Anomaly Detection in Smart City Traffic Based on Time Series Analysis

Mohammad Bawaneh
Department of Networked Systems and Services
Budapest University of Technology and Economics
Budapest, Hungary
mbawaneh@hit.bme.hu

Vilmos Simon
Department of Networked Systems and Services
Budapest University of Technology and Economics
Budapest, Hungary
svilmos@hit.bme.hu

***Abstract*—Anomaly detection in city traffic is playing a key role in intelligent transportation systems. Anomalies can be caused by different factors, such as accidents, extreme weather conditions or rush hours. In this paper, we propose a method which can detect anomalies in city traffic by analyzing the historical dataset collected from smart city sensors. The proposed Occupancy based anomaly detection algorithm (OBADA) is analyzing occupancy data of the roads, by searching for subsequence of major changes in values in the occupancy's time series which reflects an inordinate behavior. This was done by transforming the time series with a derivative estimation model, into a symbolic representation sequence. To detect the anomalies in the symbolic sequence, the modified z-score method was used. We have also introduced an enhancement by proposing a majority voting technique (OBADA_MV). The OBADA algorithm was evaluated using a historical dataset generated by the Simulation of Urban Mobility (SUMO) framework. By studying four different congestion scenarios, the results have shown that our algorithm can identify anomalies with more than 95% accuracy. OBADA was evaluated by comparing with other methods as well. The results have shown that OBADA anomalies Detection Rate (DR) is 100% and False Alarm Rate (FAR) is 0% which outperformed other methods, but this requires a higher time for detection.**

***Index Terms*—Anomaly Detection, Intelligent Transportation Systems, Big Data, Data Analysis, Time Series**

I. Introduction

Intelligent Transportation System (ITS) provides solutions for transportation problems such as traffic management, safety on roads or traffic jams [1], [2]. Therefore ITS can improves our daily life by helping road users to move in a smart way which makes transport easier and more reliable. This is done by using several approaches and algorithms which lead to several ITS applications, such as Electronic Toll Collection (ETC), Emergency Vehicle Preemption (EVP) or Traffic Management Systems (TMS) [3]–[7].

Traffic prediction algorithms are playing an important role in many of ITS applications. Therefore much research has been done in recent years to study road traffic, to predict future traffic behavior [8], [9]. Traffic prediction applications rely on the analysis and careful mining of sensed data from smart city environments. The general steps for building such applications is first, collecting trace data for moving objects by sensors, secondly, to process and structure the collected data for mining to build knowledge about the mobility trends and processes, and finally, the applications can use this knowledge to help the urban planners and city operators to be able to adapt and react to future traffic conditions [10]. Many of sensor technologies have been studied and used. There are two categories of data collection technologies: intrusive and non-intrusive. The intrusive technologies are depending on sensors placed in the road, such as magnetic loops, piezoelectric sensors, and pneumatic road tubes. On the contrary, non-intrusive technologies are depending on remote observations, such as passive and active infrared, passive magnetic, ultrasonic and passive acoustic, and video image detection [11], [12].

Traffic behavior prediction can address several road traffic challenges, like predicting the periods when the roads will become congested or estimating the effect of accidents or extreme weather conditions. The behavior of the road traffic can be influenced by many different factors, such as weather, the date of the year, the day of the week, and the time period of the day. Moreover, it is influenced by collisions or any sudden event, which is hard or some times impossible to predict. From this point of view, by developing an algorithm to identify any inordinate behavior of the road traffic can help the prediction process. To do this, the gathered road traffic data should be to be processed, structured and analyzed, after which the patterns of the road traffic can be learned. From these patterns, the anomalous events can be identified, using machine learning techniques or statistical methods. After detecting the anomalies in the road traffic behavior, the sections on the different roads which affect each other can be learned, and the movement of the anomalies between these sections can be predicted. For example, if congestion happens on a specific point in the city on a specific time (which will indicate an inordinate event), we will know that this congestion will influence other points' traffic which are related to this point. In other words, the congestion will move to the other points after a specific time. In this case, we can take action to prevent this congestion moving, such as adapting traffic lights or sending non-congested directions suggestions to the vehicles. In this paper we introduce an anomaly detection technique, which can detect the inordinate behavior in road traffic, based on the analysis of the occupancy time series of the roads, which can be observed using induction loops detectors. This research paper is organized as follows: related works are discussed in section 2, section 3 introduces the proposed anomaly detection technique, while section 4 and 5 present the experimental results and the performance evaluation respectively. Conclusion is discussed in section 6.

II. Related Works

ITS algorithms for anomaly detection have been discussed more intensively in recent years, as these anomaly detection techniques can be used for detecting the inordinate behavior in road traffic. The spatio-temporal data of the road traffic needs to be analyzed to study the behavior of the traffic. These data streams can consist of trajectories, which are time series for a particular vehicle or pedestrian, containing the vehicle's or pedestrian's location per time. Another form of data is made from time series for a particular location, containing information about the traffic on that location within a time interval [13]. Vehicle trajectory data can be observed using GPS equipped devices. Diverse research has used trajectories data to study traffic behavior [14]–[18]. Most of this research has used GPS data of taxis, since taxis can move around the city 24 hours. Some of them have used the GPS data of buses, as buses have a fixed route, they cannot choose a route by themselves based on traffic behavior. On the contrary, taxis can change their routes if they face traffic jams. However, these methods have some drawbacks. It is hard to force users to use such applications and send their location information. Moreover, even if taxis' data is used, we can still suffer from lack of information, because taxis will not be in all of the roads all the time, so, some we will not have a coherent picture from every part of the city. Location's time series data can be obtained using stationary sensors or using video image detection. These time series contain information about the traffic on a particular location, such as traffic flow and average speed within the time interval. This Information can be studied to learn the traffic behavior of the location, and to detect the anomalies of the traffic. Hasanzadeh et al. [19] have used location's time series data in their experiment, which were obtained from the Dallas-Forth Worth area. The travel time index (TTI) was used as a definition of road congestion. TTI is equal to the current travel time of the road divided by the free flow travel time of the road. They have used 1.7 as a threshold, so if the value of TTI for a road is greater than 1.7, then the road is congested. In other words, they are comparing the travel time on the road with the free flow travel time on the same road. If the current travel time is much higher than the free flow travel time, they consider that the road is congested.

Anomalies can be also detected by comparing time series with a training time series database, which consists of either normal time series database, or normal and abnormal time series database [20]. Similarity-based methods can be used to detect anomalies by comparing the tested time series with the training time series to assign the anomaly score [21], [22]. However, this training database is not always available. In addition, the seasonality of road traffic makes it harder. For example, in holiday time the road traffic behavior is totally different than on workdays. Anomalies can be also an abnormal subsequence relative to the rest of the time series. Analyzing the time series could help to find such anomalies. However, time series usually suffer from many challenges such as high dimensionality and noise. Therefore, to make handling and analyzing time series easier, the transformation of data is recommended. Keogh and Pazzani [23] proposed a similarity measurement algorithm which was called Derivative Dynamic Time Warping (DDTW). A transformation model was proposed in their algorithm to find information about the shape of the time series. DDTW estimation model can be obtained by computing the first derivative estimate for each point. They defined it as the average of the slope of the line between the right neighbor and the left neighbor, and the slope of the line between the point and the left neighbor. Gullo et al. [24] proposed a time series representation model called Derivative time series Segment Approximation (DSA). In their first step of transformation, they transform the time series to the derivative estimation. For doing that, they modify the DDTW estimation model. They consider computing the slope of the line between the right neighbor and the left neighbor for each point. This modification simplifies the transformation and makes it more accurate. The derivative estimation contains the trend information about the time series. Zhang and Pi [25] proposed a new method for time series representation and similarity measurement, called Fragment Alignment Distance (FAD). FAD estimates the derivative of time series using DSA derivative estimation. Then, FAD converts the DSA estimation model into a symbolic sequence, by setting a threshold and comparing it with the derivative estimation value of each point. If the value is less than the threshold, the point has a small change compared to the previous point, and they will be presented with the same symbol. If the value is bigger, the point has a big change compared to the previous point, so a different symbol will be assigned for this point. Our work was motivated by DSA and FAD algorithms since DSA estimation model contains the trend information about the time series.

III. Occupancy Based Anomaly Detection Algorithm (OBADA)

Anomaly detection in road traffic is the process of finding any inordinate behavior in traffic. It indicates that there are congestions and jams in the traffic, which can be caused by various factors such as incidents, rush hours, or even weather. In our paper the points where the traffic is measured (by induction loop detectors) will be called control points. These detectors can capture different attributes of traffic such as speed or occupancy. The occupancy is the percentage of the time a detector is occupied by a vehicle. This an interesting feature for studying the anomalies. In normal traffic behavior periods, occupancy observations values will be very close to each other since the vehicles are just passing the detectors and do not stop. However, in traffic jam cases, the detectors will be occupied much longer time. Moreover, the observations' values will keep changing along the traffic jams period. From this point of view, we proposed to search for the sequences of major changes in values in the occupancy time series, which reflects the inordinate behavior of traffic.

Steps of OBADA:

A. Derivative estimation

The first step of our OBADA algorithm is to obtain the derivative estimation model. DSA [24] is used to transform the occupancy time series into a DSA estimation model. As we mentioned before, the DSA estimation model contains information about the trend, which allows us to follow the trend of the time series. A given time series T of a length of h observations can be defined as:

$$T = (X_1, X_2, ..., X_h)$$

The DSA estimation model for a time series T can be defined as:

$$\hat{T} = (\hat{X}_1, \hat{X}_2, ..., \hat{X}_h)$$

Where the derivative estimation for each point can be computed as:

$$\hat{X}_i = \begin{cases} X_{i+1} - X_i & i = 1 \\ \frac{1}{2}(X_{i+1} - X_{i-1}) & i \in [2, ..., n] \\ X_i - X_{i-1} & i = n \end{cases} \tag{1}$$

B. Segmentation

The second step is to segment the DSA estimation model and to transform it into the FAD symbolic representation sequence [25] based on the values of the derivative estimation, by setting a threshold and comparing each value with this threshold. If the value is lower than the threshold, it will mean that this point contains only a minor change to the previous one, so they will be presented with the same symbol. If it was greater than the threshold, it means that a big change happened, and different symbols will be assigned for the points. Formally:

$$R_i = \begin{cases} \lambda & \hat{X}_i > \lambda \cdot \epsilon \\ ... & ... \\ 2 & 2 \cdot \epsilon < \hat{X}_i \leq 3 \cdot \epsilon \\ 1 & \epsilon < \hat{X}_i \leq 2 \cdot \epsilon \\ 0 & \mid \hat{X}_i \mid \leq \epsilon \\ -1 & -2 \cdot \epsilon < \hat{X}_i \leq -\epsilon \\ -2 & -3 \cdot \epsilon < \hat{X}_i \leq -2\epsilon \\ ... & ... \\ -\lambda & \hat{X}_i < -\lambda \cdot \epsilon \end{cases} \tag{2}$$

where ϵ is the threshold, and λ gives how many symbols will be used to represent the time series.

C. Outlier Detection

The symbolic representation sequence contains information about tendency changing. The normal behavior will be assigned with a number of symbols which depend on the threshold values. However these symbols will have a high frequency compared with inordinate behavior. On the contrary, inordinate behavior will be assigned with more symbols, and the frequency will be low for each symbol. A statistic method should be applied to the symbolic representation, which can detect outliers or extremeness in data, to identify the symbols with low frequencies. These symbols represent the congestion period or other inordinate behavior. The modified z-score method [26] is used in our algorithm to find the anomaly symbols. It measures how much a particular point differs from the rest of dataset. The modified z-score is defined as:

$$M_i = \frac{0.6745(R_i - \tilde{R})}{MAD} \tag{3}$$

where $\tilde{R}$ is the sample median and MAD is the median absolute deviation and is defined as:

$$MAD = median\{\mid R_i - \tilde{R} \mid\} \tag{4}$$

If the modified z-score is greater than a threshold the point is labeled outlier (the recommended threshold by authors of [26] is 3.5).

D. Majority Voting

The OBADA algorithm may wrongly identify some points as outliers, because of noise in the original time series, which will cause false alarms. Moreover, it can miss some outliers as well. To make the algorithm robust to noise, an enhancement technique is proposed by us, the OBADA majority voting (OBADA_MV) method. OBADA_MV uses the majority voting for each point after applying OBADA to ensure that the point was classified correctly. The OBADA_MV method is classifying the points based on neighbors points classification's results. The voting results of two neighbors' windows for each point will decide if it is an outlier or not. The most frequent class within the window is the result of the voting for the window. The point is labeled as an outlier if the neighbors in the right window or neighbors in the left window voted to be an outlier. The size of the window depends on the problem itself, so different sizes should be applied to historical data, and then the one which showed the best accuracy should be picked. In our experiment, we found that a window size of 5 neighbors will provide the highest accuracy. Figure 1 shows an illustrative example of OBADA_MV. If congestion happened, OBADA will identify the outliers points. These points should be successive. Therefore, if OBADA_MV was applied on a wrongly classified point within the congestion subsequence, the point will be classified as an outlier because of its neighbors' votes. On the other hand, if a false alarm happened the neighbors will be normal points, so the false alarm point will be labeled as a normal point.

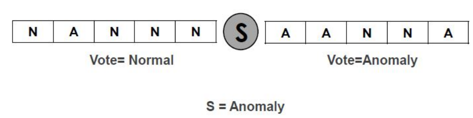

Fig. 1. Illustrative example of how OBADA_MV works. Where N means: normal point, A: anomaly point, S the tested point.

IV. EXPERIMENTAL RESULTS

In this section, the experimental results are introduced. We show how the proposed OBADA algorithm is able to detect the inordinate behavior of the road traffics based on the occupancy data.

A. Dataset Description

The Simulation of Urban Mobility (SUMO) framework [27] was used to simulate the traffic scenarios used for research, generating the needed dataset. The simulations were applied on a road network in Budapest, Hungary. A randomized traffic was generated on the network. The occupancy dataset was collected using induction loop detectors. The aggregation period that used for this simulation was 30 seconds. So, every sample in the dataset represents 30 seconds of the simulation, overall 9 hours of traffic was simulated. The congestion is set to happen after 3 hours, and continue for 30 minutes. Four scenarios were applied and studied, to verify the ability of the proposed method for different congestion scenarios. The scenarios are the following: accident, severe accident, traffic jams caused by road maintenance, and traffic jams caused by rush hour. Figure 2 shows a simulation snapshot of the congestion caused by accident (the red rectangle shows the position of the accident).

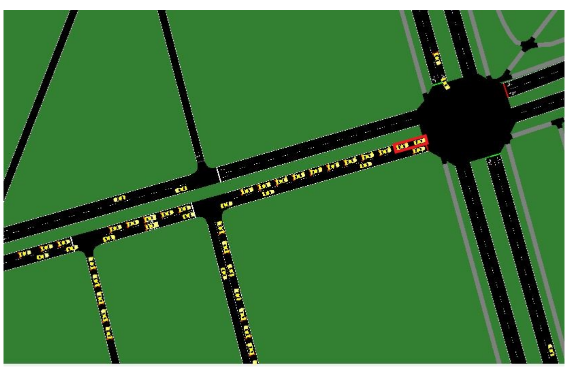

Fig. 2. Simulation snapshot for road congestion caused by accident (the red rectangle marks the site of the accident).

B. Dataset preprocessing

- Data Cleaning: Some values are missing because there was no traffic on the road within some data sample intervals. The forward fill approach was used to supplement these missing values, which is the process of supplementing the missing value with the previous value. Figure 3 shows the occupancy time series for the accident scenario. The shaded area represents the congestion period.
- Data Smoothing: Usually the data is affected by some noise, such as some abnormal behavior for seconds. For instance, a driver may suddenly press the brake. This does not reflect an abnormal behavior of road traffics. Therefore, the data was smoothed using an exponential moving average with α =0.75, where the exponential moving average can be computed recursively as:

$$S_t = \alpha \cdot X_t + (1 - \alpha) \cdot S_{t-1}$$

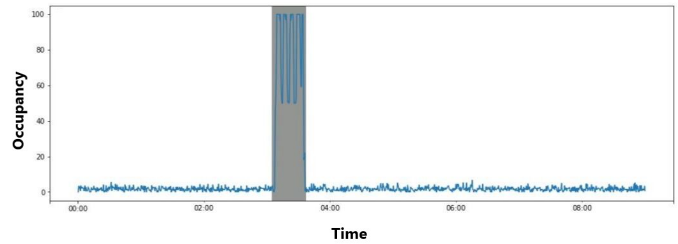

Fig. 3. The occupancy time series for 9 hours of traffic.

The smoothed occupancy time series is shown in Figure 4.

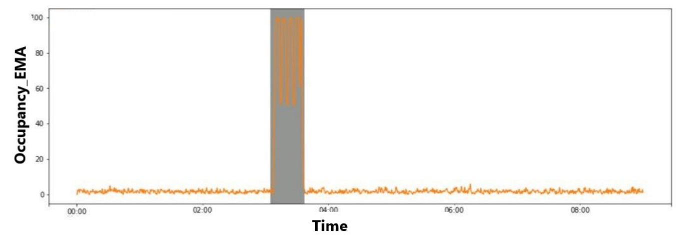

Fig. 4. The smoothed occupancy time series

C. Data Representation

As mentioned in section 3, the proposed algorithm is to search for the patterns of major changes in values in the occupancy's time series.

This can be done by transforming the data into the DSA estimation model, which has information about the shape of the time series, then transform the data into a symbolic representation sequence using the segmentation process from the FAD representation. The resulted DSA estimation model, as well as FAD symbolic representation sequence of the occupancy time series, are shown in Figure 5 and Figure 6 respectively.

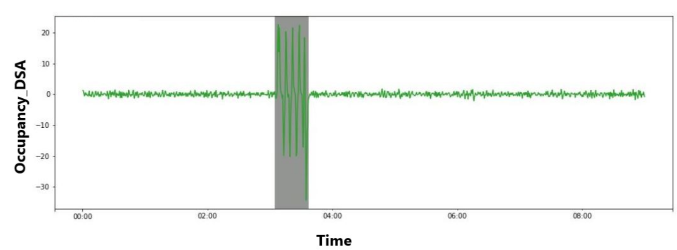

Fig. 5. The DSA estimation model of the occupancy time series.

D. Outlier Detection

After representing the occupancy time series by the FAD symbolic representation sequence, the modified z-score method is applied to detect the least frequently symbols, then OBADA_MV technique is used to enhance the result. Figure 7 shows the result of the detection. 1 is an outlier observation, 0 is an ordinary observation.

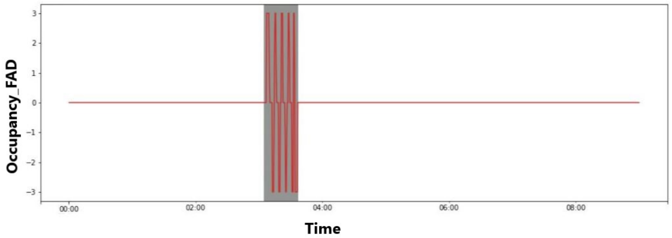

Fig. 6. The FAD symbolic representation of the occupancy time series.

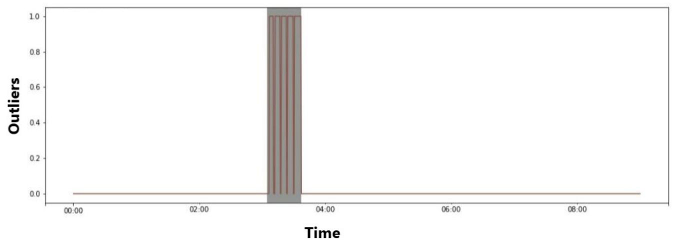

Fig. 7. The detected outliers by the OBADA algorithm.

V. Performance Evaluation

In order to evaluate the proposed algorithm, 2 different aspects of the evaluation were Taken into consideration. The first aspect is about evaluating the accuracy of OBADA in terms of classification correctness of data samples as outlier samples or normal samples. The second aspect is evaluating OBADA by comparing it with the commonly used anomaly detection methods in traffic in terms of numbers of congestion detected, numbers of false alarms, and the time needed to detect the congestion. Furthermore, in order to increase the reliability of our experiment 10 different simulations were applied and tested for each congestion scenario (the overall 40 simulations). The details of both aspects, as well as their results, are shown in the following:

A. Classification Correctness

Anomaly detection is a classification process. The data sample either classified as an outlier or normal point. Every sample represents 30 seconds of the simulation. The accuracy of the classification can be defined as the percentage of correctly classified samples. Formally:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the true positive, TN is the true negative, FP is the false positive, FN is the false negative. The average accuracy of the 10 simulations was calculated for each scenario. Table I shows the accuracy results of applying OBADA.

From table I, we can say that OBADA is able to detect the anomalies in traffic with high accuracy.

B. Performance Comparison

In this subsection, we are comparing OBADA with the commonly used anomaly detection methods in roads traffic

TABLE I
The average accuracy of the 10 simulations results for the 4 scenarios

	Accuracy
Accident	97.8 %
Severe Accident	95.4 %
Traffic Jams because of road maintenance	98.3 %
Traffic Jams because of rush hours	96.6 %

field. There are 3 commonly used performance measurements in the literature which are: Detection Rate (DR), False Alarm Rate (FAR), Mean Time To Detection (MTTD).

- Detection Rate (DR): The detection rate is the percentage of the correctly detected anomalies, Formally:

$$DR = \frac{No.\ of\ anomalies\ detected}{No.\ of\ true\ anomalies} * 100\% \quad (5)$$

- False Alarm Rate (FAR): The false alarm rate is the percentage of wrong alarm signals of detection to the total number of alarm signals. Formally:

$$DR = \frac{No.\ of\ false\ alarms}{No.\ of\ all\ alarms} * 100\% \quad (6)$$

- Mean Time To Detection (MTTD): is the average of the anomaly detection delay. Where the delay is the time difference between the anomaly's alarm, and the real congestion time. Formally:

$$DR = \frac{\sum_{i=1}^{N}(t_a - t_r)}{N} \quad (7)$$

Where N is the number of the detected anomalies, t_a is the time when the anomaly detected (alarm time), and t_r is the real congestion started time.

Same as the comparison performed in these works [28]–[32], we compared OBADA's results of DR, FAR, and MTTD with the reported values of some anomaly detection methods in traffic. We should note here that these measured values were not computed on the same environment, so the comparison here is qualitative. We also considered the number of congestions or incidents tested in each method to enhance the comparison interpretation. Table II shows the performance comparison.

TABLE II
Performance comparison between OBADA and previous works in anomaly detection

Method	Input Parameter	No. of Anomalies	DR	FAR	MTTD (min)
Bayesian [33]	Occupancy	17	100%	0%	3.9
SND [34]	Occupancy	35	92%	1%	1.1
Fuzzy Logic Based (1) [31]	Speed & Flow	80	91%	0%	2.2
Fuzzy Logic Based (2) [35]	Flow	90	100%	3.3%	0.68
OBADA	Occupancy	40	100%	0%	3.3

From table II, we can see that OBADA Outperformed Bayesian in terms of alarm's speed. Moreover, it outperformed SND and fuzzy logic based (1) in terms of DR and FAR, while they were faster than OBADA. Finally, OBADA outperformed fuzzy logic based (2) in terms of FAR, but fuzzy logic based (2) outperformed OBADA in terms of time. Therefore, from table II we can state that OBADA has high accuracy, but this required higher time for detection.

VI. Conclusion

In this paper, we introduced an anomaly detection algorithm for road traffic. In our method, we searched for the patterns of major changes in values in the occupancy's time series which represents the congestion period. This was done by transforming the occupancy time series into the DSA estimation model, then converting it to a symbolic representation sequence. The congestion patterns was detected by applying the modified z-score method to detect the anomaly points in the time series by detecting the least frequent symbols in the symbolic sequence. OBADA_MV technique was proposed as well to enhance the results, which was robust to noise. The proposed method was evaluated using the dataset generated by SUMO, and the performance evaluation shows that the proposed method can provide high accuracy for anomaly detection.

References

[1] C. Gosman, T. Cornea, C. Dobre, F. Pop, and A. Castiglione, "Controlling and filtering users data in intelligent transportation system," *Future Generation Computer Systems*, vol. 78, pp. 807–816, 2018.

[2] L. Figueiredo, I. Jesus, J. T. Machado, J. R. Ferreira, and J. M. De Carvalho, "Towards the development of intelligent transportation systems," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 1206–1211, IEEE, 2001.

[3] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Transactions on Intelligent transportation systems*, vol. 7, no. 3, pp. 377–392, 2006.

[4] S. Chattoraj, S. Bhowmik, K. Vishwakarma, and P. Roy, "Design and implementation of low cost electronic toll collection system in india," in *Electrical, Computer and Communication Technologies (ICECCT), 2017 Second International Conference on*, pp. 1–4, IEEE, 2017.

[5] Y.-S. Huang, Y.-S. Weng, and M. Zhou, "Design of traffic safety control systems for emergency vehicle preemption using timed petri nets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2113–2120, 2015.

[6] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy, "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Communications Surveys & Tutorials*, vol. 17, 2015.

[7] K. N. Qureshi and A. H. Abdullah, "A survey on intelligent transportation systems," *Middle-East Journal of Scientific Research*, vol. 15, no. 5, pp. 629–642, 2013.

[8] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[9] A. Abadi, T. Rajabioun, P. A. Ioannou, *et al.*, "Traffic flow prediction for road transportation networks with limited traffic data.," *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[10] G. Pan, G. Qi, W. Zhang, S. Li, Z. Wu, and L. T. Yang, "Trace analysis and mining for smart cities: issues, methods, and applications," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 120–126, 2013.

[11] G. Leduc, "Road traffic data: Collection methods and applications," *Working Papers on Energy, Transport and Climate Change*, vol. 1, no. 55, 2008.

[12] C. R. Bennett, H. De Solminihac, and A. Chamorro, "Data collection technologies for road management," 2006.

[13] G. Souto and T. Liebig, "On event detection from spatial time series for urban traffic applications," in *Solving large scale learning tasks. Challenges and algorithms*, pp. 221–233, Springer, 2016.

[14] A. Kinoshita, A. Takasu, and J. Adachi, "Real-time traffic incident detection using a probabilistic topic model," *Information Systems*, vol. 54, pp. 169–188, 2015.

[15] W. Kuang, S. An, and H. Jiang, "Detecting traffic anomalies in urban areas using taxi gps data," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[16] L. X. Pang, S. Chawla, W. Liu, and Y. Zheng, "On mining anomalous patterns in road traffic streams," in *International conference on advanced data mining and applications*, pp. 237–251, Springer, 2011.

[17] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 344–353, ACM, 2013.

[18] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: Long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[19] A. Hasanzadeh, X. Liu, N. Duffield, K. R. Narayanan, and B. Chigoy, "A graph signal processing approach for real-time traffic prediction in transportation networks," *arXiv preprint arXiv:1711.06954*, 2017.

[20] M. Teng, "Anomaly detection on time series," in *2010 IEEE International Conference on Progress in Informatics and Computing*, vol. 1, pp. 603–608, IEEE, 2010.

[21] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.

[22] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, 1994.

[23] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 2001 SIAM international conference on data mining*, pp. 1–11, SIAM, 2001.

[24] F. Gullo, G. Ponti, A. Tagarelli, and S. Greco, "A time series representation model for accurate and fast similarity detection," *Pattern Recognition*, vol. 42, no. 11, pp. 2998–3014, 2009.

[25] M. Zhang and D. Pi, "A new time series representation model and corresponding similarity measure for fast and accurate similarity detection," *IEEE Access*, vol. 5, pp. 24503–24519, 2017.

[26] B. Iglewicz and D. C. Hoaglin, *How to detect and handle outliers*, vol. 16. Asq Press, 1993.

[27] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.

[28] P. Xu, "122. a novel automatic incidents detection method based on support vector machine," *Boletín Técnico, ISSN: 0376-723X*, vol. 55, no. 17, 2017.

[29] A. B. Nikolaev, Y. S. Sapego, A. M. Ivakhnenko, T. E. Melnikova, and V. Y. Stroganov, "Analysis of the incident detection technologies and algorithms in intelligent transport systems," *International Journal of Applied Engineering Research*, vol. 12, no. 15, pp. 4765–4774, 2017.

[30] B. Khorashadi, F. Liu, D. Ghosal, M. Zhang, and C.-N. Chuah, "Distributed automated incident detection with vgrid," *IEEE Wireless Communications*, vol. 18, no. 1, pp. 64–73, 2011.

[31] X. Binglei, H. Zheng, and M. Hongwei, "Fuzzy-logic-based traffic incident detection algorithm for freeway," in *2008 International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1254–1259, IEEE, 2008.

[32] Y.-K. Ki, J.-W. Kim, and D.-K. Baik, "A traffic accident detection model using metadata registry," in *Fourth International Conference on Software Engineering Research, Management and Applications (SERA'06)*, pp. 255–259, IEEE, 2006.

[33] M. Levin and G. M. Krause, "Incident detection: A bayesian approach," *Transportation Research Record*, vol. 682, pp. 52–58, 1978.

[34] C. L. Dudek, C. J. Messer, and N. B. Nuckles, "Incident detection on urban freeways," *Transportation Research Record*, vol. 495, pp. 12–24, 1974.

[35] A. B. Nikolaev, Y. S. Sapego, A. N. Jakubovich, L. I. Berner, and A. M. Ivakhnenko, "Simulation of automatic incidents detection algorithm on the transport network.," *International Journal of Environmental and Science Education*, vol. 11, no. 16, pp. 9060–9078, 2016.