in $O(k * n^2)$ with respect to the worst time complexity, where k is the number of dimensions. However, this approach is sensitive to the parameter p and d and the results are instable for this reason. It requires the user to test different p and d to set appropriate values for the parameters by resample technology in advance, so, it is difficult to get the suitably defined parameters in practical applications. In addition, it has to calculate the distance between all the samples in k dimension data set, the efficiency is low for the great data set in high dimensional space. In order to address these drawbacks, Ramaswamy, Rastogi, and Shim (2000) modified the definition of outlier. The new definition of outlier is based on the distance of a point p from its kth nearest neighbor, denoted with $D^k(p)$, and it can be described as follows: Given a k and m, a point p is an outlier if no more than $m-1$ other points in the data set have a higher value for D^k than p. This means that the top m points having the maximum D^k values are considered as outliers. Angiulli and Pizzuti (2005) proposed a new definition of distance-based outlier and an algorithm, called HilOut, designed to efficiently detect the top m outliers of a large and high dimensional data set. Given an integer m, the weight of a point is defined as the sum of the distance separating it from its k nearest neighbors. Outliers are those points scoring the largest values of weight.

2.3. Density-based outlier detection approach

The density-based outlier detection was proposed by Breunig, Kriegel, and Ng (2000). A new notion of local outlier is introduced that measures the degree of an object to be an outlier with respect to the density of the local neighborhood. This degree is called local outlier factor, LOF, and is assigned to each object.

Given a positive integer k, define the k-distance of x, denoted as k-distance(x), is defined as the distance $d(x,o)$ between object x and object $o \in D$, such that

(1) for at least exists k objects $o' \in D \setminus \{x\}$, it holds that $d(x,o') \leqslant d(x,o)$, and
(2) for at most exists $k-1$ objects $o' \in D \setminus \{x\}$, it holds that $d(x,o') < d(x,o)$;

Define the k-distance neighborhood of x as

$$N_k(x) = \{q \in D \setminus \{x\} | d(x,q) \leqslant k - \text{distance}(x)\} \tag{5}$$

The local reachability distance of object x with respect to object o is defined as,

$$\text{reach-disp}_k(x,o) = \max\{k - \text{distance}(o), d(x,o)\} \tag{6}$$

The local reachability density of x is defined as the inverse of the reachability distance-based on the k-nearest neighborhood,

$$lrd_k(x) = \frac{1}{\frac{\sum_{o \in N_k(x)} \text{reach\_disp}_k(o)}{|N_k(x)|}} \tag{7}$$

where, $|N_k(x)|$ is the cardinality of $N_k(x)$.

The local outlier factor of x is defined as

$$lof_k(x) = \frac{\sum_{o \in N_k(x)} \frac{lrd_k(o)}{lrd_k(x)}}{|N_k(x)|} \tag{8}$$

The outlier factor of object x captures the degree of a point being an outlier. It is the average of the ratio of the reachability density of x and its k-nearest neighbors. The key difference between this notion and other existing notions is that being outlying is not a binary property. Instead, each object is assigned an outlier factor, which is the degree the object is being outlier. However, the

cost the algorithm spends on searching the neighborhood is high. Another density-based outlier mining algorithm is local correlation integral (LOCI) proposed by Spiros, Hiroyuki, and Gibbons Phillip (2003), which use approximate computation to speed up the outlier detection. This method computes the MDEF (multi-granularity deviation factor) of every point and uses it as an evidence of an outlier. But the calculating of the standard deviation is still very expensive. Ren, Wang, and Perrizo (2004) has brought forward a new density definition, relative density factor (RDF), to measure the density of one point with respect to the density of its neighbors, and proposed an RDF-based outlier detection method which can efficiently prune the data points which are deep in clusters, and detect outliers only within the remaining small subset of the data. In a few words, density-based outlier mining has attracted more research and application, because of its uniqueness and the ability to find the local outlier that other methods may not be able to detect.

3. Empirical study

The objective of the case study is to compare how the different outlier detection methods mentioned above perform if they are used to clean an ITS database by identifying erroneous or suspect observations that are likely caused by equipment failure or unusual traffic events. The comparison of three methods for outlier detection is discussed in this section with several experiments conducted. We start with a data set of travel time which is one-dimensional and will be used to build a prediction model, and then proceed to a real world dataset of traffic flow, which has been collected for studying the relation between speed and traffic flow rate. All algorithms were implemented in Matlab 6.5, and were run on the computer with 1.50 GHz of intel pentium processor and 256 MB of memory.

3.1. Detect outlier prior to modeling travel time prediction

Travel time information turns out to be very valuable for the traffic management and traveler information applications, this is because, firstly, travel time is one crucial measure to assess the performance of traffic conditions; secondly, travel time has become the most critical travel information for the prediction results can directly affect the decision of drivers to choose their way or the departure time when they start their journey (Ban et al., 2007). Travel time can be estimated from many types of sources, such as loop detector data and probe vehicles data. However, travel times collected or generated from various sources may contain significant amount of outliers. The presence of outliers distorts the statistical properties of the data, and such distortions can result in incorrect statistical estimates for travel time. As a consequence, outliers may severely degrade the performance of the travel time prediction model. In the case of route navigation, non-handled outliers can distort the navigation solution in such a way that the prediction of travel time lacks accuracy.

Various methods and techniques can be used for handling outliers. There are two main strategies, one is using estimation techniques that are robust to deal with the outliers, and the other is detecting and removing the outliers before they are used in further calculations. Here, we adopt the latter. In such a case, a data should be removed from the data set before building the prediction model if it is flag as an outlier. Travel time data set is extracted from the probe vehicle data. The route is about 16km and travel time can be obtained directly from the passage times at the starting point and ending point. The data set was collected within 24 h on 2007-09-19. We limit travel times within 0–3600 s (2 h), while those outside this range were ignored. After this preprocess, we obtained a data set with 2899 data points. The excessively long time to travel the route is possibly caused by the vehicles that left and re-entered the route at some point, or the data is recorded spuriously.

We conducted a number of experiments on the travel time data with three different outlier detection algorithms and presented the comparison results in the following subsections.

3.1.1. Statistics-based approach

First of all, to carry out a hypothesis test to determine which kind of distribution the data set follows. As the first step, we perform a Kolmogorov–Smirnov test to compare the values in the data set X with a normal distribution. The null hypothesis is that X has a Normal distribution. The alternative hypothesis is that X does not have that distribution. The result showed that we can reject the null hypothesis that the values come from a normal distribution.

Similarly, test against the following hypothesized distribution, including Beta distribution, Poisson distribution, Chi-square distribution, Exponential distribution, Gamma distribution, Lognormal distribution, Negative binomial distribution, Rayleigh distribution, Student's t distribution, and Weibull distribution. It seems that the data set could not follow any distributions mentioned above. This example illustrates that the real data often do not comply with any ideal mathematics distribution.

In such case, how to employ the statistics-based approach to find out outliers? We proposed the following method. From the previous Kolmogorov–Smirnov test, we found that the distribution of the data set is closer to Normal distribution, Gamma distribution, Lognormal distribution, and Negative binomial distribution, so we use the statistics-based approach under the assumption that it follows these distributions roughly. Set the confidence level $\alpha = 0.01$, the numbers of outliers obtained under different distribution assumptions are listed in Table 1. It can be seen that the number of outliers found is very close, from 62 to 73.

Fig. 1 shows the detection results that are reasonable, the top is the original data contains many outliers, from the second to the bottom in this figure, each one shows the data after outliers removed. Most of outliers are detected successfully. To avoid a bias to any distribution, one solution is that one data point can be regards as an outlier if it is detected under morn than three distributions.

3.1.2. Distance-based outlier detection approach

Since the outliers detected by this approach depend on the parameters set, let parameter p vary from 0.10 to 0.90 with the step of 0.1, and d vary from s to 9^*s with the step of s, where $s = 248.46$, is the average distance between all the samples, run the distance-based algorithm on this data set. The numbers of outliers detected are listed in Table 2. It can be seen that the parameters, p and d, have much influence on this method, and the number of outliers change wildly and monotonically, from 2899 to 3, according to the increase of p and d. When $p = 0.1$ and $d = s$, the algorithm take all the data as outliers. Obviously, this is over detected, as it labels all the normal data as abnormal data. With the increase of d if fix $p = 0.1$, fewer data are regarded as outliers. When $p = 0.1$ and $d = 9s$, it finds only 14 outliers. Obviously, this is under detected, as it recognizes many abnormal data as the nor-

Table 1
The possible distribution and the number of outliers detected.

Distribution	Number of outliers
Normal	62
Gamma	70
Lognormal	73
Negative binomial	70

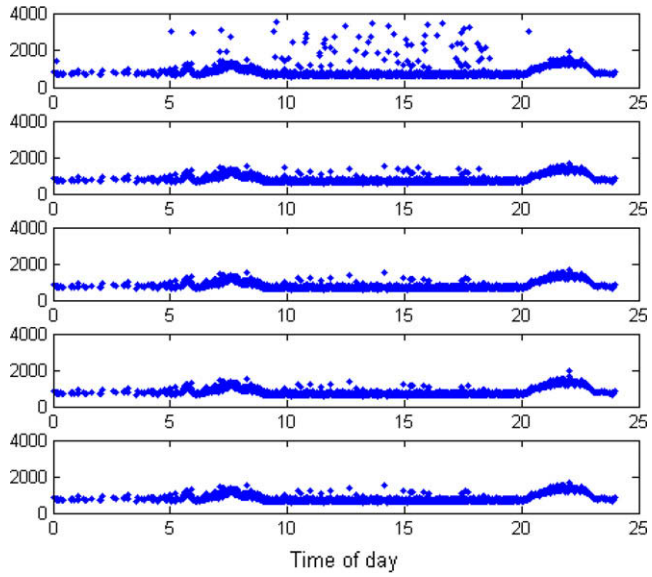

Fig. 1. Traffic time collected and the outliers recognized by statistics-based outlier mining algorithm.

Table 2
The numbers of outliers changed with the parameters set.

p	*d*								
	1s	2s	3s	4s	5s	6s	7s	8s	9s
0.1	2899	1361	101	56	49	38	29	23	14
0.2	1888	318	91	56	49	36	27	23	14
0.3	639	305	90	55	48	36	27	23	14
0.4	614	285	88	55	45	36	27	22	14
0.5	587	257	79	54	44	36	26	21	13
0.6	551	224	74	54	43	35	25	19	12
0.7	500	190	64	54	42	34	25	18	11
0.8	347	90	55	45	36	26	22	14	7
0.9	109	56	49	36	27	22	14	7	3

mal data. Fig. 2 illustrates the original data and the data after outliers removed, the moddle one corresponds to under detected, while the bottom one corresponds to over detected, since there are many 'good' data point are regarded as outliers.

From the bottom of Fig. 2, we also find that some data points corresponding to the peak hours have been labeled as outliers and be removed. Actually, these outliers are inliers since the travel time is time dependent. In order to deal with this question, we divide the whole data set into different parts and use the distance-based approach on each partition, which yields better detection results, as shown in Fig. 3, where we set $p = 0.2$ and $d = 2s$. Compare the middle with the bottom, one can find that the inliers corresponding to the pear hours are kept as the normal data if we partition the data set.

3.1.3. Density-based detection approach

When using this approach, the parameter k has to been chosen to compute the density in the neighborhood of object p, which is a measure of the volume to determine the local outlier factor of point p. Breunig et al. (2000) pointed that the LOF value is influenced by the choice of the k value, and proposed a heuristic method to pick the right k values for the LOF computation. Let LB and UB to denote the "lower bound" and the "up bound", they provided several guidelines for picking the range of k values. The first guideline is that k should be at least 10 to remove unwanted statistical fluctuations. This value could be application-dependent. For most of the datasets that they experimented with, picking 10–20 appears to work well in general. The second guideline is that picking the upper bound value for k as the maximum number of "close by" objects that can potentially be local outliers.

Following such guidelines, we pick 10 for LB and 400 for UB in this experiment. Having determined LB and UB, we can compute for each object its LOF values within this range. Firstly, choose $k = 10$ to calculate the local outlier factors of all data. Then, increase k at the step of 5, repeat the procedure of computation until k is larger than UB. For each k value between this ranges, the minimum, maximum and mean LOF values are shown in Fig. 5. LOF has a basic property, namely that for objects deep inside a cluster, its LOF is close to 1, and should not be labeled as a local outlier (Breunig et al., 2000). Here, we set 2.0 as a threshold for LOF, and an object is labeled as an outlier if its LOF exceed 2.0. Thus, for each k value between the ranges, we obtained the number of data outlying and also showed them at the bottom of Fig. 4.

It can be seen from this figure that the minimum and average LOF values change little when the k value is adjusted. There is a

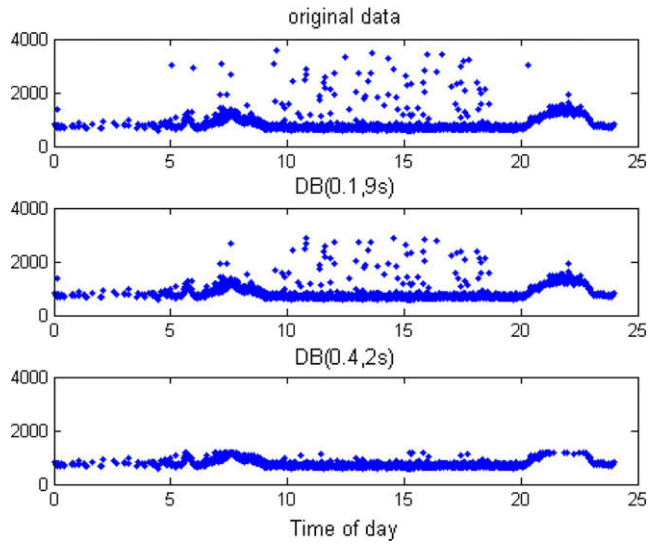

Fig. 2. Traffic time collected and the data after outliers recognized by distance-based outlier mining algorithm.

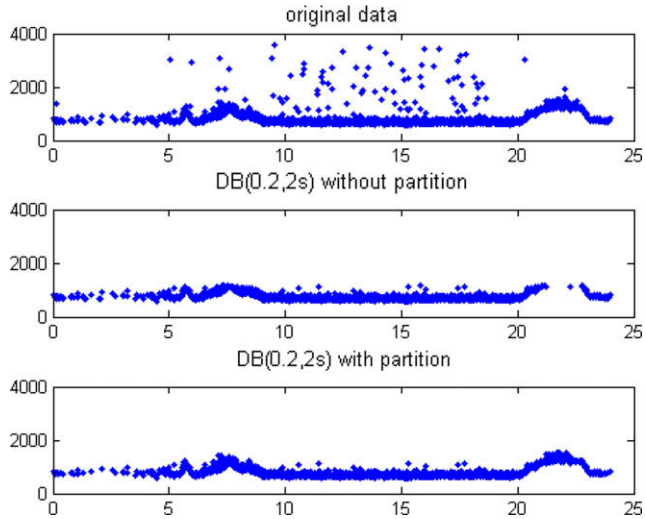

Fig. 3. Comparison distance-based approach between with partition and without partition.

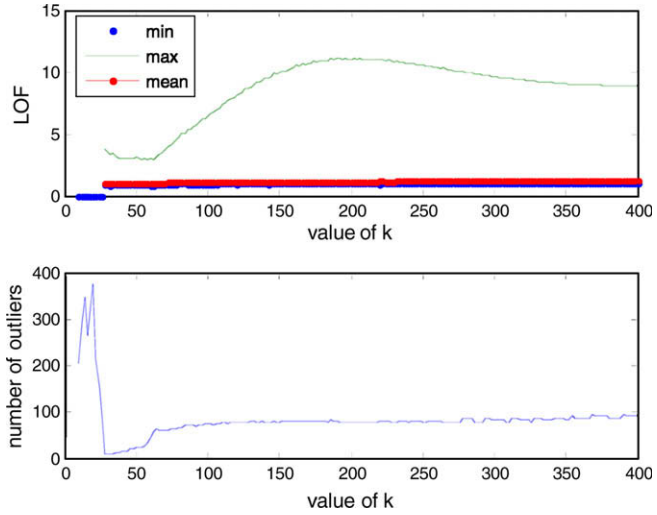

Fig. 4. Fluctuation of the numbers of outliers detected and the outlier-factors with k value. Here, the threshold for LOF is set to 2.0, that means a data is labeled as outlier if its LOF exceed 2.0.

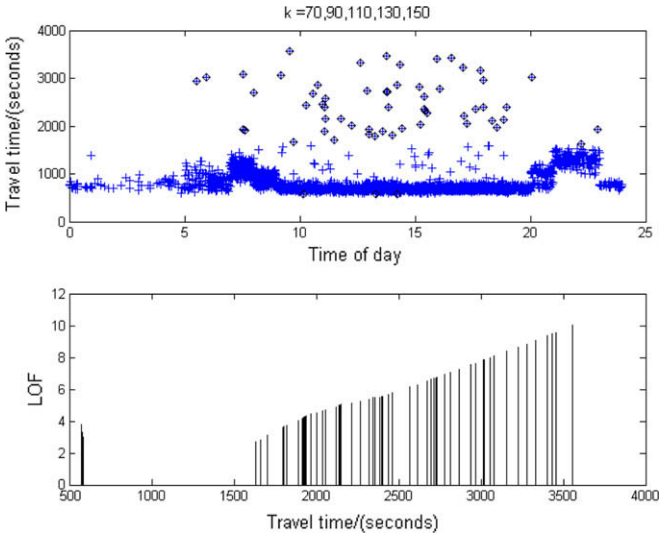

Fig. 5. The outliers found by density-based algorithm and their average LOF.

sharp leap when k equals to 20. Before this k value, the minimum and average LOF value is very stable. In fact, if we pick k from the range of Knorr, Ng, and Tucakov (2000), the minimal LOF of all the objects is 0, and the average of LOF is positive infinity (note that infinity can not be shown in this figure), the reason for this is that, there is at least an object p whose k neighbors are same as it, which make the local reachability density of p become infinity, for this special case, we define its LOF as 0. For the same reason, there is an object p which has a neighbor whose local reachability density is infinity, which makes its LOF become infinity. Pick k value from the range of [20,400], the minimum and average LOF value is also very stable, the minimum LOF changed from 0.82 to 0.98, and the average LOF changed from 1.03 to 1.21. In contrast, the maximum LOF values change wildly, however, they neither decreases nor increases monotonically. Increase k from 20 to 54, there is not a corresponding monotonic sequence of changes to maximal LOF. Increase sequence of k values from 56 to 178, the maximal LOF increases monotonically. As the k value continues to increase, the maximum LOF value goes down slightly, and eventually stabilizes at a certain value. As the number of the outliers detected is concerned, it also changes widely, even much than the maximal LOF, especially when k is given a small value, corresponding to the left side of the curve, the number of outliers goes up and down dramatically. As the k value continues to increase, this curve fluctuates slightly.

According to the analysis above, it is suitable to choose the range of [60, 400] for k picks its value from in this special application. Having determined LB and UB, we compute for each object its LOF values within this range. How to determine LOF for each object based on these LOF? Breunig et al. (2000) proposed a heuristic method to rank all objects with respect to the maximum LOF value within the specified range. That is, the ranking of an object p is based on $\max\{\mathrm{LOF}_k(p)|k \in [\mathrm{LB}, \mathrm{UB}]\}$. Instead of using the maximum LOF, averaging the LOF that each data obtained under different value of k as its LOF is an alternative. Generally speaking, using the average LOF can produce stable detect result than the maximum LOF.

Here, we employ the average LOF. For speed of detection, we compute the average LOF in the k range of 70–150 at the interval of 20 instead of 1. Then, we labeled 60 points whose average LOF is the highest among all the data as the local outliers. These outliers and their LOF are plotted on the Fig. 5.

At the top of this figure, there are three points labeled as outliers, 573 with the maximum LOF of 10.15, 580.00 with13.29, and 585.00 with 14.20, respectively. In fact, they are not outliers. The reason for this is that their k-nearest neighbors have a higher local reachability density, which makes these three points obtain higher LOF. Now reduce the number of potential outliers that we want to find, then what will be happened? Fig. 6 illustrates the evolution process. From the plot, it is clear that the fake outliers disappeared one by one with the decrease of the number value. When the number value reaches 57, the algorithm mislabels two points, continue reducing, it mislabels one when the number value reaches to 53, at last, it mislabels none until the number is reduced to 51. Thus, besides picking the right range for k, we have to take careful to pick the right number of outliers that we think the data set possibly contain, when we employ the density-based algorithm to find outliers.

The results of the three methods are comparable, but a few differences exist. Compared to statistics-based and distance-based algorithms, density-based algorithm is prone to mislabel some data deep inside a cluster, indicating that they cannot be labeled as outliers, but they are considered as outliers, due to higher local

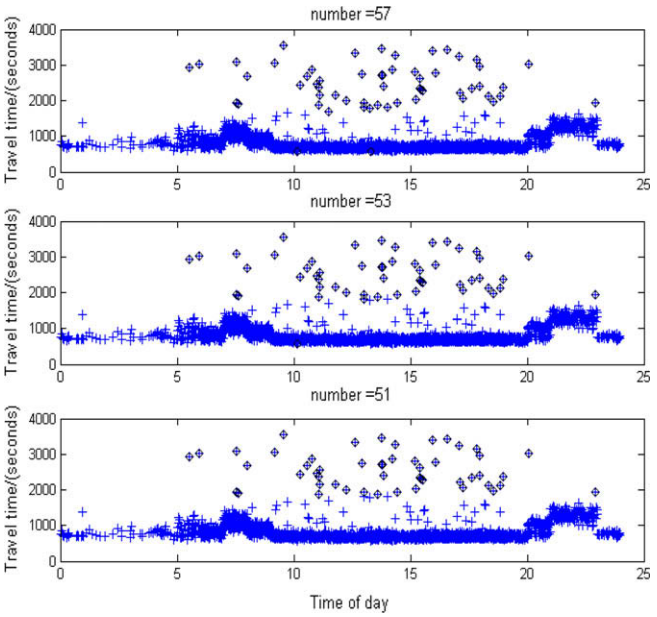

Fig. 6. Fake outliers disappear along with the decrease of the number of outliers that we want to find.

reachability density of its neighbors, since the higher the local reachability densities of p's k-nearest neighbors are, the higher is the LOF value of p. However, this possibility can be decreased by reducing the number of outliers or increasing the threshold of LOF for outliers, these two strategies have the same effect.

Another significant difference is the runtime that the algorithms need. Statistical-based algorithm run very fast, density-based algorithm run very slow, and distance-based algorithms run middle, this is because that distance-based and density-based algorithms speed much time calculating the distance between any pair of all the points, in particularly, density-based algorithm has to speed much time calculating the local reachability density of k-nearest neighbors of each data point. Take this travel time data set with about 3000 points as an example, three algorithms need 1.47, 26.56 and 290.74 s, respectively. With the increase of data set size, the difference of runtime between three algorithms become very large, this has been illustrated by our experiments.

3.2. Detect outlier from multi-dimensional traffic flow data

Now turn to the experiments conducted on the multi-dimensional traffic flow data. The traffic flow on freeways is described traditionally in term of three parameters: the mean speed, the traffic flow rate, and the traffic density, and the relationship between flow rate and traffic density can form a curve called the fundamental diagram for traffic flow. In order to study the relationships between the three traffic parameters and speed dispersion, traffic data were obtained by video-tapping from Lukou airport freeway, Nanjing, Jiangsu province, Wang et al. (2007). The site data collected from is a 5 km segment between two ramps in Lukou airport freeway, instrumented with video cameras every two kilometers each side, and the traffic data collected from station 3 during the period from 14:34 to17:37 on July 3, from 07:40 to 10:45 on July 6, and from 16:32 to 17:50 on July 11, 2007 were utilized in our study. By a program code, traffic flow rate and density are captured for each minute, and there are 709 data points generated from inmost lane at station 3 in two directions.

Abnormal traffic flow data might be caused by a fault of the detection devices or transmission lines, or caused by an unexpected traffic incident which makes the traffic flow breakdown. Whether outliers are sampling errors or caused by an abnormal traffic event, these abnormal data may make the gist of the model ambiguous and cannot reflect the essence of the true system behavior. Therefore, the raw data collected needs to be processed to remove or correct outliers before modeling in order to reduce the influence of outliers on the normal data, thus to improve the validity and reliability of the traffic data to represent observations of the modeled traffic process.

3.2.1. Experiment with the original data

For the benefit of outlier algorithms, before the following processes, all the data were normalized to the range of [0, 1]. Firstly, we used the statistics-based algorithm, speaking precisely, the Mahalanobis distance (Hotelling's T^2-statistic), to find out outliers. Set the significance level to 99.9%, and we found 11 outliers. Then, we used the distance-based algorithm. In order to find the suitable values for p and d, we run this algorithm with different parameter values, p varying from 0.15 to 0.6, d varying from $2s$ to $5s$, where s is the average distance between all the data, $s = 0.28$. As a result, we chose 0.22 and $3s$ for parameters p and d, respectively, and found out 11 DB(0.22,39i) outliers. Next, we applied the density-based algorithm. Since the parameter k has much influence on the detect results, we let the parameter k increase from 20 to 150 at the step of 10 to calculate the value of LOF for each data. Instead of using the maximum LOF, here we average the LOF of each data obtained under different value of k as its LOF. As a result, 12 data with the top average LOF were labeled as outliers. The detection results of three methods were shown in Table 3. The symbol '*' means that this data point is detected as an outlier by a method, and the last column given the number of methods which detected this data as an outlier.

Although three methods detect the outliers from different angles, the results are quite consistent according to the results. Among the 16 outliers labeled by at least one method, there are six data detected by all of the three methods, which are the most probable outliers, Nos. 57, 229, 251, 653, 663 and 689. Besides, six data points, Nos. 54, 88, 160, 219, 255 and 691, are considered as outliers by two of the three methods. It seems that three algorithms perform much similarly. This can also been seen from Fig. 8, which plot the outliers found by three outlier detection algorithms on the same traffic flow data. The top is the results obtained by statistics-based algorithm, the middle is by distanced-based algorithm, and the bottom is by density-based algorithm. It can be seen obviously that the results were quite comparable. Comparing three sub-figures in Fig. 7, most of the visually observed outliers in one sub-figure are also located in other figures. This means that all the three methods were effective and the outliers detected were reasonable.

However, it should be noted that there are some differences between outliers obtained by the three methods. From this figure, it can be seen that two data points locating below the curves, No. 88 with value of (22.81, 1200) and No. 160 with value of (22.14, 780), are potential outliers, and statistics-based and distance-based methods successfully labeled them as outliers, but distance-based method failed to detect. As we pointed out before, parameters p and d can affect the results, and the two data have too few neighbors within a small area, so we decrease d and increase p, we found

Table 3
Comparison of the detection results among three algorithms.

No.	Density	Volume	Statistics-based	Distance-based	Density-based	Detected by algorithms
33	31.88	2040		*		1
54	33.87	1920	*	*		2
57	*37.24*	*1980*	*	*	*	*3*
58	31.61	1860	*			1
88	22.81	1200	*		*	2
160	22.14	780			*	2
219	35.65	2280		*	*	2
229	*34.79*	*2100*	*	*	*	*3*
251	*37.10*	*2040*	*	*	*	*3*
255	35.70	2400			*	2
308	4.06	360			*	1
518	4.12	360			*	1
653	*36.28*	*2280*	*	*	*	*3*
663	*37.01*	*2100*	*	*	*	*3*
689	*35.76*	*2280*	*	*		*3*
691	33.68	2040	*	*		2

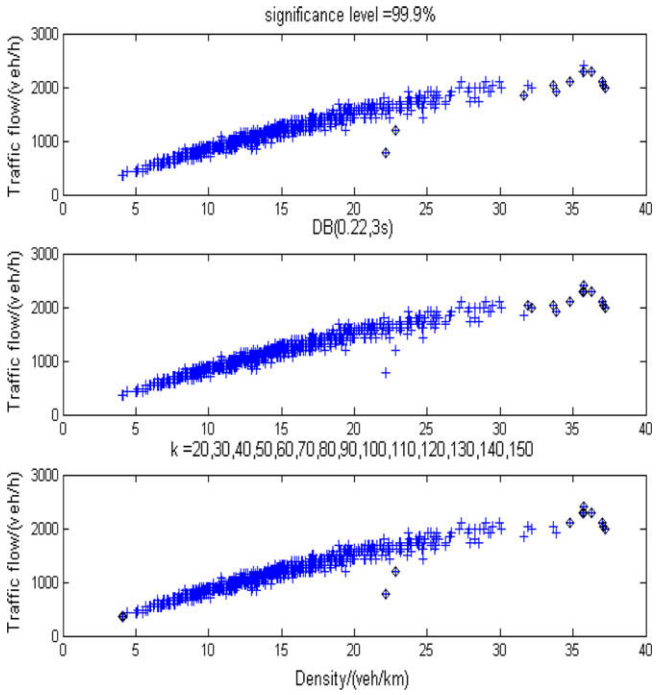

Fig. 7. Comparison among three outlier detection algorithms on traffic flow data. From the top to the bottom, the results are obtained by statistics-based, distanced-based, and density-based algorithm.

Nos. 88 and 160 can be detected under some parameters setting. However, when No. 88 is detected, too many data are labeled as outliers, which means that No. 88 is more like to belong to the normal data. Fig. 8 showed the detect results by distance-based method with $d = 0.28$ and p varied from 0.85 to 0.95 with a step size of 0.05. When $p = 0.85$, it labeled too many data as outliers. When p is increased to 0.95, it failed to detect No. 160. It seems that $d = 0.28$ and $p = 0.90$ is the optimal setting.

In practice, we do not know in advance which method performs best as well as which parameter values are most suited for the collected data. If we combine the detection results of different algorithms, that is, one data item should be labeled as an outlier if a majority of algorithms regard it as such. Such outlier detection method based on an ensemble can takes advantage of different algorithms, and could be more reasonable.

As the runtime is concerned, statistics-based, distance-based and density-based algorithms need 0.29, 1.71 and 16.19 s, respectively. It supports the conclusion again that the statistics-based algorithm run fast, and the density-based algorithm run slowly.

3.2.2. Experiment with artificial constructed data

To assess the effectiveness of these methods of outlier mining on traffic data, it's better to have a "gold standard" data set to conduct experiments. In the "gold standard" data, the outliers are known so that the methods can be tested, and the detection rate, the false detection rate as well as precision of detection can be computed as follows,

$$\text{detection rate} = \frac{\text{the number of outliers correctly detected}}{\text{total number of true outliers}} * 100 \tag{9}$$

$$\text{false detection rate} = \frac{\text{the number of inliers detected as outliers}}{\text{total number of inliers}} * 100 \tag{10}$$

$$\text{precision} = \frac{\text{number of outliers correctly labelled}}{\text{total number of outliers labelled}} * 100 \tag{11}$$

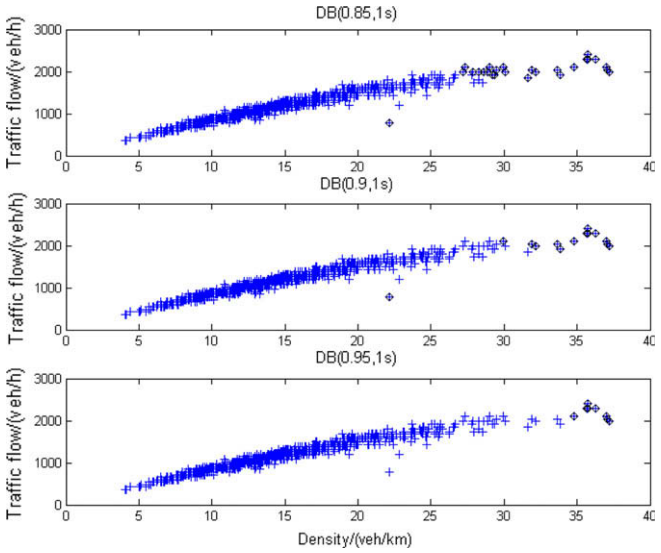

Fig. 8. The detect results by distance-based method with $d = 0.28$ and p varied from 0.85 to 0.95 at the step of 0.05.

Due to the absence of gold standard data, we constructed a date set by introducing some outliers artificially into the traffic flow data collected in Lukou airport. First, remove all the outliers detected by three methods in the previous experiment to make sure that there were no suspicious values. Then generate 10 outliers to simulate malfunction or anomalous true conditions, and add them to the data set. Now, the data set has 703 data points listed in Table 4.

When wet the significance level to 99.9%, the statistics-based algorithm found 11 outliers, including eight artificial outliers. When we run the distance-based algorithm, we tried more than 100 different values for parameters p and d, and find $p = 0.92$ and $d = 0.91$ s, where $s = 0.26$, being the average distances between all data in the constructed data set, yield the best results. It detected 9 outliers including five artificial outliers. Next, we run the density-based algorithm. Let parameter k increase from 20 to 150 with a step size of 10 to calculate the value of LOF for each data point, and averaged the LOF for each data. This method detects all 10 outliers but one (Outlier 4). This is the best result among the three methods obtained. On the base of these detection results, the detection performances of the three methods were calculated and shown in Table 5. Excluding the distance-based method, these methods achieved satisfactory detection results at the cost of very lower false detection rate, lower than 0.5%.

The detection results of the three methods are plotted in Fig. 9, as well as the constructed data for the purpose of comparison. From this figure, one can find easily which outlier is successfully detected, which outlier is failed to detect, and which "inlier" is false detected.

4. Conclusions and future research

In this paper the outlier detection problem in ITS have been discussed. We introduce the application of outlier mining for the purpose of outlier detection. Three approaches for detecting outliers, the statistics-based method, the distance-based method and the density-based method, have been described, and the performance of these methods has been compared. The analysis is completed with the applications to two traffic data sets, one is one-dimensional travel time data, and the other is multi-dimensional traffic flow data. In addition, we constructed a data set based on traffic flow data by adding some artificial outliers to investigate how well three methods performed. Based on our experiment, the statistics-

Table 4
Examples of outliers generated artificially.

No.	Density	Volume	Scenario
1	0	589	Density equals zero and volume is low
2	7.49	0	Volume equals zero and density is low
3	9.70	488	Density is low and volume is closer to minimum threshold (420 mph)
4	23.56	2050	Density is high and volume is near the maximum threshold (2100 mph)
5	3.36	352	Density < minimum threshold (4.41), volume < minimum threshold
6	4.90	799	Density is near minimum threshold, and volume is low
7	10.75	1246	Density is low but volume is near the average
8	31.09	1571	Density is near maximum threshold (32.14), and volume > average
9	24.61	1042	Density is higher than average, volume is near the average
10	33.03	2149	Density > maximum threshold, volume > maximum threshold

Table 5
Comparison between three methods on the constructed data with artificial outliers.

Methods	Detection		False detection		Precision (%)
	Number	Rate	Number	Rate (%)	
Statistics-based	8	80.0	3	0.4	72.7
Distance-based	5	50.0	4	0.7	55.5
Density-based	9	90.0	1	0.1	90.0

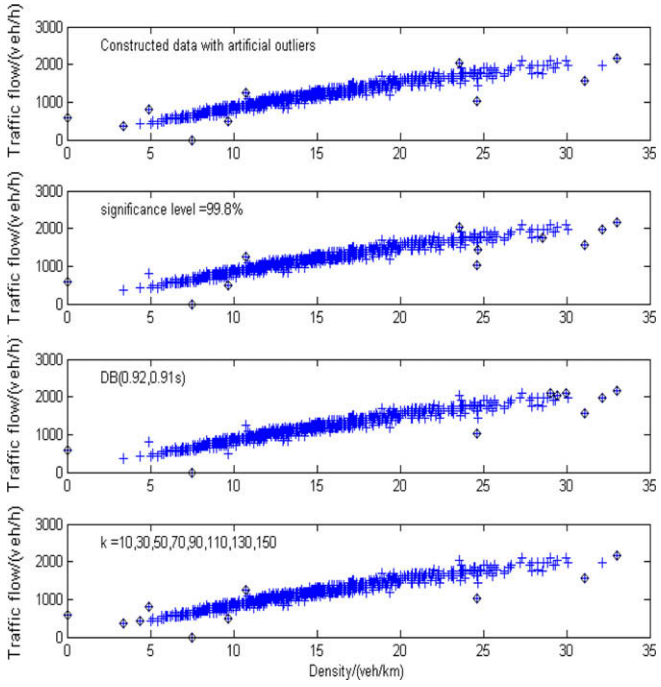

Fig. 9. Comparison among three outlier detection algorithms on a constructed data set with artificial outliers. The top is the constructed data set, where the circle denotes the artificial outliers. Next three pictures are obtained by statistics-based, distanced-based, and density-based algorithm, respectively.

based method and the density-based method are superior to the distance-based method. It also can be concluded that the parameters for each method have much influence on the detection performance, and should be carefully chosen. In order to develop a better outlier identifier, we proposed three strategies, one recommend is a further development of an integrated technique combining the detection results of different approaches, second is partition the data set and then used statistics-based or distance-based approaches on each subset to preserve the characteristics of time-dependant data set, the last one is to average the local outlier factors of each point under different k as a criterion rather than the maximal local outlier factor while using density-base approach.

There is still much work to do further. One is to extend the application of the outlier mining algorithms. Abnormal traffic data have two meanings: sampling errors or true data containing valuable information about some unexpected events. As pointed out in Huang et al. (2006), one person's noise could be another person's signal, thus, outliers themselves can be of great importance. In some cases, the intention of data analysis is the outlier detection, for the outliers may contain important information and may just be our pursuit. At present, we are studying the application of outlier mining method to traffic incident automatic detection. When the traffic incident happen, it will definitely cause the relevant traffic data change rapidly, thus, we can detect traffic incident by recognizing the outliers.

After the recognition of the outlier, the next step is to reveal the meanings of these outliers, answer the question why the outliers are generated, i.e., distinguish outliers due to anomalous conditions from outliers due to equipment failure. Such a study is still rare. How to distinguish erroneous data from abnormal traffic conditions is a very difficult task. To our best knowledge, up to now, there are no any outlier methods or data mining techniques that can cope with this problem. At present, we can make a decision manually after detect outliers by considering the environment, such as traffic conditions, traffic pattern, time when outliers happened, and neighbors of outliers. Our future study include the development of algorithms revealing the meanings of these outliers so that one can distinguish different outliers automatically, although it is a difficult work to be done.

Acknowledgements

This research was supported in part from National Basic Research Program of China (973 Project) (No. 2006CB705500), Key project of National Natural Science Foundation of People's Republic of China (No. 50738001), China Postdoctoral Science foundation (20070411016), and Jiangsu Planned Projects for Postdoctoral Research Funds (0701002C). Shuyan Chen is grateful to Prof. Shawn Turner in Texas Transportation Institute, USA, for his providing much information for this study. Also many thanks are given to Dr. Meng Lu for her revising this manuscript which improve the writing quality of it further.

References

Angiulli, F., & Pizzuti, C. (2005). Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering, 17*(2), 203–215.

Ban, X. (Jeff), Li, Y., & Skabardonis, A. (2007). Local MAD method for probe vehicle data processing. In *Proceedings of the 14th world congress of intelligent transportation systems (ITSWC 2007)*, Beijing, China.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., et al. (2000). LOF: Identifying density-based local outliers. In W. Chen, J. F. Naughton, & P. A. Bernstein (Eds.), *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 93–104). Dallas, Texas: ACM Press.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. Morgan Kaufmann Publishers.

Huang, H., Lin, J., Chen, C., & Fan, M. (2006). Review of outlier detection. *Journal of Research on Computer Application, 8*, 8–13.

Kingan, R. J., & Westhuis, T. B. (2006). Robust regression methods for traffic growth forecasting. *Journal of the Transportation Research Board, 1957*, 51–55.

Knorr, E., & Ng, R. (1998). Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24th VLDB conference* (pp. 392–403). New York, USA.

Knorr, E., Ng, R., & Tucakov, V. (2000). Distance-based outlier: Algorithms and applications. *VLDBJ, 8*(3–4), 237–253.

Omenzetter, P., Brownjohn, J. M. W., & Moyo, P. (2004). Identification of unusual events in multi-channel bridge monitoring data. *Mechanical Systems and Signal Processing, 18*, 409–430.

Park, E. S., Turner, S., & Spiegelman, C. H. (2003). Empirical Approaches to Outlier Detection in Intelligent Transportation Systems Data. *Transportation Research Record, 1840*, 21–30.

Ramaswamy, S., Rastogi, R., & Shim, K. (2000). Efficient algorithms for mining outliers from large data set. In *Proceedings of ACM SIDMOD international conference on management of data* (pp.1–20).

Ren, D., Wang, B., & Perrizo, W. (2004). RDF: A density-based outlier detection method using vertical data representation. In *Proceedings of the fourth IEEE international conference on data mining (ICDM'04)* (pp. 8–11).

Spiros, P., Hiroyuki, K., & Gibbons Phillip, B. (2003). LOCI: Fast outlier detection using the local correlation integral. In *Proceedings of the 19th international conference on data engineering* (pp. 315–326).

Turner, S. M., Albert, L., Gajewski, B., & Eisele, W. (2000). Archived intelligent transportation system data quality: Preliminary analyses of San Antonio transguide data. *Transportation Research Record: Journal of the Transportation Research Board, 1719*, 77–84. TRB, National Research Council, Washington, DC.

Turochy, R. E., & Smith, B. L. (2000). Applying quality control to traffic condition monitoring. In *Proceedings of the 3rd IEEE conference on intelligent transportation systems* (pp. 15–20).

Wang, H., & Wang, W. (2007). Experimental features and characteristics of speed dispersion in urban freeway traffic. In *Proceeding of 86th annual conference of transportation research board*, Washington, USA.