ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20150616163905

CORRECTING AND COMPLEMENTING FREEWAY TRAFFIC ACCIDENT DATA USING MAHALANOBIS DISTANCE BASED OUTLIER DETECTION

Bin Sun, Wei Cheng, Guohua Bai, Prashant Goswami

Subject review

A huge amount of traffic data is archived which can be used in data mining especially supervised learning. However, it is not being fully used due to lack of accurate accident information (labels). In this study, we improve a Mahalanobis distance based algorithm to be able to handle differential data to estimate flow fluctuations and detect accidents and use it to support correcting and complementing accident information. The outlier detection algorithm provides accurate suggestions for accident occurring time, duration and direction. We also develop a system with interactive user interface to realize this procedure. There are three contributions for data handling. Firstly, we propose to use multi-metric traffic data instead of single metric for traffic outlier detection. Secondly, we present a practical method to organise traffic data and to evaluate the organisation for Mahalanobis distance. Thirdly, we describe a general method to modify Mahalanobis distance algorithms to be updatable.

Keywords: *accident data; data labelling; differential distance; Mahalanobis distance; outlier detection; traffic data; updatable algorithm*

Ispravljanje i nadopunjavanje podataja o prometnim nesrećama na autocesti putem Mahalanobis udaljenosti na temelju otkrivanja netipičnih vrijednosti

Pregledni članak

Arhivirana je ogromna količina podataka o prometu koji bi se mogli koristiti za dobivanje specifičnih podataka. Međutim, oni se u potpunosti ne koriste zbog nepostojanja točnih podataka o prometu (oznaka). U ovom radu poboljšavamo algoritam zasnovan na Mahalanobis udaljenosti za procjenu promjena toka prometa i otkrivanje nesreća i primjenjujemo ga kod ispravljanja i dopunjavanja informacija o nesreći. Algoritam za otkrivanje outliera (netipičnih vrijednosti) pruža točne podatke o vremenu događanja nesreće, trajanju i smjeru. Razvijamo i sustav s interaktivnim sučeljem korisnika u svrhu ostvarenja ovog postupka. Predlažu se tri načina za manipulaciju podacima. Najprije, za otkrivanje outliera u prometu predlažemo uporabu multi-metričkih podataka o prometu umjesto jedno metričkih. Nadalje, predlažemo praktičnu metodu za organizaciju prometnih podataka i evaluaciju organizacije Mahalanobis udaljenosti. Kao treće, dajemo opis opće metode za modifikaciju algoritama Mahalanobis udaljenosti kako bi se mogli ažurirati.

Ključne riječi: *algoritam kojeg je moguće ažurirati; diferencijalna udaljenost; Mahalanobis udaljenost; otkrivanje outliera; označavanje podataka; podaci o nesreći; podaci o prometu*

1 Introduction

Nowadays, increasing road traffic is causing more accidents and it gains more attention from authorities. Therefore, a vast number of traffic monitoring devices have been installed to collect traffic data. As a consequence, a huge amount of traffic data has been archived, sometimes together with related information such as accident records [1]. However, patterns and relationships between the traffic data and accident records are invisible. To discover hidden information, the stored huge amount of data can be investigated using data mining techniques [2, 3], especially supervised learning [4, 5] for analysis and prediction. To do this, we need to know related traffic data given an accident record, i.e. labelled data is required. Nevertheless, accident records are neither accurate nor complete. In many authorities' databases, accident occurring time is estimated by the witnesses or drivers and duration of accident time is often missing. What is worse is that the direction of road where accident happened is also missing. Those problems make it impossible to know which archived traffic data is related exactly. Messy accident information is hard to be used directly to label traffic data in supervised learning [6]; even semi-supervised learning needs some initial labels [7]. Thus, it is necessary to correct and complement accident records.

The procedure of manual correction and complementation of accident records requires repeated actions and consumes a lot of time [8]. Besides, it is hard to make decisions regarding accident occurring time, duration and direction from millions of raw traffic values without any external help. To regain the accident related information, in this work we developed a system to help correct and complement accident records. The system will calculate accident occurring time, duration and direction using the accident detection technique that we propose.

Accident detection techniques can be separated into two categories [9]. The first category provides "recognition" of accidents if the monitored traffic situation is similar to previous accidents situation. The second category discovers observations that are significantly different from typical values and this procedure is called outlier detection [10].

The first category includes conventional methods such as McMaster [11] as well as novel machine learning based classification methods [12]. The conventional methods usually require physical location characteristics like shape of the roads or multi-device data as part of the input, and machine learning based classification requires labelled data.

The second category outlier detection compares one observation with normal situation or prediction. There are two types of outliers, global outlier and local outlier [6]. Global outliers are considered as outliers regardless of the concept, whereas local outliers are concept-related. For example, 40 °C temperature is normal in India, but outlier value in Sweden. Much research provided methods to detect outliers while assuming outliers as global for transportation [13]. However, efforts made to adapt local outlier detection for transportation are insufficient.

Further, existing research preferred to use single metric and its threshold to detect outliers. For example, [9, 14] compare monitored flow rate with its threshold. In [15] researchers use speed to do comparison and detection, and [16÷18] use density (occupancy). As there are fundamental differences among characteristics of different roads, procedures considering only single metric are unsuitable. For example, in [19] flow speed is influenced more than flow rate during breakdown events. In [20] events change flow rate suddenly but maintain flow speed. Some data also shows that during certain events, both flow speed and rate can change. However, we cannot find existing research which considers this kind of change in data. Though some work has been done with regard to transit fundamental diagram [20], none considered differential time-varied fundamental diagram. Gonzalez [15] uses the term "level of change" to describe one kind of change, but this change detection is based only on speed. Mentioning differential calculation, the most popular algorithms are ARIMA-like algorithms. Nevertheless, they can handle only single variable and the variable needs to be stationary [21, 22]. Moreover, their related vectorised versions require more assumptions which are not practical [23].

In this paper, we propose to use Mahalanobis distance [24] (M-distance) based analysis to detect accidents. M-distance is a general distance used in multivariate analysis and has been widely used for detecting outliers [25÷27]. However, few works used it for detecting traffic accidents. One possible reason might be that few researchers are considering more than one metric together, so M-distance is not necessary. Even if multivariate data is considered, another problem is M-distance is only for multivariate data that is clustering like filled ellipses, i.e., data is normally distributed [24, 28, 29]. However, traffic data in traditional speed-flow fundamental diagram [30] cannot hold this assumption as shown in Fig. 1.

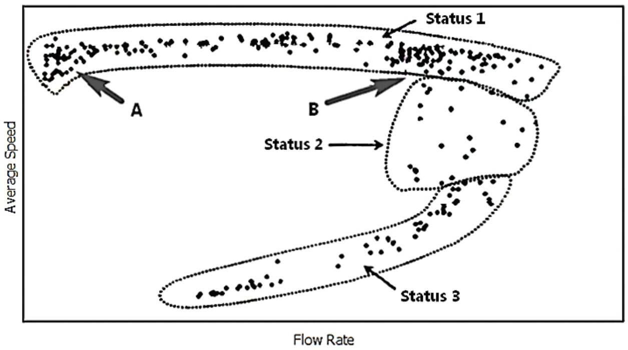


Figure 1 Speed-flow fundamental diagram. Status (dotted circle) 1: undersaturatated flow. 2: queue discarge flow. 3: over-saturated flow. High density area A: typical night time flow. B: day time flow.

The outline of the paper is as follows. In the second section we propose the method to organise traffic data so that it is suitable for M-distance. The third section describes the methodology that detects accidents and complements accident information based on M-distance analysis. The experimental results using real world data are presented in the fourth section. In the fifth section we present the implemented system followed by conclusions in the last section.

2 Data pre-processing

In this section we propose a general method that can pre-process traffic data to be suitable for the main methodology in the third section.

2.1 Metrics selection

Flow rate, speed and density are three fundamental metrics in traffic engineering [30]. It is possible to calculate one metric given both the other two, so two metrics should be considered at the same time. Previous research prefers to consider only one of them within time domain, usually flow rate or speed. We use both flow rate and speed within the time domain.

2.2 Time-separated data organisation

As shown in the density plot of speed-flow fundamental diagram (Fig. 1), data instances (points) are clustering mainly as day time and night time parts with transits between them. It is unsuitable to use M-distance directly in this conventional diagram. One solution is to find a time period to organise data according to time of day. The results are time-separated datasets. The time period should separate different data points into several datasets. Data in each dataset are clustered as filled ellipse. Additionally, the separation should keep the differences between neighbouring datasets ignorable (insignificant) to avoid breaking continuous time series data, otherwise the time period should be changed to a smaller value.

Below is a mathematical description of how hypothesis testing [31] can be used to evaluate time-separated traffic data.

H_0 (null hypothesis): there is no difference among all datasets. Consequently, its competing hypothesis H_1 (alternative hypothesis) is: there are differences among all or some datasets.

In this hypothesis testing, we are considering two-dimensional data with flow rate r and speed$s$. Both r and s are normally distributed in datasets. We now pick two datasets (two groups of data according to different time of day) and name them Dataset 1(D_1) and Dataset 2 (D_2). Thus, we get two distributions for datasets D_1 and D_2 as:

$$(r_1, s_1) \sim \mathcal{N}_q(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \tag{1}$$

$$(r_2, s_2) \sim \mathcal{N}_q(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) \tag{2}$$

where $q = 2$ (2 dimensions); $\boldsymbol{\mu}$ is dataset's centroid and $\boldsymbol{\Sigma}$ is covariance matrix.

According to the properties of operations on independent multivariate normal distributions, a linear combination of multivariate normal distributed variables is still distributed normally:

$$\sum_{i=1}^{n} k_i \boldsymbol{x}_i \sim \mathcal{N}_q(\sum_{i=1}^{n} k_i \boldsymbol{\mu}_i, \sum_{i=1}^{n} k_i^2 \boldsymbol{\Sigma}_i) \tag{3}$$

Therefore, the difference between two neighbouring datasets:

$$\big((r_1, s_1) - (r_2, s_2)\big) \sim \mathcal{N}_2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \tag{4}$$

is a multivariate normal distribution. Now, the null hypothesis can be written as:

$$H_0\colon \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0} \tag{5}$$

i.e.

$$H_0\colon \boldsymbol{\mu}_{D1-D2} = \mathbf{0} \tag{6}$$

which means no difference between two centroids.Meanwhile, the alternative hypothesis is equivalent to:

$$H_1\colon \boldsymbol{\mu}_{D1-D2} \neq \mathbf{0} \tag{7}$$

For that pair of hypotheses, we can calculate p-values by using hypothesis testing and compare the testing results with significance level to know if there are significant differences among datasets. When the separation is done, we can proceed to analyse separated data using the methodology proposed in the next section.

3 New accident Information from outlier detection

In this section we firstly introduce existing works in the first two subsections and then propose our modifications and improvements so that we can correct and complement traffic accident information.

3.1 Mahalanobis distance

Euclidean distance [32] is a widely used traditional and ordinary distance for outlier detection [33, 34]. It is easy to understand, implement and fast to calculate [35]. However, it cannot represent a concept-related distance, as it does not consider the shape of distribution (scatter) [36]. To avoid this weakness, M-distance [24] based analysis is used in this work to detect multivariate outliers.

Here is a brief description of M-distance. Suppose $i\,(1 \leq i \leq n)$ is instance index and $j\,(1 \leq j \leq k)$ is variable index in dataset $\boldsymbol{X} = \left[\left[a_{ij}\right]\right]$ that contains n observations of k variables. The covariance between variable l and variable m is:

$$q_{lm} = \frac{1}{n-1}\sum_{i=1}^{n}(a_{il} - \mu_l)(a_{im} - \mu_m) \tag{8}$$

where μ_l and μ_m are variables' expected values.

Thus the covariance matrix of dataset $\boldsymbol{X}$ can be expressed as a k×k matrix $\boldsymbol{\Sigma} = \left[\left[q_{lm}\right]\right]$.

Finally, M-distance to centroid is the distance between instance $\boldsymbol{x}_i$ and centroid $\boldsymbol{\mu}$:

$$MD_i = \sqrt{(\boldsymbol{x}_i - \boldsymbol{\mu})\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu})^{\mathrm{T}}} \tag{9}$$

Although M-distance can also be used to measure distance between any two points, it stands for "M-distance to centroid" in our work especially.

When being compared with conventional thresholds, Fig. 2 shows that M-distance is suitable for multivariate outlier detection and can provide better thresholds by considering the shape of distribution [37].

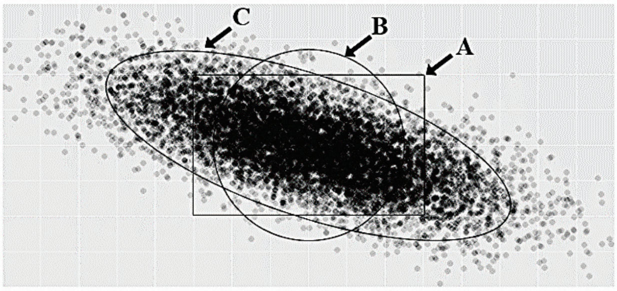
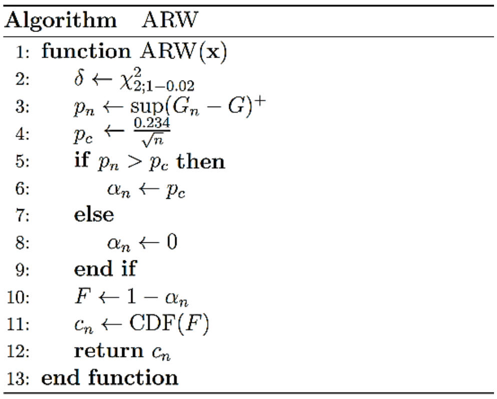


Figure 2 Example of different thresholds. A: the rectangule shows threshold considering each metric separately. B: the circle shows threshold considering two metrics together. C: the ellipse shows M-distance threshold considers two metrics together as well as the shape of distribution.

3.2 Adaptive threshold

Though M-distance considers the shape of distribution, it comes with a shortcoming. When calculating the covariance, M-distance is sensitive to outliers and extremes [29, 38]. To reduce the sensitivity, adaptive reweighted location and scatter estimation [37] (ARW) is used to determine an adaptive threshold c_n. If the distribution function of χ_p^2 is noted as $G(u)$, and the empirical distribution function as $G_n(u)$, where n is the number of observations, ARW can be described as pseudo code below. p_c is used to describe difference between theoretical distribution and empirical distribution.



```
Algorithm ARW
function ARW(x)
    $\delta \leftarrow \chi^2_{2;1-0.02}$
    $p_n \leftarrow \sup(G_n - G)^+$
    $p_c \leftarrow \frac{0.234}{\sqrt{n}}$
    if $p_n > p_c$ then
        $\alpha_n \leftarrow p_c$
    else
        $\alpha_n \leftarrow 0$
    end if
    $F \leftarrow 1 - \alpha_n$
    $c_n \leftarrow \mathrm{CDF}(F)$
    return $c_n$
end function
```

A data instance can be detected as an outlier if its M-distance is bigger than the adaptive threshold. As estimating the centroid and scatter consumes a vast amount of computing resources, minimum covariance determinant [39] (MCD) is used to calculate centroid and scatter.

Outliers can now be detected by comparing time-separated data's M-distance with adaptive threshold, and those outliers are time-separated outliers.

3.3 Differential outlier

Though flow rate and speed are considered together within time domain, the detection is not using a nature of the time domain that is differential characteristic. Thus, we also propose to use differential data to detect outliers. For differential calculation, we can get the differential data from two consecutive instances. That is, the differential data is a result of subtracting one data instance with its neighbour in a consecutive time series.

If we note each instance as $\boldsymbol{x}_i = [r_i, s_i]$, we can get the differential data as:

$$\boldsymbol{x}_{diff} = \boldsymbol{x}_{i+1} - \boldsymbol{x}_i = [r_{i+1} - r_i, s_{i+1} - s_i] \qquad (10)$$

For example, if there are originally 100 instances, we can have 99 differential instances. Then the differential data is divided by time of day according to the time stamp of instance$\boldsymbol{x}_i$. Each differential dataset has almost five thousand instances. The remaining analysis procedure is the same as normal outlier detection. Each differential dataset was analysed using M-distance to detect outliers. Now the detected outliers are time-separated differential outliers.

3.4 Monthly updatable algorithm

The aforementioned outlier and differential outliers are detected by comparing data instances with thresholds calculated from all archived data. Therefore, the model that is being compared with is not timely dynamic. This might cause two time-related problems. Firstly, if some weeks or months are special, the algorithm may behave unexpectedly. Secondly, the algorithm cannot reflect the fact that more and more cars are coming on the road nowadays gradually. To solve those problems, one solution is to calculate weighted instances in ARW and give more weighting to recent instances. This solution requires the whole archived data and increases calculation cost. Another solution is to use updatable method.

Updatable algorithms [40] can produce a new result from the latest old result and new data without old data, which can dramatically reduce calculation time. Thus, we improved the original algorithm to be updatable which will consider recent data instances more than other history instances when deciding if the new instance is an outlier. In addition, we weighted the influence of old data by limiting the number of old instances, so the new result gets adjusted according to the new data dynamically.

Below is a description of the improved algorithm. Consider $\boldsymbol{X}_1 = [\boldsymbol{r}_1, \boldsymbol{s}_1]$ containing n_1 old instances and $\boldsymbol{X}_2 = [\boldsymbol{r}_2, \boldsymbol{s}_2]$ containing n_2 newly arrived instances, we can use the existing covariance matrix of $\boldsymbol{X}_1$ and instances in $\boldsymbol{X}_2$ to detect updatable outliers. If the old covariance matrix is noted as:

$$\boldsymbol{C}_1 = \begin{bmatrix} \mathrm{COV}(\boldsymbol{r}_1, \boldsymbol{r}_1) & \mathrm{COV}(\boldsymbol{r}_1, \boldsymbol{s}_1) \\ \mathrm{COV}(\boldsymbol{s}_1, \boldsymbol{r}_1) & \mathrm{COV}(\boldsymbol{s}_1, \boldsymbol{s}_1) \end{bmatrix} \qquad (11)$$

and the covariance matrix for new data is:

$$\boldsymbol{C}_2 = \begin{bmatrix} \mathrm{COV}(\boldsymbol{r}_2, \boldsymbol{r}_2) & \mathrm{COV}(\boldsymbol{r}_2, \boldsymbol{s}_2) \\ \mathrm{COV}(\boldsymbol{s}_2, \boldsymbol{r}_2) & \mathrm{COV}(\boldsymbol{s}_2, \boldsymbol{s}_2) \end{bmatrix} \qquad (12)$$

then overall updatable covariance matrix is:

$$\boldsymbol{C}_0 = \begin{bmatrix} \mathrm{COV}(\boldsymbol{r}_0, \boldsymbol{r}_0) & \mathrm{COV}(\boldsymbol{r}_0, \boldsymbol{s}_0) \\ \mathrm{COV}(\boldsymbol{s}_0, \boldsymbol{r}_0) & \mathrm{COV}(\boldsymbol{s}_0, \boldsymbol{s}_0) \end{bmatrix} \qquad (13)$$

where $\boldsymbol{r}_0{}^{\mathrm{T}} = [\boldsymbol{r}_1{}^{\mathrm{T}} \boldsymbol{r}_2{}^{\mathrm{T}}]$, $\boldsymbol{s}_0{}^{\mathrm{T}} = [\boldsymbol{s}_1{}^{\mathrm{T}} \boldsymbol{s}_2{}^{\mathrm{T}}]$.

In accordance with Eq. (8), we derived:

$$\begin{aligned} &\mathrm{COV}(\boldsymbol{r}_0, \boldsymbol{s}_0) \\ &= \mathrm{COV}\left(\begin{bmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{bmatrix}, \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix}\right) \\ &= \frac{n_1 - 1}{n_0 - 1} \times \mathrm{COV}(\boldsymbol{r}_1, \boldsymbol{s}_1) + \frac{n_1 \times (n_0 - n_1) \times \mu_{\mathrm{r}_1} \times \mu_{\mathrm{s}_1}}{n_0 \times (n_0 - 1)} \\ &+ \frac{n_2 - 1}{n_0 - 1} \times \mathrm{COV}(\boldsymbol{r}_2, \boldsymbol{s}_2) + \frac{n_2 \times (n_0 - n_2) \times \mu_{\mathrm{r}_2} \times \mu_{\mathrm{s}_2}}{n_0 \times (n_0 - 1)} \\ &- \frac{n_1 \times n_2 \times (\mu_{\mathrm{r}_1} \times \mu_{\mathrm{s}_2} + \mu_{\mathrm{r}_2} \times \mu_{\mathrm{s}_1})}{n_0 \times (n_0 - 1)} \end{aligned} \qquad (14)$$

where $n_0 = n_1 + n_2$. The other three items in $\boldsymbol{C}_0$ can also be calculated using similar equations. $(\mu_{\mathrm{r}_1}, \mu_{\mathrm{s}_1})$ and $(\mu_{\mathrm{r}_2}, \mu_{\mathrm{s}_2})$ are centroids of old and new data instances respectively. Therefore, new updatable centroid is calculated as:

$$\begin{aligned} &(\mu_{\mathrm{r}_0}, \mu_{\mathrm{s}_0}) \\ &= \left(\frac{n_1 \times \mu_{\mathrm{r}_1} + n_2 \times \mu_{\mathrm{r}_2}}{n_0}, \frac{n_1 \times \mu_{\mathrm{s}_1} + n_2 \times \mu_{\mathrm{s}_2}}{n_0}\right) \end{aligned} \qquad (15)$$

To take advantage of this new algorithm, time series data can be grouped by a time gap, say a month, and then each group's centroid, scatter and threshold can be calculated separately. Hence, we get updatable M-distance based algorithm that leads to Updatable Time-Separated Outliers and Updatable Time-Separated Differential Outliers.

3.5 Accident occurring time, duration and direction

Through previous calculation, we already have four different outliers, and we need to select suitable ones to use according to real world data. For one data instance, the system will calculate its M-distance and then divide by adaptive threshold. If the quotient is bigger than threshold, and its timestamp is near selected accident record, it is the accident occurring time. Otherwise, the biggest quotient should be used.

The outlier following occurring time is accident ending time which means the traffic starts to recover from the accident. Sometimes traffic situation will resume from one accident gradually and the second outlier is unobvious, then the largest quotient related timestamp in three hours after the accident occurring time will be considered as road cleaning time. This is due to the fact that most accident duration is less than three hours [41].

Accident direction is given by accident indicator which measures the deviation of accident traffic from non-accident traffic (in Fig. 3).

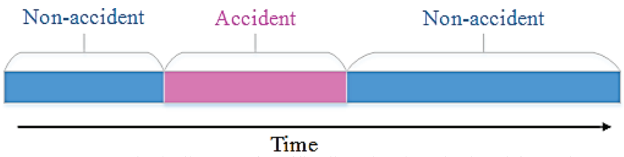


Figure 3 The indicator of traffic direction is calculated from the difference between accident traffic and non-accident traffic.

The indicator is defined as:

$$indicator = \frac{\mathrm{MD}(\boldsymbol{\mu}_{\mathrm{accident}} - \boldsymbol{\mu}_{\mathrm{non.accident}})}{c_n} \qquad (16)$$

Centroids of traffic during accident time $\boldsymbol{\mu}_{\text{accident}}$ and non-accident time $\boldsymbol{\mu}_{\text{non.accident}}$ are calculated respectively. The difference between those two centroids, i.e., differential centroid, is analysed using differential outlier detection. The M-distance is compared with adaptive threshold to get accident indicator. The biggest indicator gives the detection device as well as the direction of accident.

3.6 Proposed steps

The aforementioned procedure is robust to outliers, also adaptive to both degree of freedom and number of instances.

Our work uses the procedure to analyse traffic data as shown in Fig. 4.

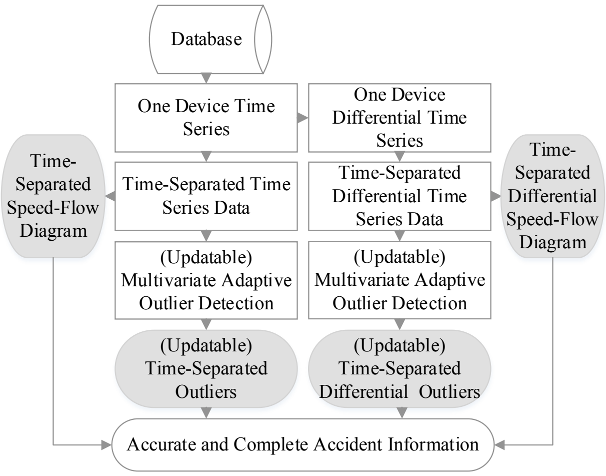

Database
One Device Time Series
One Device Differential Time Series
Time-Separated Speed-Flow Diagram
Time-Separated Time Series Data
Time-Separated Differential Time Series Data
Time-Separated Differential Speed-Flow Diagram
(Updatable) Multivariate Adaptive Outlier Detection
(Updatable) Multivariate Adaptive Outlier Detection
(Updatable) Time-Separated Outliers
(Updatable) Time-Separated Differential Outliers
Accurate and Complete Accident Information

Figure 4 Main steps of the proposed method. The data is analysed device by device. In-process outcomes (grey coloured) are two types of diagrams and four types of outliers. Final outcome is accurate and complete accident information.

Firstly, the system queries the data from database. The original time series is processed according to the left side steps while the differentialized time series is processed according to the right side steps. Secondly, the time series will be separated into several datasets according to data's timestamps. The results can be visualized as two types of diagrams. Thirdly, the time-separated data will be analysed. Thereafter, both updatable and non-updatable algorithms are used to detect outliers. Thus, non-differential and differential data lead to four types of outliers. Finally, the system considers those outliers together with the inaccurate or incomplete accident records to produce accurate and complete accident information as suggestions to fix the record.

4 Experiments and results

In this section, the proposed steps are used to process real world data. Our source of data comes from devices monitoring a freeway named Kunshi in Southwest China (in Fig. 5). The freeway is 70 kilometres long and thirty devices are collecting traffic data. The devices report traffic statistics every 5 minutes. Each reported record contains total number of cars in 5 minutes and time averaged speed during the same period. The data is from April 2013 to May 2014, and more than six million records are stored in the database.

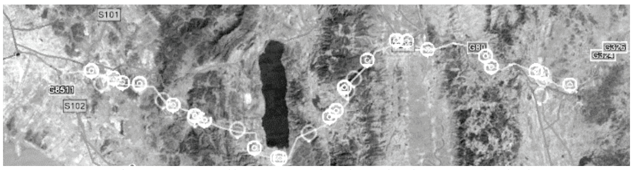


Figure 5 There are totally 30 monitoring devices as circled on map. Multi circle icons may be displayed as one when close to each other.Total length of monitored freeway is70 km.

4.1 Cleaning data

The raw data needs to be cleaned before data pre-processing. For each monitoring device, there can be one or more cameras and each camera will report records of one lane statistic data independently. This brings two problems. Firstly, one monitoring device generates multiple records according to lanes and those records should be aggregated. Secondly, the arriving timestamps of reported records from different cameras might have several seconds' delay which makes it harder to find the right records to aggregate. After analysing the raw data, we found that the records from one device have the same timestamp until minute level but different at the second level. We aggregate the records according to timestamps until minute level to get data instances. The aggregated results show that this method is working correctly when being compared with raw data.

4.2 Hourly data as time-separated data

Among collected traffic metrics, flow rate and speed are used in our analysis as proposed in the second section. Cleaned data is plotted in Fig. 6. The road usually carries under-saturated flow that is part of the conventional speed-flow fundamental diagram in Fig. 1.

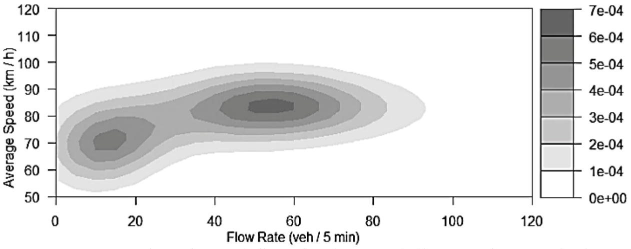


Figure 6 Density of speed-flow fundamental diagram for one device data during the whole day. The distribution cannot hold M-distance assumptions.

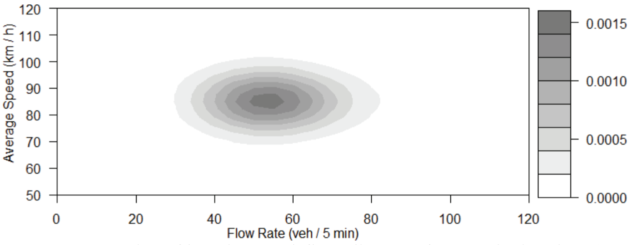
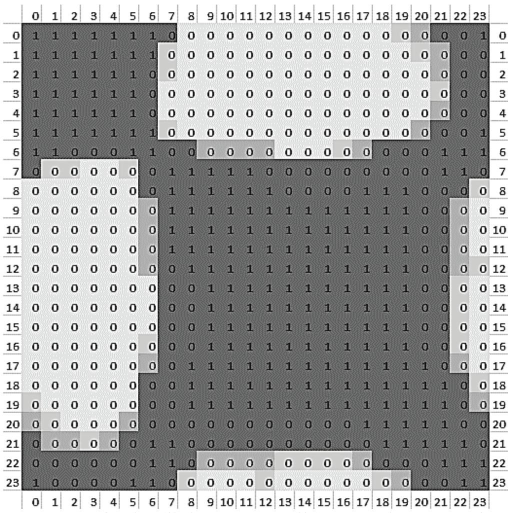


Figure 7 Density of hourly speed-flow diagram for one device data (10 AM to 11 AM). The distribution can hold M-distance assumptions.

The data then is ready for organisation using the method in section 2.2. We found that dividing data into datasets by "hour of day" is suitable for M-distance based algorithm. Each dataset of one device has about five thousand instances. Fig. 7 shows an example that the density plot which shows that the data points from a device during one hour are clustering as an ellipse and suitable for M-distance to use.

By using inferential statistics and hypothesis testing, we get p-values of differences among all hourly datasets. As shown in Fig. 8, there are significant differences among 30% of those hourly dataset pairs, so it is necessary to analyse data according to its hour of day. Thus, the null hypothesis H_0 is rejected and H_1 is accepted. Therefore, the data needs to be divided according to its hour of day before analysis.

On the other hand, no pairs of neighbouring hourly datasets are significantly different (all dark cells are connected). Which means the transits between neighboured hourly datasets are smooth. Therefore, choosing one hour as the time gap not only separates different data but also keeps the transits smooth.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	
0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
3	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
4	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
5	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5
6	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	6
7	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	7
8	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	8
9	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	9
10	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	10
11	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	11
12	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	12
13	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	13
14	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	14
15	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	15
16	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	16
17	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	17
18	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	18
19	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	19
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	20
21	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	21
22	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	22
23	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	23
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	

Figure 8 30% of pairs (lightest grey cells) are under or equal to 0.02 (values are rounded to 1 or 0), so data should be seperated by hour of day before analysis. 35% are under or equal to 0.05. 41% are under or equal to 0.1 (darkest cells), which means transits of hourly data are smooth and suitable.

4.3 Different outliers in speed-flow diagram

After data organisation, M-distance analysis introduced in the third section is used to detect outliers.

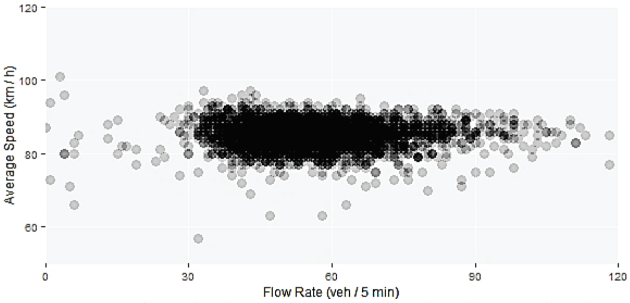


Figure 9 Before adaptive M-distance outlier detection, one hourly dataset in speed-flow diagram (10 AM to 11 AM).

Below is an example of analyses result from a dataset that contains data from 10 AM to 11 AM. Fig. 9 is the dot plot of Fig. 7.

After applying adaptive M-distance outlier detection, we calculated outliers in hourly speed-flow fundamental diagrams. Non-hourly outliers and hourly outliers are plotted respectively with their thresholds.

When compared to non-hourly outliers and threshold (in Fig. 10), hourly outliers (in Fig. 11) are more reasonable.

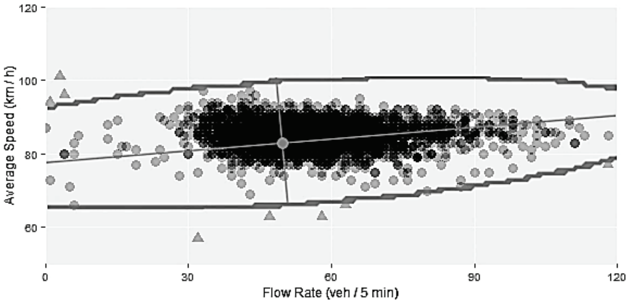


Figure 10 After using all data (non-hourly data) in adaptive M-distance outlier detection. Non-hourly outliers are marked as triangles in speed-flow diagram, unacceptable poor quality (10 AM to 11 AM).

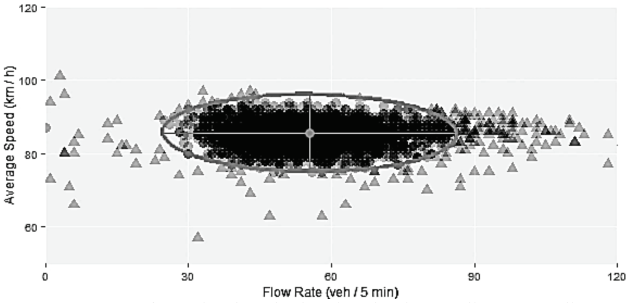


Figure 11 After using hourly data in adaptive M-distance outlier detection. Hourly outliers are marked as triangles in speed-flow diagram. Quality is better than non-hourly outliers (10 AM to 11 AM).

4.4 Hourly differential outlier in speed-flow diagram

Using differential calculation, we can get differential datasets. The differential data points are clustering as a cross shape instead of ellipse, so it is not possible to use M-distance directly. However, if we plot hourly differential datasets, we can see the expected ellipse (in Fig. 12).

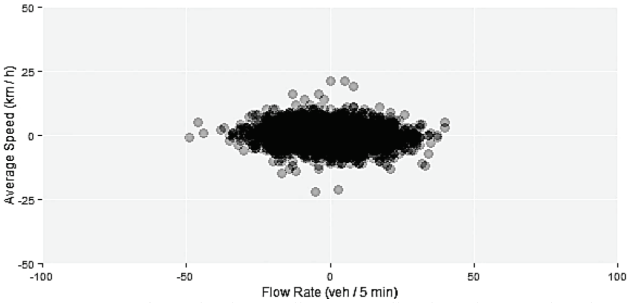


Figure 12 Before adaptive M-distance outlier detection, one hourly differential dataset in differential speed-flow diagram (10 AM to 11 AM).

Applying hourly adaptive M-distance analysis, we finally calculated hourly differential outliers. When being compared to non-hourly differential outliers (in Fig. 13), hourly differential outliers in differential speed-flow diagram (Fig. 14) show improvement that the threshold ellipse is more reasonable.

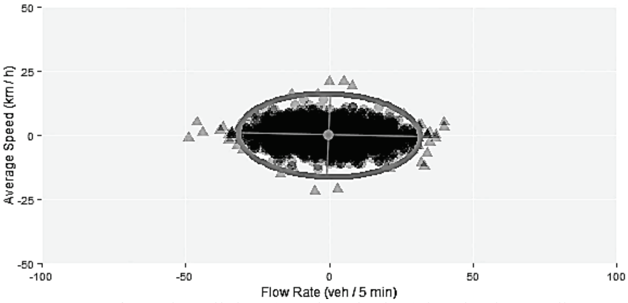


Figure 13 After using all data (non-hourly data) in adaptive M-distance outlier detection. Non-hourly differential outliers are marked as triangles in differential speed-flow diagrams (10 AM to 11 AM).

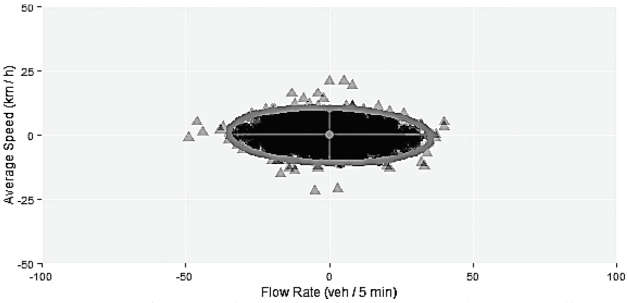


Figure 14 After using hourly data in adaptive M-distance outlier detection. Hourly differential outliers are marked as triangles in differential speed-flow diagrams. Quality is slightly improved from non-hourly differential outliers (10 AM to 11 AM).

4.5 Different outliers in time series

We can see outliers in speed-flow diagrams, but it is hard for analysts to use. Time-series plot is necessary for further analysis. Fig. 15 displays several hours traffic situation on a holiday. The hourly outliers spread over this period. Instead, hourly differential detection is more stable and accident is detected correctly.

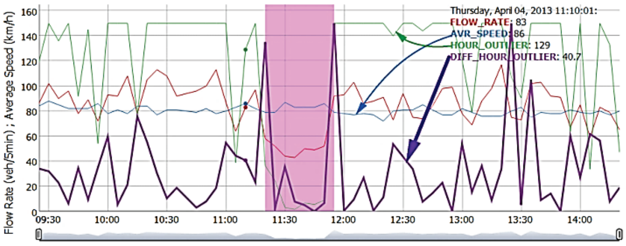


Figure 15 Hourly differential outlier can detect accident correctly. It is robust to extreme high traffic during holidays. Detected accident duration is shaded. Outlier threshold is set to 100.

In our system, one month length is used in updatable algorithm to make sure there is enough data to be analysed to update centroids and thresholds. One important thing to notice is that n_1 will be limited to maximum 360 which is product of 12 instances per hour and 30 days per month, i.e., the old result has less weighting than the new one.

Performance of the updatable algorithm is usually similar as the original one, but it performs better when there is a change for monthly traffic data. For example, when the biggest annual festival came earlier (the festival is based on lunar calendar), updatable detection algorithm adjusts to this change and can stay below threshold, while the original hourly outlier detection algorithm cannot get used to this situation and gives many outliers (Fig. 16). Outlier displayed threshold is enlarged from 1 to 100 as well as the quotient of data's M-distance and threshold to ease visualisation. For the same reason, displayed maximum quotient values are limited to 150.

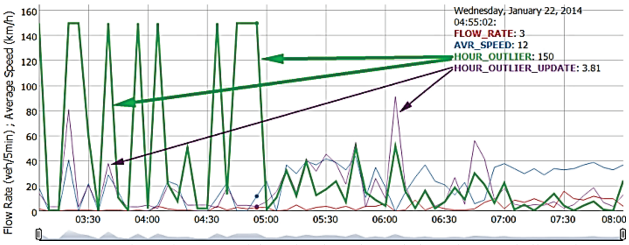


Figure 16 There is no accident in the plotted duration and direction. Updatable hourly outlier is adopting to traffic situation change and more accurate than normal hourly outlier. Outlier threshold is set to 100.

Besides, updatable differential hourly outlier performs well even during abnormal fluctuation in holidays (Fig. 17).

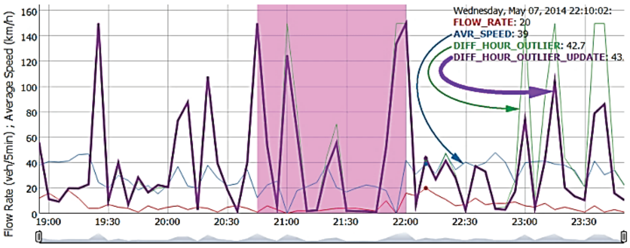


Figure 17 Updatable differential hourly outliers can detect the accident, and stay stable after the accident compared with non-updatable ones. Detected accident duration is shaded. Outlier threshold is set to 100.

5 Interactive user interface

On one hand, R [42] provides a multiplatform commonly used environment [43, 44] to perform state-of-the-art data analysis [45] (R environment is used throughout this paper). On the other hand, the speed-flow diagrams (Fig. 9 ÷ Fig. 14) are hard to understand and use, it is necessary to have a rich-media user interface to provide illustrative information for analysts. Web-based system is widely used for data monitoring and visualization due to its excellent display effect and user-friendly interface [46÷48]. Shiny [49] is R's web framework and both of them are available as open source software under GNU General Public License. Based on the proposed algorithms, we developed an interactive system for analysts using R and Shiny.

As shown in Fig. 18, the main panel has four areas. The first area is used to select which panel to display.

The source area contains three subareas (Fig. 19). In metric selection subarea, the analyst selects some metrics that should be displayed. Flow rate, speed and two updatable types of outliers are selected by default, the other two types of non-updatable outliers are optional. Then the analyst selects one accident that needs to be investigated from accident selection subarea. According to the selected accident, the system finds out the nearest four devices for both directions. In addition, two buttons named "previous accident" and "next accident" can be used to navigate among accidents when it is necessary to go through the accidents one by one. Besides using accidents, the analyst can also manually fill a time point and a place in manual input subarea for the system to pick related devices.

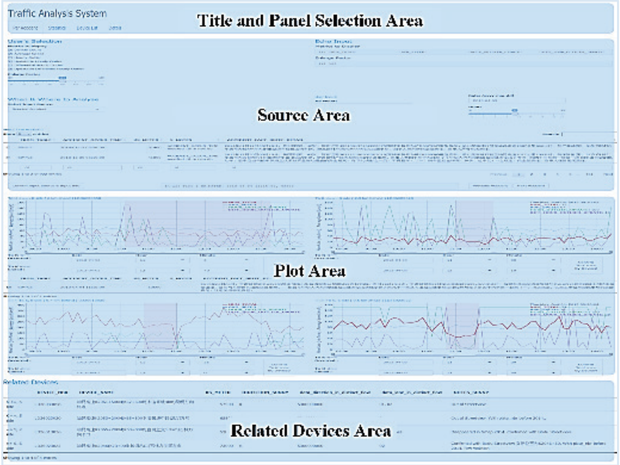


Figure 18 Main panel. It contains four areas. Details are explained in the following zoomed in figures.

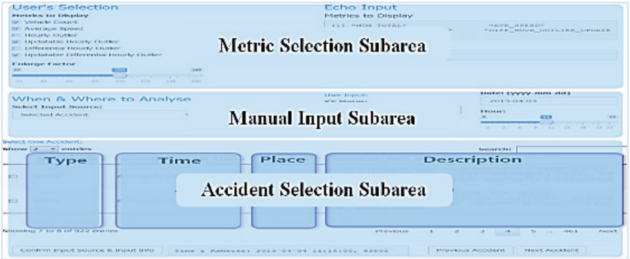
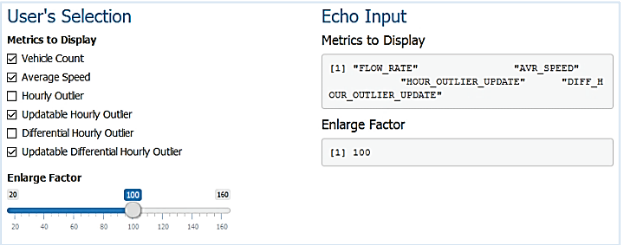


Figure 19 The Source Area structure. It contains three subareas and the details are shown in the following three figures.

On the top left of Metric Selection Subarea, there are several metrics that are ready to be displayed in the plot. As mentioned in the previous chapter, different lanes carry different level of flow, to ease the view of outlier displayed threshold, an enlarge factor is used. All the selections are echoed from server to make sure the actions are right.



User's Selection

Metrics to Display

- ☑ Vehicle Count
- ☑ Average Speed
- □ Hourly Outlier
- ☑ Updatable Hourly Outlier
- □ Differential Hourly Outlier
- ☑ Updatable Differential Hourly Outlier

Enlarge Factor

Echo Input

Metrics to Display

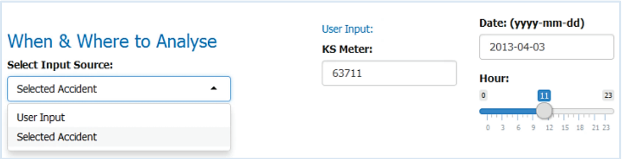
```
[1] "FLOW_RATE"                    "AVR_SPEED"
            "HOUR_OUTLIER_UPDATE"        "DIFF_H
OUR_OUTLIER_UPDATE"
```

Enlarge Factor

```
[1] 100
```

Figure 20 Metric Selection Subarea for Data Source Area.

The figure below shows Manual Input Subarea. For "user manual input source" mode, the system requires a rough milestone position and a rough time.



When & Where to Analyse

Select Input Source: Selected Accident

User Input:

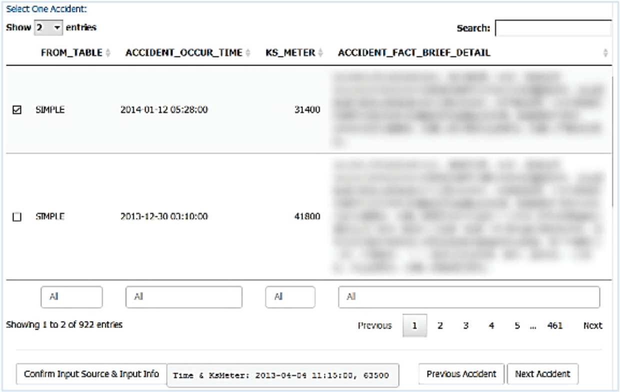
KS Meter: 63711

Date: (yyyy-mm-dd) 2013-04-03

Hour: 11

Figure 21 Manual Input Subarea for Data Source Area. The left side is the switch of source "user manual input" or "selected accident".The right side is the manual input of milestone and time.

For "selected accident source" mode, Fig. 22 below shows the accidents to be selected. The accident information includes the type of accident (archived in different tables), raw recorded accident time, milestone position and brief accident fact (which is blurred for privacy).



Select One Accident:

Show 2 entries   Search:

	FROM_TABLE	ACCIDENT_OCCUR_TIME	KS_METER	ACCIDENT_FACT_BRIEF_DETAIL
☑	SIMPLE	2014-01-12 05:28:00	31400	[redacted]
□	SIMPLE	2013-12-30 03:10:00	41800	[redacted]
	All	All	All	All

Showing 1 to 2 of 922 entries

```
Time & KsMeter: 2013-04-04 11:15:00, 63500
```

Figure 22 Accident Selection Subarea for Data Source Area. It is the most important subarea as the selection leads to four related devices. The information shown here includes accident’s type, raw time, position and description, which helps users to idenfy the correct accident.

Given the time and four devices from previous step, related data is analysed by the system and the results are displayed in the plot area (Fig. 23). The data from two devices in upstream direction is shown in the top subarea, one device is ahead of accident point and the other behind. The data from downstream devices is displayed in the bottom subarea. Then the selected accident is displayed in the middle to ease the analyst's comparison with plots.

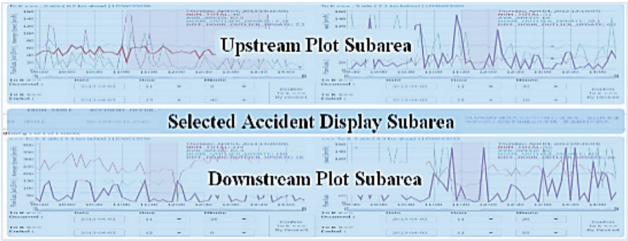
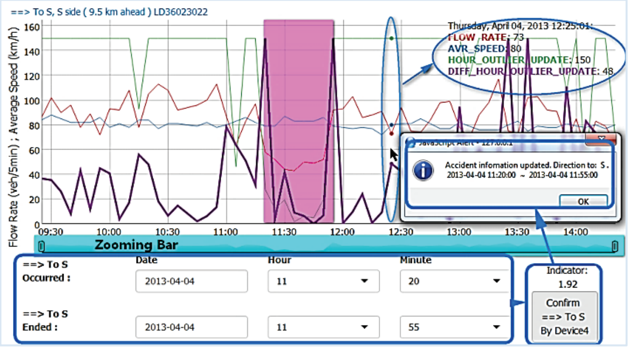


Figure 23 Plot Area. It contains three subareas. The selected accident is displayed in the middle, surrounded with related traffic data from four related devices. The Selected Accident Display Subarea shows the same information as Accident Selection Subarea from source as a confirmation, and the plot is shown in detail in the next figure.



Accident information updated. Direction to: S.
2013-04-04 11:20:00 ~ 2013-04-04 11:55:00

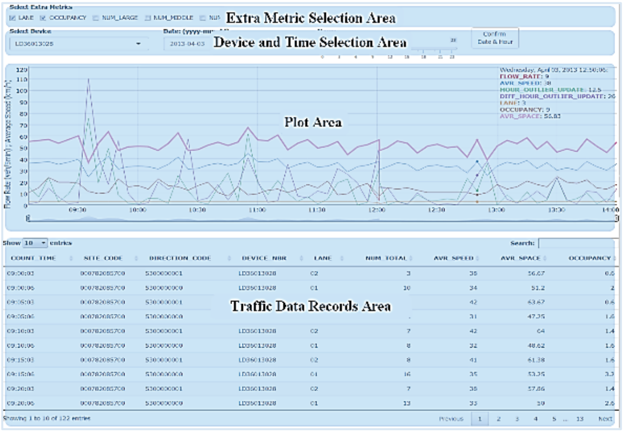
	Date	Hour	Minute
==> To S Occured :	2013-04-04	11	20
==> To S Ended :	2013-04-04	11	55

Indicator:
1.92

Figure 24 One device's analysis result plot, part of plot area.Detected accident duration is shaded (11:20 to 11:55). Top-right values follows mouse hovering position. Zooming in or out can be done using zooming bar or direct mouse-drag selection. The four plots are syncronized for the zooming rate, which means zooming one plot will also let other three plots zoom to the same rate.

Taking one plot as an example (Fig. 24), the system suggests a new occurring time and duration for the selected accident shown as the shaded area. The analyst can check metric values by moving and hovering mouse. Then the analyst can tune the system suggested information and confirm it. Hence, there is an accurate accident occurring time. Two new attributes are accident duration and direction. Based on the four accident indicators, the analyst can choose the biggest one which means the most obvious detection. Accident information is updated when the analyst confirms the analysis results. When the system promotes confirmation window, those attributes are stored in addition to the existing ones.

The plot title indicates only device distance and device ID, more detailed information such as number of monitored lanes, device type, installation place and surrounding environment is shown in the related devices area below the plot area.


Select Extra Metrics
☑ LANE ☑ OCCUPANCY □ NUM_LARGE □ NUM_MIDDLE □ NU
Extra Metric Selection Area
Select Device
LD36013028
Date (yyyy-mm-dd)
2013-04-03
Device and Time Selection Area
Confirm Date & Hour
Wednesday, April 03, 2013 12:50:06:
FLOW_RATE: 9
AVR_SPEED: 38
HOUR_OUTLIER_UPDATE: 12.5
DIFF_HOUR_OUTLIER_UPDATE: 26
LANE: 3
OCCUPANCY: 9
AVR_SPACE: 55.83
Plot Area
Flow Rate (veh/5min) ; Average Speed (km/h)
120
110
100
90
80
70
60
50
40
30
20
10
0
09:30
10:00
10:30
11:00
11:30
12:00
12:30
13:00
13:30
14:00
Traffic Data Records Area

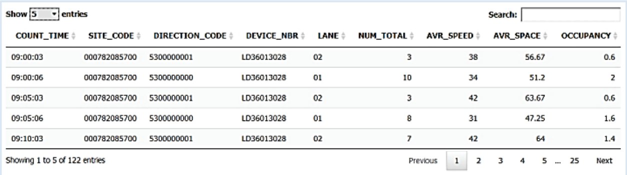
Show 10 entries Search:

COUNT_TIME	SITE_CODE	DIRECTION_CODE	DEVICE_NBR	LANE	NUM_TOTAL	AVR_SPEED	AVR_SPACE	OCCUPANCY
09:00:03	000782085700	5300000001	LD36013028	02	3	38	56.67	0.6
09:00:06	000782085700	5300000000	LD36013028	01	10	34	51.2	2
09:05:03	000782085700	5300000001				42	63.67	0.6
09:05:06	000782085700	5300000000				31	47.25	1.6
09:10:03	000782085700	5300000001	LD36013028	02	7	42	64	1.4
09:10:06	000782085700	5300000000	LD36013028	01	8	32	48.62	1.6
09:15:03	000782085700	5300000001	LD36013028	02	8	41	61.38	1.6
09:15:06	000782085700	5300000000	LD36013028	01	16	35	53.25	3.3
09:20:03	000782085700	5300000001	LD36013028	02	7	38	57.86	1.4
09:20:06	000782085700	5300000000	LD36013028	01	13	33	50	2.6

Showing 1 to 10 of 122 entries Previous 1 2 3 4 5 ... 13 Next

Figure 25 Details panel. It contains four areas, including a detailed plot for the selected device and un-aggregated raw traffic statistic records. The plot is simplified from the main panel by removing accident time span, but with more metrics like lane, occupancy, average space, types of vehicles that are selected from Extra Metric Selection Area on the top. Traffic Data Records Area is zoomed in and shown in details in the next figure.

During the above procedure, if the analyst wants to analyse the data deeply, "details" panel is useful. There are four areas in this panel (Fig. 25). The top area provides extra metrics such as monitored lane, occupancy, average space and flow rate of different types of vehicles. The analyst can select a specific device and time in the second area. The plot area then visualises time series which is similar with the plot area in the main panel but with more details using extra metrics as well as the main panel's basic metrics. The bottom area shows raw traffic statistic records before aggregation which can be checked when there is suspicious device malfunctioning.


Show 5 entries Search:

COUNT_TIME	SITE_CODE	DIRECTION_CODE	DEVICE_NBR	LANE	NUM_TOTAL	AVR_SPEED	AVR_SPACE	OCCUPANCY
09:00:03	000782085700	5300000001	LD36013028	02	3	38	56.67	0.6
09:00:06	000782085700	5300000000	LD36013028	01	10	34	51.2	2
09:05:03	000782085700	5300000001	LD36013028	02	3	42	63.67	0.6
09:05:06	000782085700	5300000000	LD36013028	01	8	31	47.25	1.6
09:10:03	000782085700	5300000001	LD36013028	02	7	42	64	1.4

Showing 1 to 5 of 122 entries Previous 1 2 3 4 5 ... 25 Next

Figure 26 Traffic Data Records Area. One record contains several fields like timestamp, ID (including numbering of site, device, direction and lane), flow rate, speed and occupancy. More fields such as statistics for different type of vehicles can be shown instead of ID's considering display space.

6 Conclusion and future work

In this research, we propose to use multi-metric data instead of commonly used single metric data for traffic analysis. We also introduce a general method to pre-process traffic data to be suitable for multivariate M-distance based algorithm. In this process, we introduce the importance of differential distance. Then we modify the algorithm to be updatable and describe the methodology to detect accident and correct and complement accident data. Finally, based on proposed algorithm, we develop a system with illustrative and interactive user interface to visualize different outliers in time domain and help to fix accident information efficiently.

One issue should be concerned during data organisation, which is that the hourly datasets may cluster as ovals instead of eclipses. In that situation, evaluation of eclipse shape and appropriate transformation, such as logarithmic transformation or exponential transformation, are recommended. In addition, due to the lack of accurate flow-accident data, we are not able to statistically compare fluctuation estimation and accident detection performance with other distance types and algorithms. The usages and issues of proposed methodology and procedure will be investigated in future work, for instance traffic analysis and prediction using supervised learning.

Acknowledgements

We want to thank the editors of Technical Gazette for their efficient response, and reviewers for their time and contribution. We are thankful to get support from Florian Westphal with proof reading and advices.

We also thank authors and editors of referred papers for their important academic basis. Fig. 1 is reprinted by permission of Pearson Education, Inc., Upper Saddle River, New Jersey.

This work was supported by National Natural Science Foundation of China (NSFC), No. 61364019.

7 References

[1] Guo, J.; Huang, W.; Williams, B. M. Real time traffic flow outlier detection using short-term traffic conditional variance prediction. // Transportation Research Part C: Emerging Technologies, 50(2015), pp. 160–172. https://doi.org/10.1016/j.trc.2014.07.005

[2] Xuesong, W.; Qiang, G.; Shanshan, L.; Rongfei, C. Design and Implementation of School Hospital Information Analysis and Mining System. // Applied Science, Materials Science and Information Technologies in Industry, 513(2014), pp. 498–501.

[3] Prasad, N.; Kumar, P.; Naidu, M. M. An Approach to Prediction of Precipitation Using Gini Index in SLIQ Decision Tree. // Fourth International Conference on Intelligent Systems, Modelling and Simulation / Bangkok, 2013, pp. 56–60. https://doi.org/10.1109/ISMS.2013.27

[4] Bhatt, A. S. Comparative Analysis of Attribute Selection Measures Used for Attribute Selection in Decision Tree Induction. // International Conference on Radar, Communication and Computing / SKP Engineering College Tiruvannamalai, 2012, pp. 230–234.

[5] Using Supervised Learning and Comparing General and ANTI-HIV Drug Databases Using Chemoinformatics. // Pattern Recognition and Machine Intelligence, Proceedings

/ Taneja Shweta; Raheja Shipra; Kaur Savneet. Berlin: Springer-Verlag, 2009, pp. 177–183.

[6] Jiawei, H.; Kamber, M. Data mining: concepts and techniques, 3rd ed. Singapore: Elsevier, 2012.

[7] Zhao, M.; Zhan, C.; Wu, Z.; Tang, P. Semi-Supervised Image Classification Based on Local and Global Regression. // IEEE Signal Processing Letters, 22, 10(2015), pp. 1666–1670. https://doi.org/10.1109/LSP.2015.2421971

[8] Honig, A.; Howard, A.; Eskin, E.; Stolfo, S. Adaptive Model Generation: An Architecture for Deployment of Data Mining-based Intrusion Detection Systems. // Applications of data mining in computer security, 8720(2002), pp. 153–194. https://doi.org/10.1007/978-1-4615-0953-0_7

[9] Thomas, T.; Van Berkum, E. C. Detection of incidents and events in urban networks. // IET Intelligent Transport Systems, 3, 2(2009), pp. 198–205. https://doi.org/10.1049/iet-its:20080045

[10] Anomaly Detection. // Introduction to Data Mining / Pang-Ning Tan; Michael Steinbach; Vipin Kumar. Boston: Pearson Addison Wesley, 2006, pp. 651–665.

[11] Hall, F. L.; Shi, Y.; Atala, G. On-line testing of the McMaster incident detection algorithm under recurrent congestion. // Transportation Research Record, 1394(1993), pp. 1–7.

[12] Yuan, F.; Cheu, R. L. Incident detection using support vector machines. // Transportation Research Part C-Emerging Technologies, 11, 3(2003), pp. 309–328. https://doi.org/10.1016/S0968-090X(03)00020-2

[13] Ghosh-Dastidar, S.; Adeli, H. Wavelet-Clustering-Neural Network Model for Freeway Incident Detection. // Computer-Aided Civil and Infrastructure Engineering, 18, 5(2003), pp. 325–338. https://doi.org/10.1111/1467-8667.t01-1-00311

[14] Ahmed, F.; Hawas, Y. E. A Threshold-Based Real-Time Incident Detection System for Urban Traffic Networks. // Transport Research Arena, 48(2012), pp. 1713–1722. https://doi.org/10.1016/j.sbspro.2012.06.1146

[15] Gonzalez, H.; Han, J.; Ouyang, Y.; Seith, S. Multidimensional Data Mining of Traffic Anomalies on Large-Scale Road Networks. // Transportation Research Record, 2215, 1(2011), pp. 75–84. https://doi.org/10.3141/2215-08

[16] Lin, W. H.; Daganzo, C. F. A Simple Detection Scheme for Delay-Inducing Freeway Incidents. // Transportation Research Part A-Policy and Practice, 31, 2(1997), pp. 141–155. https://doi.org/10.1016/S0965-8564(96)00009-2

[17] Stephanedes, Y. J.; Chassiakos, A. P. Freeway incident detection through filtering. // Transportation Research Part C: Emerging Technologies, 1, 3(1993), pp. 219–233. https://doi.org/10.1016/0968-090X(93)90024-A

[18] Wang, Y.; Zhang, C. Alternative Route Strategy for Emergency Traffic Management Based on Its: A Case Study of Xi'an Ming City Wall. // Tehnicki Vjesnik-Technical Gazette, 20, 2(2013), pp. 359–364.

[19] Yeon, J.; Hernandez, S.; Elefteriadou, L. Differences in Freeway Capacity by Day of the Week, Time of Day, and Segment Type. // Journal of Transportation Engineering-ASCE, 135, 7(2009), pp. 416–426. https://doi.org/10.1061/(ASCE)0733-947X(2009)135:7(416)

[20] Li, J.; Zhang, H. M. Fundamental Diagram of Traffic Flow - New Identification Scheme and Further Evidence from Empirical Data. // Transportation Research Record, 2260(2011), pp. 50–59. https://doi.org/10.3141/2260-06

[21] Kamarianakis, Y.; Gao, H. O.; Prastacos, P. Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions. // Transportation Research Part C-Emerging Technologies, 18, 5(2010), pp. 821–840. https://doi.org/10.1016/j.trc.2009.11.001

[22] Vlahogianni, E. I.; Karlaftis, M. G.; Golias, J. C. Statistical methods for detecting nonlinearity and non-stationarity in univariate short-term time-series of traffic volume. // Transportation Research Part C-Emerging Technologies, 14, 5(2006), pp. 351–367. https://doi.org/10.1016/j.trc.2006.09.002

[23] Multivariate forecasting methods. // Econometric Forecasting / Robert M. Kunst. Vienna: Institute for Advanced Studies, 2012, pp. 31–40.

[24] De Maesschalck, R.; Jouan-Rimbaud, D.; Massart, D. L. The Mahalanobis distance. // Chemometrics and Intelligent Laboratory Systems, 50, 1(2000), pp. 1–18. https://doi.org/10.1016/S0169-7439(99)00047-7

[25] Cho, S.; Hong, H.; Ha, B.-C. A hybrid approach based on the combination of variable selection using decision trees and case-based reasoning using the Mahalanobis distance: For bankruptcy prediction. // Expert Systems with Applications, 37, 4(2010), pp. 3482–3488. https://doi.org/10.1016/j.eswa.2009.10.040

[26] Krishnaswamy, J.; Bawa, K. S.; Ganeshaiah, K. N.; Kiran, M. C. Quantifying and mapping biodiversity and ecosystem services: Utility of a multi-season NDVI based Mahalanobis distance surrogate. // Remote Sensing of Environment, 113, 4(2009), pp. 857–867. https://doi.org/10.1016/j.rse.2008.12.011

[27] Zhang, Y.; Huang, D.; Ji, M.; Xie, F. Image segmentation using PSO and PCM with Mahalanobis distance. // Expert Systems with Applications, 38, 7(2011), pp. 9036–9040. https://doi.org/10.1016/j.eswa.2011.01.041

[28] Cunderlik, J. M.; Burn, D. H. Switching the pooling similarity distances: Mahalanobis for Euclidean. // Water Resources Research, 42, 3(2006), p. 3409. https://doi.org/10.1029/2005WR004245

[29] Farber, O.; Kadmon, R. Assessment of alternative approaches for bioclimatic modeling with special emphasis on the Mahalanobis distance. // Ecological Modelling, 160, 1–2(2003), pp. 115–130. https://doi.org/10.1016/S0304-3800(02)00327-7

[30] Roess, R. P.; Prassas, E. S.; McShane, W. R. Traffic Engineering, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[31] Hypothesis Testing. // An Introduction to Mathematical Statistics and Its Applications / Richard J. Larsen; Morris L. Marx. 5th ed. Boston: Pearson, 2012.

[32] Euclidean distance. // The Cambridge dictionary of statistics / Brian Everitt. Cambridge: Cambridge University Press, 2002, p. 134.

[33] Qi, Y.; Smith, B. L. Identifying nearest neighbors in a large-scale incident data archive. // Transportation Research Record, 1879, 1(2004), pp. 89–98. https://doi.org/10.3141/1879-11

[34] Zhijun, H.; Chuangwen, X. A Kind of Algorithms for Euclidean Distance-Based Outlier Mining and its Application to Expressway Toll Fraud Detection. // International Asia Conference on Informatics in Control, Automation and Robotics / Los Alamitos, 2009, pp. 414–417.

[35] Maurer, C. R.; Qi, R. S.; Raghavan, V. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. // IEEE Transactions on Pattern Analysis and Machine Intelligence, 25, 2(2003), pp. 265–270. https://doi.org/10.1109/TPAMI.2003.1177156

[36] Laurikkala, J.; Juhola, M.; Kentala, E.; Lavrac, N.; Miksch, S.; Kavsek, B. Informal identification of outliers in medical data. // Fifth International Workshop on Intelligent Data Analysis in Medicine and Pharmacology / Berlin, 2000, pp. 20–24.

[37] Filzmoser, P.; Garrett, R. G.; Reimann, C. Multivariate outlier detection in exploration geochemistry. // Computers & Geosciences, 31, 5(2005), pp. 579–587. https://doi.org/10.1016/j.cageo.2004.11.013

[38] Rousseeuw, P. J.; Van Zomeren, B. C. Unmasking multivariate outliers and leverage points. // Journal of the American Statistical Association, 85, 411(1990), pp. 633–639. https://doi.org/10.1080/01621459.1990.10474920

[39] Rousseeuw, P. J.; Driessen, K. V. A fast algorithm for the minimum covariance determinant estimator. // Technometrics, 41, 3(1999), pp. 212–223. https://doi.org/10.1080/00401706.1999.10485670

[40] Krzywicki, A.; Wobcke, W. Exploiting Concept Clumping for Efficient Incremental E-Mail Categorization. // Advanced Data Mining and Applications Pt II, 6441(2010), pp. 244–258. https://doi.org/10.1007/978-3-642-17313-4_25

[41] Sun, C. C.; Chilukuri, V. Dynamic Incident Progression Curve for Classifying Secondary Traffic Crashes. // Journal of Transportation Engineering, 136, 12(2010), pp. 1153–1158. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000187

[42] R Core Team. The R Project for Statistical Computing. R. 1993. http://www.r-project.org/. (17.05.2015).

[43] Song, Y. E.; Stein, C. M.; Morris, N. J. strum: an R package for structural modeling of latent variables for general pedigrees. // Bmc Genetics, 16(2015), p. 35. https://doi.org/10.1186/s12863-015-0190-3

[44] Palarea-Albaladejo, J.; Antoni Martin-Fernandez, J. zCompositions - R Package for multivariate imputation of left-censored data under a compositional approach. // Chemometrics and Intelligent Laboratory Systems, 143(2015), pp. 85–96. https://doi.org/10.1016/j.chemolab.2015.02.019

[45] Ahlin, C.; Stupica, D.; Strle, F.; Lusa, L. medplot: A Web Application for Dynamic Summary and Analysis of Longitudinal Medical Data Based on R. // Plos One, 10, 4(2015), p. 121760. https://doi.org/10.1371/journal.pone.0121760

[46] Janos, S.; Martinović, G. Web based distant monitoring and control for greenhouse systems using the Sun SPOT modules. // 7th International Symposium on Intelligent Systems and Informatics 2009, pp. 165–169. https://doi.org/10.1109/SISY.2009.5291170

[47] Alder, J. R.; Hostetler, S. W. Web based visualization of large climate data sets. // Environmental Modelling & Software, 68(2015), pp. 175–180. https://doi.org/10.1016/j.envsoft.2015.02.016

[48] Wang, R.; Zhong, D.; Zhang, Y.; Yu, J.; Li, M. A Multidimensional Information Model for Managing Construction Information. // Journal of Industrial and Management Optimization, 11, 4(2015), pp. 1285–1300. https://doi.org/10.3934/jimo.2015.11.1285

[49] Winston Chang; Joe Cheng; J. J. Allaire; Yihui Xie; Jonathan McPherson. Shiny: Web Application Framework for R. Feb-2015. http://cran.r-project.org/web/packages/shiny/index.html. (17.05.2015)

Authors' addresses

Bin Sun, Ph.D. Candidate
Blekinge Institute of Technology
Karlskrona 37179, Sweden
bin.sun@bth.se

Wei Cheng, Ph.D. Prof.
Corresponding Author
Kunming University of Science and Technology
Kunming 650093, China
Blekinge Institute of Technology
Karlskrona 37179, Sweden
wei.cheng@bth.se

Guohua Bai, Ph.D. Prof.
Blekinge Institute of Technology
Karlskrona 37179, Sweden
guohua.bai@bth.se

Prashant Goswami, Ph.D. Assist. Prof.
Blekinge Institute of Technology
Karlskrona 37179, Sweden
prashant.goswami@bth.se