Real-time traffic incident detection based on a hybrid deep learning model

Article *in* Transportmetrica A: Transport Science · August 2020

DOI: 10.1080/23249935.2020.1813214

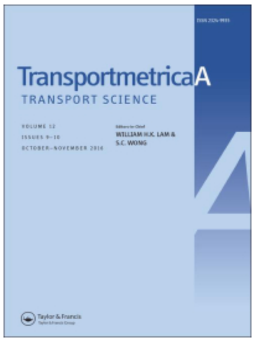
5 authors, including:

Linchao Li
Shenzhen University

Yi Lin
Sichuan University

Bowen Du
Beihang University (BUAA)

Fan Yang
University of Wisconsin–Madison



Transportmetrica A: Transport Science

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ttra21

Real-time traffic incident detection based on a hybrid deep learning model

Linchao Li, Yi Lin, Bowen Du, Fan Yang & Bin Ran

TRANSPORTMETRICA A: TRANSPORT SCIENCE
https://doi.org/10.1080/23249935.2020.1813214

Real-time traffic incident detection based on a hybrid deep learning model

Linchao Li[a], Yi Lin[b], Bowen Du[c], Fan Yang[d] and Bin Ran[d]

[a]College of Civil and Transportation Engineering, Shenzhen University, Shenzhen, People's Republic of China; [b]College of Computer Science, Sichuan University, Chengdu, People's Republic of China; [c]State Key Laboratory of Software Development Environment, Beihang University, Beijing, People's Republic of China; [d]School of Transportation, Southeast University, Nanjing, People's Republic of China

ABSTRACT

Small sample sizes and imbalanced datasets have been two difficulties in previous traffic incident detection-related studies. Moreover, real-time characteristics of incident detection models must be improved to satisfy the needs of traffic management. In this study, a hybrid model is proposed to address the above problems. In the proposed model, a generative adversarial network (GAN) is used to expand the sample size and balance datasets, and a temporal and spatially stacked autoencoder (TSSAE) is used to extract temporal and spatial correlations of traffic flow and detect incidents. Using a real-world dataset, the model is evaluated from different aspects. The results show that the proposed model, considering both temporal and spatial variables, outperforms some benchmark models. The model can both increase the incident sample size and balance the dataset. Furthermore, the sample selection method improves the real-time capacity of the detection.

ARTICLE HISTORY
Received 26 April 2019
Accepted 9 August 2020

KEYWORDS
Generative adversarial networks; deep learning; autoencoder; small sample size; imbalanced data

1. Introduction

Traffic congestion is a major concern for many large cities worldwide. For example, commuters lose over 100 hours per year because of traffic congestion in Los Angeles; in the UK, traffic congestion cost motorists over 49.7 billion dollars in 2017 (Cookson 2018). Traffic incidents can cause traffic congestion, which reduces highway capacity, increases the probability of a second crash, and increases air pollution. Therefore, traffic incident detection is important so that traffic managers can respond and manage incidents and so that travelers can select the best route to reduce travel time.

The emergence of the Intelligent Transportation System (ITS) concept provides strong support for traffic incident detection. In particular, massive numbers of sensors in the system, such as loop detectors, can collect large amounts of traffic flow data. The detectors can measure changes in the traffic flow near an incident and transmit the data to the system (Li et al. 2018). For example, when an incident occurs, the traffic volume of downstream and upstream detectors may change. In other words, collecting this large volume of traffic data

CONTACT Yi Lin yilin@scu.edu.cn

is fundamental to a successful incident detection model, and mining the patterns of traffic flow data effectively is highly important.

However, the datasets used to train such models still face two large challenges. The first challenge is the collection of sufficient incident samples, which are difficult to extract from the database – especially for some newly built highways – because the total number of incident samples is low. A small sample size may negatively affect the training of the incident detection model. The second is imbalanced samples. When we extract samples, far more non-incident samples can be obtained than incident samples, which causes some model training difficulties for machine learning-based incident detection models.

Advanced machine learning algorithms provide methodological support for traffic incident detection, which can be defined as a binary classification problem. In traffic incident classification, an incident sample can be defined as a 1, while a non-incident sample can be defined as a 0. The goal is to train an algorithm that can classify newly acquired samples. Compared with traditional statistical methods, machine learning methods, especially deep learning methods, have some advantages because they are able to extract traffic information effectively and efficiently from raw mixed data.

Based on deep learning methods, in this study, a hybrid model coupling a generative adversarial network (GAN) and a temporal and spatially stacked autoencoder (TSSAE) is developed to solve the above problems. The main contributions of this study are as follows.

- The GAN is applied to solve the small sample size and imbalance problems of traffic incident datasets. This approach can increase both the number of incident samples and their diversity, which improves the performance of the incident detection model.
- Temporal and spatial variable selection rules are proposed that are useful for capturing the temporal and spatial patterns of traffic flow. Using these rules, the incident detection model can extract the important features that different incidents and non-incidents.
- A temporal and spatial incident detection model is developed that can mine deep features in the traffic flow data. Moreover, the samples selected to train the model improve its real-time characteristics.
- The proposed hybrid model is evaluated from different aspects using a real-world dataset. The results indicate that our new model both increases the accuracy and improves the real-time characteristics of incident detection.

The remainder of this paper is organized as follows. In Section 2, some previous studies regarding incident detection and remaining problems are presented. In Section 3, we introduce our proposed model. The data used in this study are described in Section 4, and Section 5 provides an analysis of the results. Finally, Section 6 concludes the study and suggests avenues for future work.

2. Literature review

Traffic incident detection model is a popular topic in previous studies. In general, the applied models can be divided into two categories: statistics-based models and machine learning-based models (Ghosh and Smith 2014). Each model type is briefly discussed below.

- Statistics-based models: These types of models test differences in traffic flows based on statistical techniques, where a significant difference indicates a possible incident. The popular California and McMaster algorithms are representative of this type of model and have been widely applied (Hall, Shi, and Atala 1993; Samant and Adeli 2000). However, these simple models cannot provide sufficient accuracy to meet the requirements of an Intelligent Transportation System (Samant and Adeli 2000). To capture the temporal and spatial correlations among traffic flows, some studies implemented advanced statistical techniques. For example, an autoregressive integrated moving average model was built to detect traffic incidents on the Lodge Highway in Detroit (Ahmed and Cook 1979); the proposed detection logic performed smoothing using a moving average filter and obtained better results (Chassiakos and Stephanedes 1993). Later, a multiple model particle smoother was introduced to convert the incident detection problem into a traffic state prediction problem and solve it effectively (Wang, Fan, and Work 2016; Wang, Work, and Sowers 2016). Although statistics-based models have been widely applied, they have some shortcomings. First, the algorithm assumptions may not be consistent with the actual traffic flow data. Second, these models are highly dependent on user experience. When implementing a statistics-based model, the thresholds are often set manually by the users. Moreover, these models sometimes cannot simultaneously consider the temporal and spatial correlations among traffic flow data (Mak and Fan 2007; Li et al. 2019).
- Machine learning-based models: To make the incident detection model more flexible and robust, various machine learning models have been applied. Because the models are driven by the data, such models can easily be implemented without specialized knowledge. The traffic incident detection problem is first converted into a binary classification task in which an incident is defined as a 1 and a non-incident is defined as a 0. Then, a machine learning model such as a support vector machine (SVM) (Yuan and Long Cheu 2003; Chen, Wang, and Van Zuylen 2009; Xiao and Liu 2012), classification tree (CT) (Chen and Wang 2009), random forest (RF) (Liu, Lu, and Chen 2013), or artificial neural network (ANN) (Samant and Adeli 2001; Adeli and Samant 2000) can be used to solve the task. Li et al. compared some famous machine learning models and found that ensemble approaches improve the performance. Adding a bagging strategy to an SVM increased the accuracy (Li et al. 2016b). Some advanced ANN models have been widely applied in previous traffic incident detection studies and have obtained better results. As deep learning theory has developed, some models have already been applied in various transportation areas. Ma et al. used a deep neural network to recognize traffic congestion on a highway network using both temporal and spatial traffic flow characteristics (Ma et al. 2015). Zhu et al. developed an incident detection model at the network level based on a convolutional neural network (CNN) (Zhu et al. 2018). It has been proven that deep learning models outperform traditional machine learning models because they can fully mine the traffic information from the data. However, achieving a sufficient number of samples is difficult when applying a deep learning model. Consequently, simulated data have been widely used, but sometimes such data does not represent the true highway traffic flow (Lv et al. 2015; Ma et al. 2017; Zhu et al. 2018; Wu et al. 2018). Another method applied to solve the small sample size problem is to collect samples during each incident as incident samples to

increase the sample size. However, this approach could affect the real-time capacity of the model.

Thus, the previous studies exhibit some problem that still need to be solved, including the following:

- How to obtain a richer set of traffic incident samples to train and test the model;
- How to construct a balanced dataset in which the number of incident samples equals the number of non-incident samples;
- How to improve the real-time capability of a traffic incident detection model;
- How to effectively extract the spatial and temporal correlations from the traffic flow data to improve the performance of a traffic incident detection model.

To fill the above gaps, we apply deep learning theory in our study. First, we use GANs to solve the sample size problem. GANs are recent models in the deep learning area that were proposed to mimic a data distribution to create new data similar to the original data. In recent years, GANs have commonly been used to improve image processing capacity (Radford, Metz, and Chintala 2015), generate high-quality images (Mirza and Osindero 2014; Denton, Chintala, and Fergus 2015; Odena, Olah, and Shlens 2016), and address text-to-image tasks (Reed et al. 2016). In the transportation area, GANs have been applied to detect driver behaviors (Ghosh, Bhattacharya, and Chowdhury 2016), in autonomous driving (Kuefler et al. 2017) and for traffic state estimation (Liang et al. 2018). To the best of our knowledge, this study is the first attempt to solve the sample size problem for traffic incident detection.

3. Methodologies

This section first introduces the GAN applied to increase the sample size of incident cases; then, the TSSAE is implemented to detect traffic incidents. Finally, the novel hybrid model developed in this study is presented.

3.1. Generative adversarial network

The commonly applied GAN model contains two parts: a generator $G(\mathbf{z};\theta_g)$, which is used to generate new samples $G(\mathbf{z}) \in \mathbb{R}^d$ from a random prior $\mathbf{z} \in \mathbb{R}^r$ and a discriminator $D(\mathbf{x};\theta_d)$, which is used to recognize whether a newly generated sample is real or fake. The goal is to train a generative model G that can maximize the probability that the discriminative model D will misclassify generated samples as real samples. As demonstrated by Goodfellow et al. (2014), the GAN framework can be abstracted as a simple two-player minimax game that completes when Nash equilibrium is satisfied. Thus, the objective of a GAN is to minimize the following objective function:

$$\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_{data}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}}[\log(1 - D(G(\mathbf{z})))] \tag{1}$$

where p_{data} and $p_{\mathbf{z}}$ represent the distribution of the real sample and a random prior distribution (such as a Gaussian distribution), respectively. During the training process, the parameters of G and D are updated using the following two equations:

$$\theta_d \leftarrow \theta_d + \alpha \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} (\log D(\mathbf{x}_i) + \log(1 - D(G(\mathbf{z}_i)))) \tag{2}$$

$$\theta_g \leftarrow \theta_g - \alpha \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D(G(\mathbf{z}_i)) \tag{3}$$

where m represents the number of training samples and α is the step size. As demonstrated by Goodfellow et al., the parameters of the generator G are optimized by maximizing $\log(D(G(\mathbf{z})))$ to speed up GAN training. Thus, Equation (3) is rewritten as:

$$\theta_g \leftarrow \theta_g + \alpha \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(G(\mathbf{z}_i) \tag{4}$$

In this study, an alternative training method is applied that involves two steps. In the first step, the generator G is fixed, and the discriminator D is optimized to maximize its accuracy. In the second step, the discriminator D is fixed, and the generator G is optimized by minimizing the accuracy of the discriminator D. When $p_{data} = p_{\mathbf{z}}$, the training process is terminated.

The architecture of a GAN is shown in Figure 1; the two models can be any type of multilayer perceptron. In this study, two fully connected neural networks are applied as the generator and discriminator. The training procedures for a GAN are shown in Figure 2.

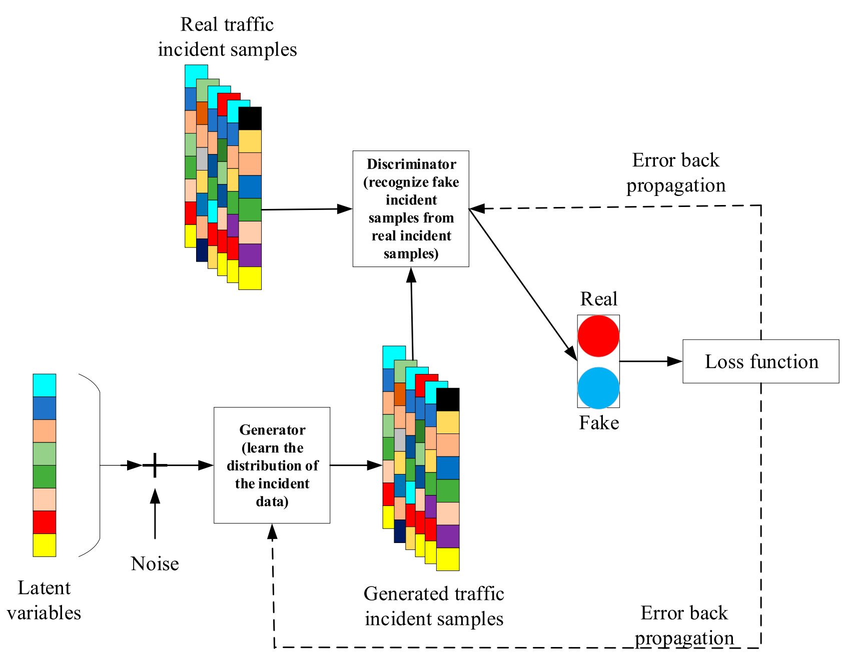


Figure 1. Architecture of the applied GANs.

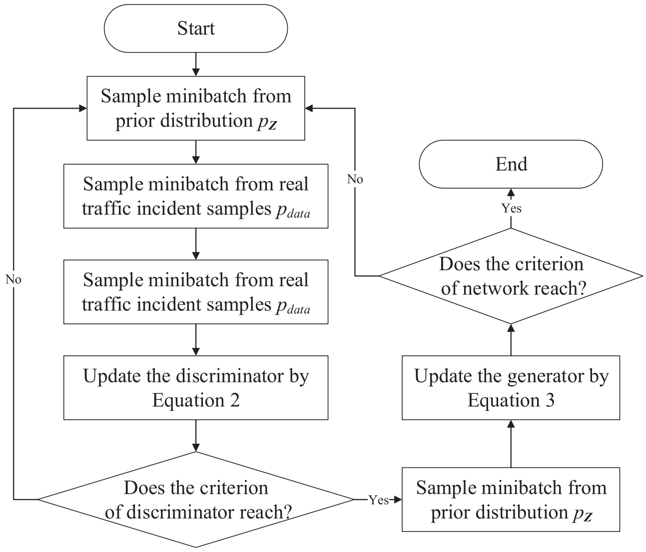


Figure 2. GAN training procedures.

3.2. Temporal and spatial stacked autoencoder (TSSAE)

3.2.1. Sparse autoencoder

An autoencoder (AE) is developed to extract latent features from raw data and then to reconstruct the raw data based on the latent features. The data reconstruction makes the AE extract deep hidden features to adequately represent the raw data. To recognize the occurrence of a traffic incident, it is necessary to mine the hidden spatial and temporal features of traffic flows. As shown in Figure 3, an AE comprises an encoder $En(\mathbf{x}, \theta_{en})$ to extract features that represent the input data $\mathbf{x}$ and a decoder $De(En(\mathbf{x}), \theta_{de})$ that reconstructs the represented features to recreate the original input data $\mathbf{x}$:

$$\mathbf{y} = sig(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \tag{5}$$

$$\mathbf{x}' = sig(\mathbf{W}_2\mathbf{y} + \mathbf{b}_2) \tag{6}$$

where $\mathbf{x}'$ is the reconstruction of $\mathbf{x}$; $\theta_{en} = (\mathbf{W}_1, \mathbf{b}_1)$ and $\theta_{de} = (\mathbf{W}_2, \mathbf{b}_2)$ are parameters of the encoder and decoder, respectively, in which $\mathbf{W}_1$, $\mathbf{W}_2$ are weight matrices and $\mathbf{b}_1$, $\mathbf{b}_2$ are biases. $sig(\cdot)$ represents the logistic sigmoid function $(1 + \exp(-x))^{-1}$ which is widely applied in traffic flow prediction.

The objective of an AE is to minimize the error between the input data and the reconstructed input data:

$$\min \frac{1}{m} \sum_{i=0}^{m} \|\mathbf{x} - \mathbf{x}'\|_2^2 \tag{7}$$

where m is the number of training samples. In our study, the goal is to extract the deep hidden features of spatial and temporal variables. Therefore, a sparsity constraint is added to the objective function to control the nonlinear mapping; then, the objective is rewritten

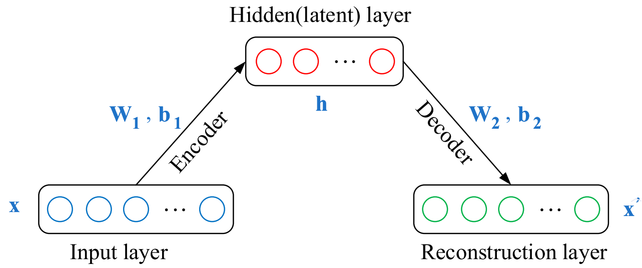


Figure 3. Architecture of the AE.

as follows:

$$\min \frac{1}{m}\left(\sum_{i=0}^{m} \|\mathbf{x}-\mathbf{x}'\|_2^2\right) + \gamma \sum_{j=1}^{N} KL(\rho || \hat{\rho}_j) \tag{8}$$

where σ represents the weight of the sparsity constraint; N is the number of variables; ρ is the sparsity parameter to control the feature set; $\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}(y_j)_i$ is the average activation of the jth hidden unit a_j over the m training samples; and KL is the Kullback–Leibler divergence, which is given by:

$$KL(\rho || \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_j} \tag{9}$$

3.2.2. *Temporal variable selection rules*

As demonstrated in Abdel-Aty et al. (2004), Hossain and Muromachi (2012), Xu, Wang, and Liu (2013) and Yu et al. (2016), finding the temporal correlations of traffic flow is essential when building a traffic incident detection model. Therefore, extracting the difference between normal traffic conditions and risky traffic conditions is critical. The detectors widely used in the current Intelligent Transportation System sense traffic flow data every 30 s, including traffic speed, traffic volume and traffic density. In this study, we adopt the three traffic flow parameters 5 min before an incident because many previous studies have shown that traffic flow conditions start to deviate 5 min before an incident (Qu et al. 2017; Xu et al. 2015). Moreover, the means and standard deviations of the traffic flow parameters during this period are also calculated and selected as temporal variables. Finally, from each detector, $3 \times 10 + 3 \times 2 = 36$ variables can be selected as temporal variables.

3.2.3. *Spatial variable selection rules*

Knowing the spatial correlations of traffic flow is also important to the incident detection model. Based on shock wave theory, it can be inferred that some time must elapse for the influence of an incident to spread. Thus, traffic flow parameters obtained from adjacent upstream and downstream detectors should also be considered because traffic flow near an incident is more sensitive than is more distant traffic flow. The traffic flow parameters of upstream or downstream detectors change earlier; therefore, considering these variables can help the model detect incidents with less delay. For this study, we also selected combinations of the traffic flow parameters obtained from the upstream and downstream detectors as spatial variables. The combinations are shown to contribute to the detection model accuracy, such as the California algorithms that apply the difference between the occupancy at two adjacent detectors as one of the variables (Karim and Adeli 2002).

3.2.4. TSSAE

After the selection, a total of 81 variables (listed in Table 1) are considered in the traffic incident detection model. To deeply mine the correlations among the temporal and spatial variables, the hierarchical model TSSAE is built, as shown in Figure 4. In the bottom layers, the variables of different detectors are input to different sparse AEs. Then, the latent temporal features extracted by the different detectors are combined by an added joint layer that learns the spatial correlations. The proposed model uses 78 variables are used rather than the three combined spatial variables because the model can capture the spatial correlation in the joint layer. After the high-level spatial and temporal feature learning, an output layer is added consisting of a softmax classifier in this study, which is a supervised model whose function is:

$$f_{out} = \frac{1}{1 + \exp(-\mathbf{W}_3\mathbf{z})} \tag{10}$$

where $\mathbf{W}_3$ represents the weights and $\mathbf{z}$ represents the learned features.

The deep neural network can easily be trained by applying the backpropagation method and the gradient-based optimization algorithm; however, it has been shown that deep architectures trained in this way perform worse. Fortunately, Hinton et al. proposed a greedy layer-wise unsupervised learning technique that can successfully optimize deep neural networks (Hinton, Osindero, and Teh 2006; Bengio et al. 2007). First, the model is

Table 1. Variables selected using the proposed temporal and spatial rules (Li et al. 2020).

Variable	Name
Speed at the upstream detector just after the incident	s_up_0
Volume at the upstream detector just after the incident	v_up_0
Occupancy at the upstream detector just after the incident	o_up_0
Speed at the downstream detector just after the incident	s_dn_0
Volume at the downstream detector just after the incident	v_dn_0
Occupancy at the downstream detector just after the incident	o_dn_0
Speed difference between the upstream and downstream detectors just after the incident	s_up_dn
Volume difference between the upstream and downstream detectors just after the incident	v_up_dn
Difference in occupancy between the upstream and downstream detectors just after the incident	o_up_dn
Speed at the upstream detector t before the incident	s_up_t
Volume at the upstream detector t before the incident	v_up_t
Occupancy at the upstream detector t before the incident	o_up_t
Speed at the downstream detector t before the incident	s_dn_t
Volume at the downstream detector t before the incident	v_dn_t
Occupancy at the downstream detector t before the incident	o_dn_t
Mean upstream traffic speed during the 5 min before the incident	m_s_up
Mean downstream traffic speed during the 5 min before the incident	m_s_dn
Mean upstream traffic volume during the 5 min before the incident	m_v_up
Mean downstream traffic volume during the 5 min before the incident	m_v_dn
Mean upstream occupancy during the 5 min before the incident	m_o_up
Mean downstream occupancy during the 5 min before the incident	m_o_dn
Standard deviation of the upstream traffic speed during the 5 min before the incident	s_s_up
Standard deviation of the downstream traffic speed during the 5 min before the incident	s_s_dn
Standard deviation of the upstream traffic volume during the 5 min before the incident	s_v_up
Standard deviation of the downstream traffic volume during the 5 min before the incident	s_v_dn
Standard deviation of the upstream occupancy during the 5 min before the incident	s_o_up
Standard deviation of the downstream occupancy during the 5 min before the incident	s_o_dn

Note: In the table, t equals: 30 s, 60 s, 90 s, 120 s, 150 s, 180 s, 210 s, 240 s, 270 s, 300 s.

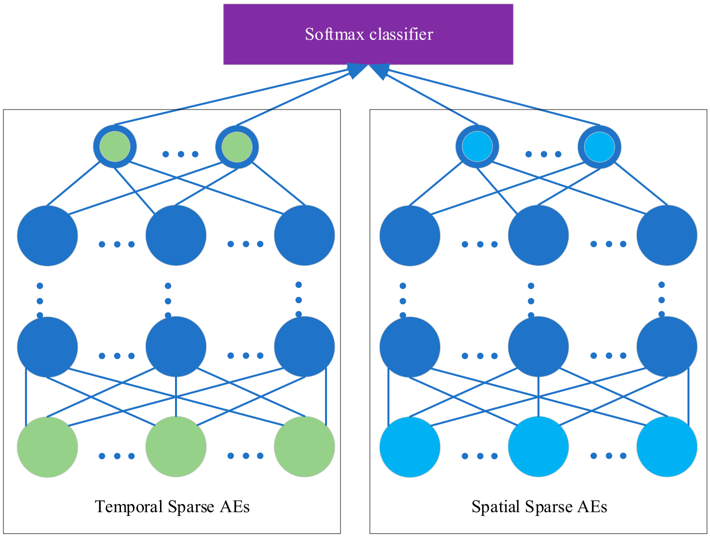


Figure 4. Architecture of the proposed TSSAE.

Table 2. Proposed TSSAE training procedures.

Training TSSAE
Given the training samples and the number of hidden layers, hidden modes, Step (1) Pretrain (1) Set the parameters for the objective function, including the weight of the sparsity constraint σ and the sparsity parameter ρ. (2) Initialize the network parameters randomly. (3) Conduct greedy layer-wise network training. (a) Input the training samples to the first hidden layer, whose output forms the input of the next hidden layer. (b) Optimize the parameters of the second layer by minimizing the objective function. (c) Repeat (a) and (b) until the last hidden layer is reached. Step (2) Fine-tuning (1) Initialize the weights of the output layer. (2) Fix the left temporal sparse AEs and apply backpropagation method and gradient-based optimization to tune the network parameters. (3) Fix the right spatial sparse AEs and apply backpropagation method and gradient-based optimization to tune the network parameters.

pretrained in a bottom-up direction. Then, the parameters of the model are tuned using backpropagation in a top-down direction. The procedural details are listed in Table 2.

3.3. *The developed hybrid model*

The preceding sections introduced the two key parts of our hybrid model, the GAN and the TSSAE. The architecture of the proposed hybrid model is shown in Figure 5. The GAN is first applied to generate new incident samples using the selected spatial and temporal variables. Then, the new datasets containing newly generated incident samples are used as the input to the TSSAE. The last step is to evaluate the performance of the proposed model. In this study, we apply four criteria: detection rate (DR), false alarm rate (FAR), classification

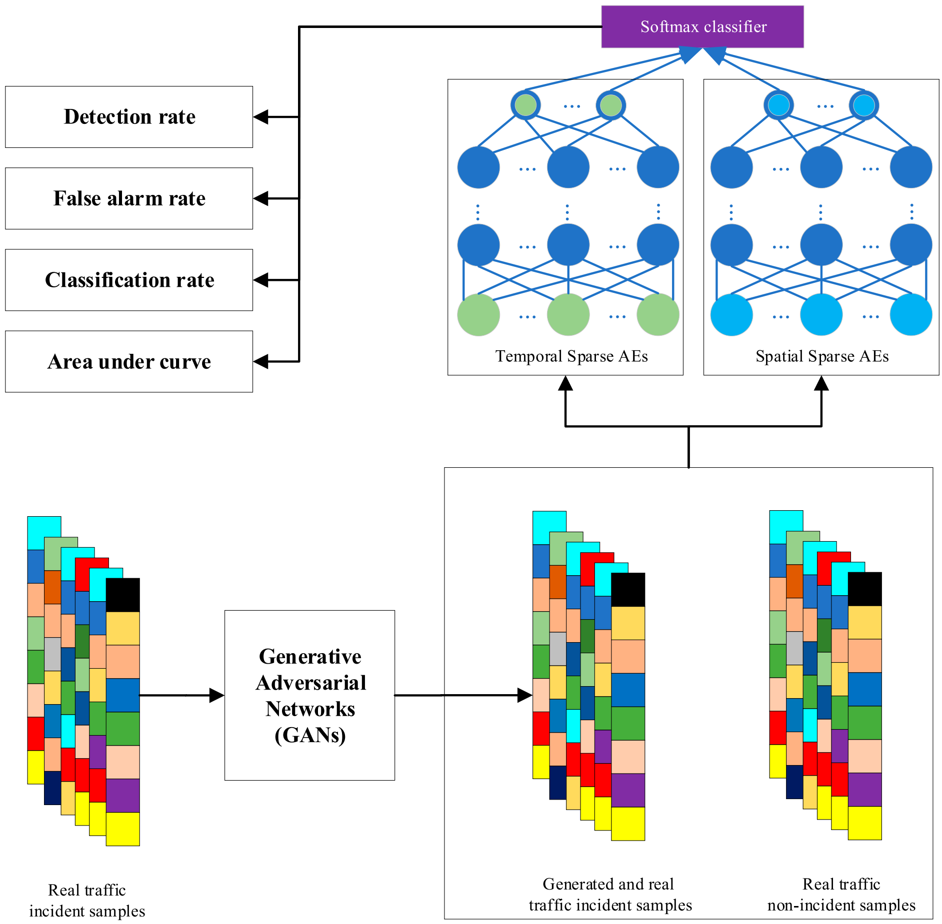


Figure 5. Architecture of the proposed hybrid model.

rate (CR) and the area under the curve (AUC).

$$DR = \frac{\textit{Number of incidents correctly detected}}{\textit{Number of actual incidents}} \tag{11}$$

$$FAR = \frac{\textit{Number of incidents falsely detected}}{\textit{Number of the samples correctly detected}} \tag{12}$$

$$CR = \frac{\textit{Number of samples correctly detected}}{\textit{Number of samples}} \tag{13}$$

DR indicates the proportion of incidents correctly detected. A higher DR represents a more accurate model. However, a model with higher DR may also be overly sensitive, that is, it falsely detects more incidents (Asakura et al. 2017). Therefore, another criterion, FAR, is introduced to evaluate model accuracy. AUC is the area under the receiver operating characteristic (ROC) curve, which represents the classification ability of the model as the discrimination threshold varies. Moreover, the computation time of the model is calculated to evaluate its efficiency.

4. Data description

The first dataset used in this study was collected from a well-known, open traffic flow data website called Caltrans Performance Measurement (PeMS), where we extracted the incidents reported on I-80 in the US state of California from the incident database. Because this study aims to model the relation between traffic flow and traffic incidents, incidents that occurred in work zone areas were deleted. Second, traffic flow data measured by loop detectors were obtained using the clearinghouse tool. Loop detectors are installed on the highway approximately every 0.5 mile. We used the traffic flow parameters (including traffic speed, volume and density) from more than 50 detectors. In addition, we calculated some introduced combined variables. The above two datasets both include position variables that can be used to join them together. We conducted this task using geographic information system software. After combining the datasets, we found that some traffic flow data corresponding to incidents were missing. To ensure the data quality, the samples with missing values were deleted. Finally, we obtained 139 complete incident samples and adopted these data as the incident dataset.

Selecting non-incident samples corresponding to the incident samples is important but difficult because it is impossible to guarantee that all conditions, such as weather conditions, are the same. To eliminate the influences of other factors, we implemented the commonly applied case control method (Abdel-Aty et al. 2004). We collected non-incident samples under similar weather conditions at the same location during the same period as the incident samples. Using this approach, several matched non-incident samples were obtained for each incident sample. After selection, we obtained a total of 834 non-incident samples and defined these data as the non-incident dataset.

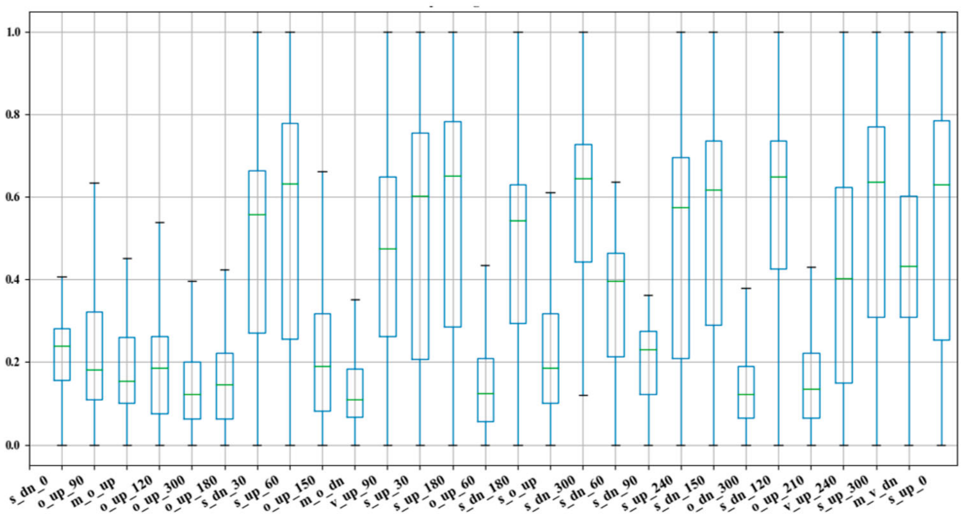
The incident samples in the incident dataset are those that occurred just after the incident. Although the traffic flow of an incident sample is different from the traffic flow when the incident happens, they are still quite similar. Furthermore, we selected some incident samples with durations greater than 120 s. The samples that represent 30 s, 60 s, 90 s, and 120,s after these incidents, are extracted as incident samples and are defined as the 30-s-incident dataset, 60-s-incident dataset, 90-s-incident dataset, and 120-s-incident dataset, respectively. We want to build a real-time traffic incident detection model that can detect an incident immediately after it happens; however, these samples can confuse the model because the traffic flow during this period and the normal period differ more widely. The real-time characteristics of our model are one of the main contributions of this study.

To set the parameters of the incident detection models, we conducted ten-fold cross-validation. In this method, the dataset is first divided into ten parts, each with an equal number of samples. Subsequently, nine parts are used to train the model, and the remaining part is used to validate the model. This cross-validation process is repeated 10 times, and each of the ten parts is used once as the validation set. Finally, the average error of the ten cross-validations is calculated as the true error.

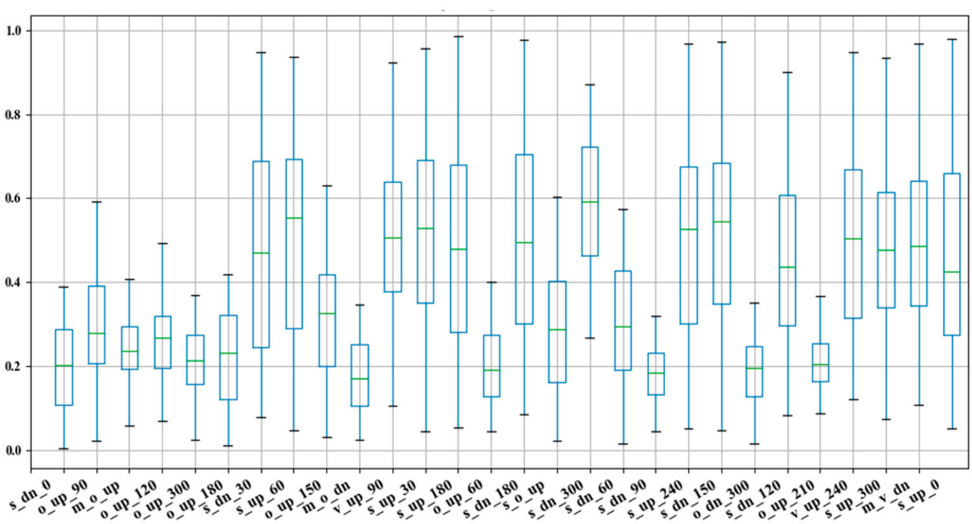
5. Results

5.1. A comparison of real and generated incident samples

Using the GAN, each raw sample is regenerated six times, creating a total of 834 incident samples. This dataset is defined as the generated-incident dataset. To evaluate the



(a) Descriptive statistics of the real data



(b) Descriptive statistics of the generated data

Figure 6. Comparison of the statistics for the real and generated data (Lin et al. 2020). (a) Descriptive statistics of the real data. (b) Descriptive statistics of the generated data.

performance of the GAN, the new incident samples are compared with the raw incident samples. The resulting descriptive statistics, including the minimum, first quartile, median, third quartile and maximum of the variables, are shown in Figure 6. Because each sample includes so many variables, we selected 28 important variables (ranked by a random forest model) to display in the figure, which shows that all five statistics of the newly generated incident sample and raw incident sample variables are similar – but not the same. These results indicate that the GAN can effectively generate incident samples. Moreover, the generated incident samples improve the sample diversity, which can contribute to improving the accuracy of the incident detection model.

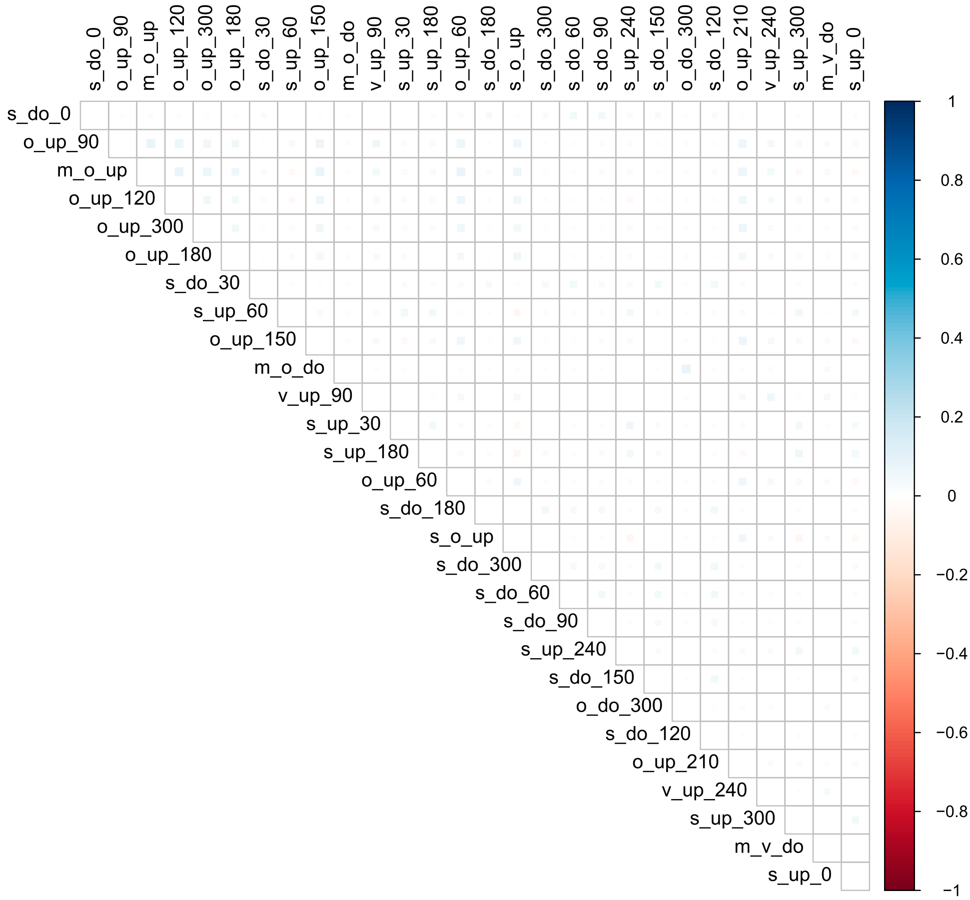


Figure 7. Differences in the correlations of variables between the real dataset and the generated dataset.

To further analyze the effectiveness of the GAN, we calculated the difference between the correlation matrix of the raw dataset and the correlation matrix of the generated dataset, as shown in Figure 7. The difference is close to 0, which means that the correlations between the variables in the raw dataset and the correlations between the variables in the generated dataset are similar. This result indicates that the GAN captures the correlations between variables in the raw samples and reflects those the correlations into the generated samples, again indicating the effectiveness of the GAN model.

5.2. GAN effectiveness

The GAN was used in the proposed model to solve imbalance and sample size problems when building a traffic incident detection model. Therefore, we conducted two experiments. The first experiment evaluates the effectiveness of the GAN in dealing with the imbalanced sample problem. In this experiment, the six different datasets shown in Table 3 are used to train the proposed model: five are imbalanced datasets, and one is a balanced dataset. The incident samples are from the defined incident dataset, and the non-incident

Table 3. Descriptions of the datasets used in the experiments.

Data set	Number of incident samples	Number of non-incident samples
I1	139	139
I2	139	278
I3	139	417
I4	139	556
I5	139	695
I6	139	834
B1	139	139
B2	278	278
B3	417	417
B4	556	556
B5	695	695
B6	834	834

samples are taken from the defined non-incident dataset. The second experiment is conducted to evaluate the effectiveness of the GAN in addressing the sample size problem. In this experiment, the six balanced datasets shown in Table 3 are used to train the proposed model. In contrast to experiment 1, this experiment uses the incident samples in the generated-incident dataset.

When the imbalanced datasets are used to train the model, the model performance tends to decrease as the ratio of non-incident samples and incident samples increases. From Figure 8, it can be seen that the CR and FAR scores of the models trained by datasets I1, I2, I3, I4, I5, and I6 tend to increase, while their DR and AUC results tend to decrease. This indicates that imbalanced samples negatively affect the accuracy of the proposed incident detection model. The results of models trained on the balanced datasets and the corresponding imbalanced datasets can be found by comparing Figures 8 and 9, which shows that a model trained on a balanced dataset performs better than do models trained on the imbalanced datasets. The average DR and AUC results of the six models using balanced datasets decreased by 1.89% and 10.34%, respectively. Moreover, the balanced datasets reduce the FAR by approximately 84.60% on average. This indicates that the generated samples effectively improve the performance of the proposed incident detection model.

A comparison of the models trained on the balanced datasets shows that as the number of training samples increases, the performance of the models improves. Compared to the dataset B1, the model trained on dataset B6, (a fivefold increase in the number of training samples) increases the DR, CR, and AUC by approximately 1.85%, 1.07%, and 8.24%, respectively, and decreases the FAR by approximately 23.18%. The results indicate that the GAN-generated additional samples can be used to train a more accurate incident detection model.

5.3. *The effectiveness of the TSSAE*

In the hybrid model, the proposed TSSAE is applied as the traffic incident detection model. To evaluate the performance of the TSSAE, several commonly used models, including BPNN, SVM and RF, are implemented as benchmark models. To ensure fairness, both spatial and temporal variables are considered in the benchmark models. Moreover, we compared the newly built model using normal stacked AEs while considering spatial variables (SSAE), temporal variables (TSAE), or both temporal and spatial variables (TSSAE).

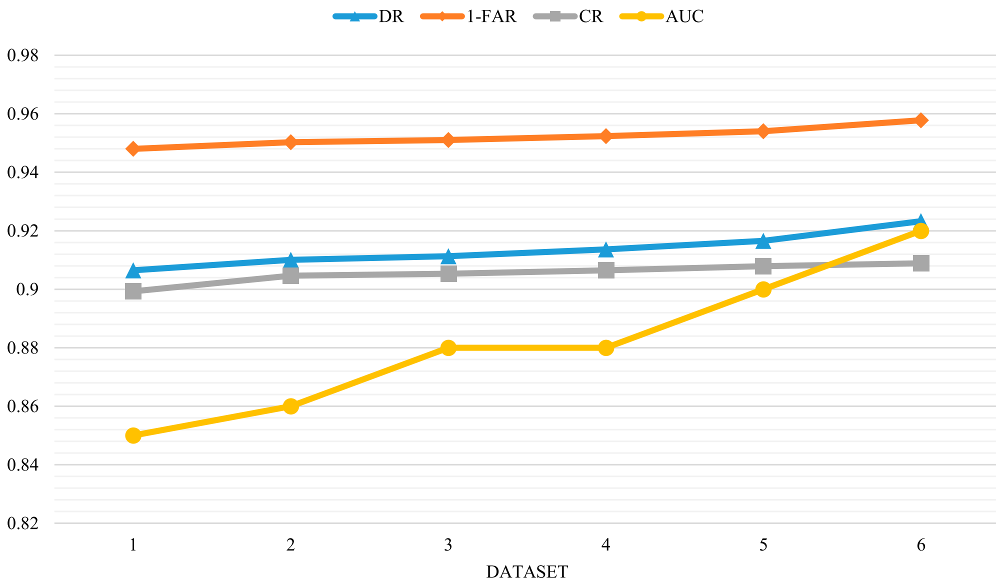


Figure 8. Description and results of experiment 1.

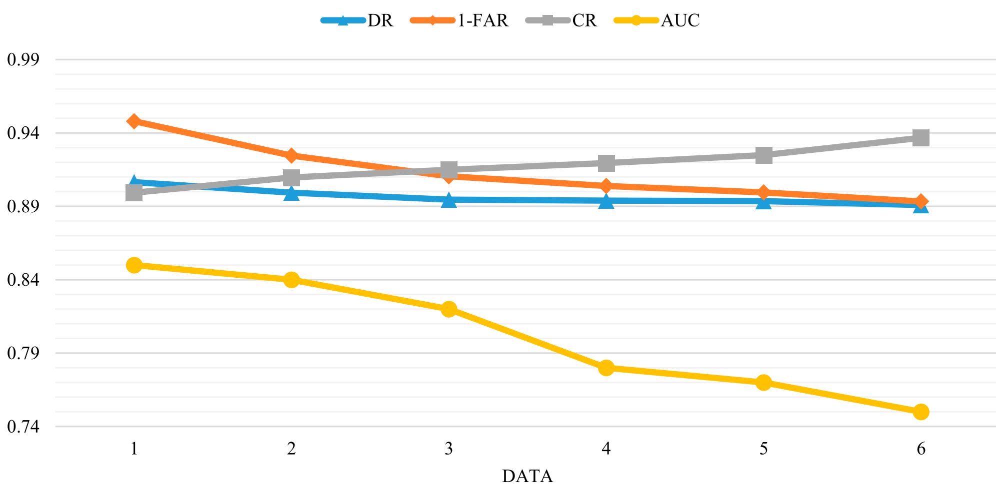


Figure 9. Description and results of experiment 2.

In the BPNN model, two parameters need to be set: the number of hidden layers and the number of hidden nodes in each hidden layer. In previous studies, it has been proven that one hidden layer is sufficient (Sheela and Deepa 2013). The number of hidden nodes in the hidden layer was set according to Sheela and Deepa (2013):

$$H_n = \frac{4n^2 + 3}{n^2 - 8} \tag{14}$$

where n represents the number of input variables, which equals 81. To conduct SVM, two parameters need to be set: gamma and soft margin C. Similar to the grid search method in Li et al. (2016a), these two parameters are set as 0.0625 and 16, respectively. In RF, only the

Table 4. A comparison of models trained on balanced samples with models trained on imbalanced samples.

	Balanced samples				Imbalanced samples			
Model	DR	FAR	CR	AUC	DR	FAR	CR	AUC
TSSAE	**0.9064**	**0.0520**	**0.8992**	**0.8518**	**0.8935**	0.1005	**0.9249**	0.7727
SVM	0.8682	0.0689	0.8771	0.8399	0.8627	**0.0701**	0.9247	**0.7883**
RF	0.8527	0.0722	0.8589	0.8332	0.8432	0.1134	0.9093	0.7639
BPNN	0.8456	0.0791	0.8503	0.8293	0.8399	0.1204	0.9011	0.7593
California	0.6898	0.0930	0.7394	0.6602	0.6549	0.1495	0.7387	0.7290
McMaster	0.6904	0.1237	0.7459	0.6836	0.6893	0.1239	0.7529	0.6859
SSAE	0.7083	0.1599	0.7340	0.7829	0.6693	0.1763	0.7993	0.7482
TSAE	0.8502	0.0599	0.8755	0.8403	0.8493	0.1127	0.9089	0.7682

number of trees needs to be set. After some calculations, 100 trees was selected because the accuracy does not increase after the number of trees reaches 100. When implementing the ANN and SVM, the variables need to be normalized, but the RF uses raw variables. The implementations of the two statistical models, California and McMaster, can be found in Karim and Adeli (2002) and Hall, Shi, and Atala (1993) which provide detailed parameter settings.

For SSAE, the 9 spatial variables in Table 1 were used as input. A single layer with 5 hidden nodes is sufficient. For TSAE, we used the 78 temporal variables listed in Table 1. The number of hidden layers was set to 3 with 39, 20, and 10 hidden nodes. For TSSAE, all the variables in Table 1 were used as input, and the parameters were the same as those in the TSAE. The performances of the proposed detection models and the previous commonly used models trained on imbalanced samples and balanced samples are listed in Table 4. On the imbalanced and balanced datasets, the ratios of incident samples and non-incident samples are 139:695 and 139:139, respectively and shows that our proposed model significantly outperforms the benchmark models on most of the criteria. The results indicate that the proposed TSSAE obtain its best performance on balanced samples but still achieves better performances than other models on imbalanced samples. In our proposed hybrid model, the imbalanced samples can be balanced; therefore, the hybrid model obtains the best performance.

5.4. Real-time analysis of the hybrid model

In previous studies, models were built and tested using data acquired during incidents (Chen, Wang, and Van Zuylen 2009; Chen and Wang 2009). For example, suppose the incident duration time is 5 min, traffic flow data can be obtained every 30 s from the detectors; then, all 10 samples collected during those 5 min are used as incident samples. However, in practice, the system should detect the incident as quickly as possible. Therefore, only the sample taken closest to the time when an incident happens would be used as the incident sample. In this study, to confirm this idea, we tested the proposed model using only the dataset collect just after an incident occurred (I-1) and the constructed 30-s incident dataset (I-2), 60-s incident dataset (I-3), 90-s incident dataset (I-4) and 120-s incident dataset (I-5). The corresponding non-incident samples (NI-1, NI-2, NI-3, NI-4, NI-5) were selected from the non-incident dataset. The training samples and test samples of the models are listed in Table 5.

Table 5. Model training and test samples.

Model	Training samples	Test samples	DR	FAR	CR	AUC
TSSAE-1	70% of I-1, NI-1	30% of I-1, NI-1	**0.9064**	**0.0520**	**0.8992**	**0.8518**
TSSAE-2	70% of I-1, NI-1, I-2, NI-2	30% of I-1, NI-1	0.8923	0.0651	0.8901	0.8382
TSSAE-3	70% of I-1, NI-1, I-3, NI-3	30% of I-1, NI-1	0.8811	0.0819	0.8852	0.8218
TSSAE-4	70% of I-1, NI-1, I-4, NI-4	30% of I-1, NI-1	0.8602	0.0993	0.8529	0.8066
TSSAE-5	70% of I-1, NI-1, I-5, NI-5	30% of I-1, NI-1	0.8524	0.0992	0.8371	0.8012

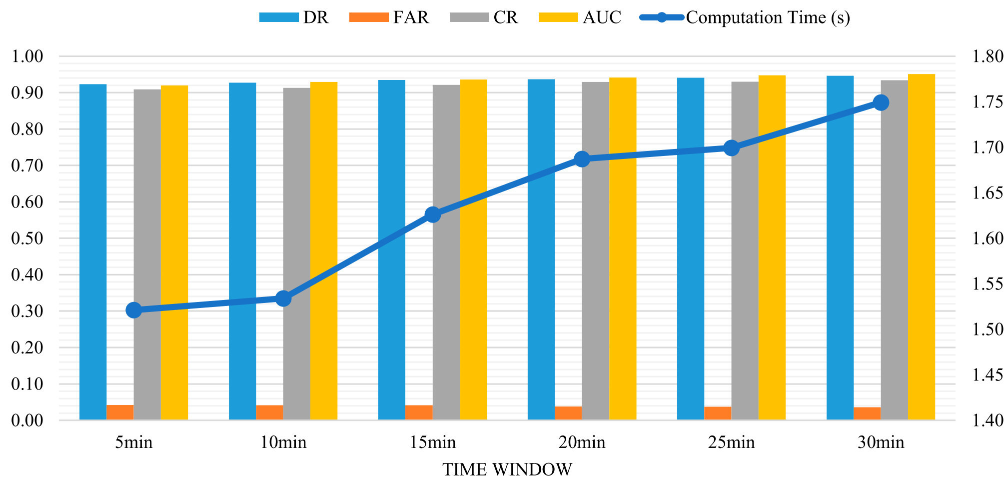


Figure 10. Analysis of the effect of the time window.

As Table 5 shows, the TSSAE-1 model obtains the best detection result, with DR, FAR, CR and AUC values of 90.64%, 5.20%, 89.92% and 0.8518, respectively, while the TSSAE-5 model obtains the worst detection result, with DR, FAR, CR and AUC values of 85.24%, 9.92%, 83.71% and 0.8012, respectively. The results show that except for FAR, the results of the detection model decrease steadily from TSSAE-1 to TSSAE-5. This result occurs because as the time when the incident samples obtained post-incident increases, the traffic flow differs more dramatically. Thus, the trained model can accurately classify these incident samples and non-incident samples, but it is not as sensitive to incident samples collected close to the time when an incident occurs. The results indicate that using samples collected just after incidents to train the proposed incident detection model can improve it, giving it strong real-time capacity and good performance. The proposed model was trained using input samples collected at different times to analyze the effect of the time window on model accuracy. The time window means the time ahead of the incident. For 5, 10, 15, 20, 25, and 30 min ahead of an incident, the number of samples are 10, 20, 30, 40, 50, and 60, respectively. The results are shown in Figure 10. As the time window of the input becomes longer, the accuracy of the model increases slightly, but the computation time increases markedly.

6. Conclusion

Traffic incident detection is an important part of a traffic monitoring system. Incident detection can help practitioners create management plans that improve traffic safety and can help travelers select the best travel routes to avoid congestion. However, incident samples are difficult to collect, which stifles research and innovation. Moreover, achieving real-time capability in an incident model is also difficult. In this study, to solve these problems, a hybrid model coupling a GAN and a TSSAE is proposed. Using a real-world dataset extracted from I-80 in California, the model is evaluated from several aspects.

The results indicate that our proposed model increases the detection accuracy and improves the real-time capability. Our proposed scheme provides better performance for the following reasons:

- Our proposed spatial and temporal variable selection rules are useful and consider both raw traffic flow variables and some extended variables.
- The generated samples not only expand the sample size but also improve sample diversity. Thus, generated samples can solve both the small sample size problem and the imbalanced sample problem.
- The proposed TSSAE captures the correlations among the selected spatial and temporal variables.
- The sample selection method selects samples just after incidents that improve the detection model, giving it strong real-time characteristics.

Although the proposed hybrid model can effectively and efficiently detect traffic incidents, some improvements could be made in future studies. First, external factors, such as weather conditions, should be considered in the sample variables. Second, ranking the contributions of the variables is important; future studies should improve the interpretability of our model.

Acknowledgments

The authors thank the anonymous reviewers and authors of cited papers for their detailed comments, without which this work would not have been possible.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The project presented in this article is supported by the Sichuan New Generation Artificial Intelligence Special Programme (No. 2018GZDZX0029) and the Shenzhen Science and Technology program (No. KQTD20180412181337494).

References

Abdel-Aty, Mohamed, Nizam Uddin, Anurag Pande, Fathy Abdalla, and Liang Hsia. 2004. "Predicting Freeway Crashes From Loop Detector Data by Matched Case-control Logistic Regression." *Transportation Research Record: Journal of the Transportation Research Board* 1897 (1): 88–95.

Adeli, H., and A. Samant. 2000. "An Adaptive Conjugate Gradient Neural Network–wavelet Model for Traffic Incident Detection." *Computer-Aided Civil and Infrastructure Engineering* 15 (4): 251–260.

Ahmed, Mohammed S., and Allen R. Cook. 1979. "Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques." Transportation Research Board Annual Meeting, 1979.

Asakura, Yasuo, Takahiko Kusakabe, Long Xuan Nguyen, and Takamasa Ushiki. 2017. "Incident Detection Methods Using Probe Vehicles with on-board GPS Equipment." *Transportation Research Part C: Emerging Technologies* 81: 330–341.

Bengio, Yoshua, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2007. "Greedy Layer-Wise Training of Deep Networks." In *Advances in Neural Information Processing Systems*, edited by Bernhard Scholkopf, John C. Platt, and Thomas Hofmann, 153–160. Vancouver: MIT Press.

Chassiakos, Athanasios P., and Yorgos J. Stephanedes. 1993. "Smoothing Algorithms for Incident Detection." *Transportation Research Record* (1394): 8–16.

Chen, Shuyan, and Wei Wang. 2009. "Decision Tree Learning for Freeway Automatic Incident Detection." *Expert Systems with Applications* 36 (2): 4101–4105.

Chen, Shuyan, Wei Wang, and Henk Van Zuylen. 2009. "Construct Support Vector Machine Ensemble to Detect Traffic Incident." *Expert Systems with Applications* 36 (8): 10976–10986.

Cookson, Graham. 2018. "Inrix Global Traffic Scorecard."

Denton, Emily L., Soumith Chintala, Arthur Szlam, and Rob Fergus. 2015. "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks." In *Advances in Neural Information Processing Systems*, edited by Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, 1486–1494. Montreal: MIT Press.

Ghosh, Arna, Biswarup Bhattacharya, and Somnath Basu Roy Chowdhury. 2016. "Sad-gan: Synthetic Autonomous Driving using Generative Adversarial Networks." arXiv preprint arXiv:1611.08788.

Ghosh, Bidisha, and Damien P. Smith. 2014. "Customization of Automatic Incident Detection Algorithms for Signalized Urban Arterials." *Journal of Intelligent Transportation Systems* 18 (4): 426–441.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. "Generative Adversarial Nets." In *Advances in Neural Information Processing Systems*, edited by Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, 2672–2680. Montreal: MIT Press.

Hall, Fred L., Yong Shi, and George Atala. 1993. "On-line Testing of the McMaster Incident Detection Algorithm under Recurrent Congestion."

Hinton, Geoffrey E., Simon Osindero, and Yee-Whye Teh. 2006. "A Fast Learning Algorithm for Deep Belief Nets." *Neural Computation* 18 (7): 1527–1554.

Hossain, Moinul, and Yasunori Muromachi. 2012. "A Bayesian Network Based Framework for Real-time Crash Prediction on the Basic Freeway Segments of Urban Expressways." *Accident Analysis & Prevention* 45: 373–381.

Karim, Asim, and Hojjat Adeli. 2002. "Comparison of Fuzzy-wavelet Radial Basis Function Neural Network Freeway Incident Detection Model with California Algorithm." *Journal of Transportation Engineering* 128 (1): 21–30.

Kuefler, Alex, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer. 2017. "Imitating Driver Behavior with Generative Adversarial Networks." In *Intelligent Vehicles Symposium (IV), 2017 IEEE*, 204–211. IEEE.

Li, Linchao, Shanglu He, Jian Zhang, and Bin Ran. 2016a. "Short-term Highway Traffic Flow Prediction Based on a Hybrid Strategy Considering Temporal–spatial Information." *Journal of Advanced Transportation* 50 (8): 2029–2040.

Li, Linchao, Shanglu He, Jian Zhang, and Fan Yang. 2016b. "Bagging-SVMs Algorithm-Based Traffic Incident Detection." CICTP 2016.

Li, Linchao, Lingqiao Qin, Xu Qu, Jian Zhang, Yonggang Wang, and Bin Ran. 2019. "Day-ahead Traffic Flow Forecasting Based on a Deep Belief Network Optimized by the Multi-objective Particle Swarm Algorithm." *Knowledge-Based Systems* 172: 1–14.

Li, Linchao, Xi Sheng, Bowen Du, Yonggang Wang, and Bin Ran. 2020. "A Deep Fusion Model Based on Restricted Boltzmann Machines for Traffic Accident Duration Prediction." *Engineering Applications of Artificial Intelligence* 93: 103686.

Li, Linchao, Jian Zhang, Yonggang Wang, and Bin Ran. 2018. "Missing Value Imputation for Traffic-related Time Series Data Based on a Multi-view Learning Method." *IEEE Transactions on Intelligent Transportation Systems* 20 (8): 2933–2943.

Liang, Yunyi, Zhiyong Cui, Yu Tian, Huimiao Chen, and Yinhai Wang. 2018. "A Deep Generative Adversarial Architecture for Network-Wide Spatial-Temporal Traffic State Estimation." arXiv preprint arXiv:1801.03818.

Lin, Yi, Linchao Li, Hailong Jing, Bin Ran, and Dongye Sun. 2020. "Automated Traffic Incident Detection with a Smaller Dataset Based on Generative Adversarial Networks." *Accident Analysis & Prevention*144: 105628.

Liu, Qingchao, Jian Lu, and Shuyan Chen. 2013. "Traffic Incident Detection using Random Forest." Technical Report.

Lv, Yisheng, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. 2015. "Traffic Flow Prediction with Big Data: A Deep Learning Approach." *IEEE Transactions on Intelligent Transportation Systems*16 (2): 865–873.

Ma, Xiaolei, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. 2017. "Learning Traffic As Images: a Deep Convolutional Neural Network for Large-scale Transportation Network Speed Prediction." *Sensors* 17 (4): 818.

Ma, Xiaolei, Haiyang Yu, Yunpeng Wang, and Yinhai Wang. 2015. "Large-scale Transportation Network Congestion Evolution Prediction Using Deep Learning Theory." *PloS One* 10 (3): e0119044.

Mak, Chin Long, and Henry S. L. Fan. 2007. "Development of Dual-station Automated Expressway Incident Detection Algorithms." *IEEE Transactions on Intelligent Transportation Systems* 8 (3): 480–490.

Mirza, Mehdi, and Simon Osindero. 2014. "Conditional Generative Adversarial Nets." arXiv preprint arXiv:1411.1784.

Odena, Augustus, Christopher Olah, and Jonathon Shlens. 2016. "Conditional Image Synthesis with Auxiliary Classifier Gans." arXiv preprint arXiv:1610.09585.

Qu, Xu, Wei Wang, Wen-fu Wang, and Pan Liu. 2017. "Real-time Rear-end Crash Potential Prediction on Freeways." *Journal of Central South University* 24 (11): 2664–2673.

Radford, Alec, Luke Metz, and Soumith Chintala. 2015. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks." arXiv preprint arXiv:1511.06434.

Reed, Scott, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. 2016. "Generative Adversarial Text to Image Synthesis." arXiv preprint arXiv:1605.05396.

Samant, A., and H. Adeli. 2000. "Feature Extraction for Traffic Incident Detection Using Wavelet Transform and Linear Discriminant Analysis." *Computer-Aided Civil and Infrastructure Engineering* 15 (4): 241–250.

Samant, A., and H. Adeli. 2001. "Enhancing Neural Network Traffic Incident-detection Algorithms Using Wavelets." *Computer-Aided Civil and Infrastructure Engineering* 16 (4): 239–245.

Sheela, K. Gnana, and Subramaniam N. Deepa. 2013. "Review on Methods to Fix Number of Hidden Neurons in Neural Networks." *Mathematical Problems in Engineering* 2013: 1–11.

Wang, Ren, Shimao Fan, and Daniel B. Work. 2016. "Efficient Multiple Model Particle Filtering for Joint Traffic State Estimation and Incident Detection." *Transportation Research Part C: Emerging Technologies* 71: 521–537.

Wang, Ren, Daniel B. Work, and Richard Sowers. 2016. "Multiple Model Particle Filter for Traffic Estimation and Incident Detection." *IEEE Transactions on Intelligent Transportation Systems* 17 (12): 3461–3470.

Wu, Yuankai, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. 2018. "A Hybrid Deep Learning Based Traffic Flow Prediction Method and Its Understanding." *Transportation Research Part C: Emerging Technologies* 90: 166–180.

Xiao, Jianli, and Yuncai Liu. 2012. "Traffic Incident Detection Using Multiple-kernel Support Vector Machine." *Transportation Research Record: Journal of the Transportation Research Board* 2324 (1): 44–52.

Xu, Chengcheng, Wei Wang, and Pan Liu. 2013. "A Genetic Programming Model for Real-time Crash Prediction on Freeways." *IEEE Transactions on Intelligent Transportation Systems* 14 (2): 574–586.

Xu, Chengcheng, Wei Wang, Pan Liu, and Zhibin Li. 2015. "Calibration of Crash Risk Models on Freeways with Limited Real-time Traffic Data Using Bayesian Meta-analysis and Bayesian Inference Approach." *Accident Analysis & Prevention* 85: 207–218.

Yu, Rongjie, Xuesong Wang, Kui Yang, and Mohamed Abdel-Aty. 2016. "Crash Risk Analysis for Shanghai Urban Expressways: A Bayesian Semi-parametric Modeling Approach." *Accident Analysis & Prevention* 95: 495–502.

Yuan, Fang, and Ruey Long Cheu. 2003. "Incident Detection Using Support Vector Machines." *Transportation Research Part C: Emerging Technologies* 11 (3–4): 309–328.

Zhu, Lin, Fangce Guo, Rajesh Krishnan, and John W. Polak. 2018. "The Use of Convolutional Neural Networks for Traffic Incident Detection at a Network Level." Technical Report.