

## Research Article

# Online Traffic Accident Spatial-Temporal Post-Impact Prediction Model on Highways Based on Spiking Neural Networks

Duowei Li , Jianping Wu , and Depin Peng

Department of Civil Engineering, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Jianping Wu; [jianpingwu@tsinghua.edu.cn](mailto:jianpingwu@tsinghua.edu.cn)

Received 2 August 2021; Accepted 29 October 2021; Published 2 December 2021

Academic Editor: Alain Lambert

Copyright © 2021 Duowei Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic accident management as an approach to improve public security and reduce economic losses has received public attention for a long time, among which traffic accidents post-impact prediction (TAPIP) is one of the most important procedures. However, existing systems and methodologies for TAPIP are insufficient for addressing the problem. The drawbacks include ignoring the recovery process after clearance and failing to make comprehensive prediction in both time and space domain. To this end, we build a 3-stage TAPIP model on highways, using the technology of spiking neural networks (SNNs) and convolutional neural networks (CNNs). By dividing the accident lifetime into two phases, i.e., clean-up phase and recovery phase, the model extracts characteristics in each phase and achieves prediction of spatial-temporal post-impact variables (e.g., clean-up time, recovery time, and accumulative queue length). The framework takes advantage of SNNs to efficiently capture accident spatial-temporal features and CNNs to precisely represent the traffic environment. Integrated with an adaptation and updating mechanism, the whole system works autonomously in an online manner that continues to self-improve during usage. By testing with a new dataset CASTA pertaining to California statewide traffic accidents on highways collected in four years, we prove that the proposed model achieves higher prediction accuracy than other methods (e.g., KNN, shockwave theory, and ANNs). This work is the introduction of SNNs in the traffic accident prediction domain and also a complete description of post-impact in the whole accident lifetime.

## 1. Introduction

People's living standards have increased all over the world, leading to an increase in the ownership of private vehicles [1]. While private vehicles have improved people's traveling experience, they have also contributed to several traffic problems, where traffic safety is one of the main concerns. According to statistical data released by World Health Organization in 2004, road traffic accidents are among the main causes of deaths and injuries all over the world, leading to 1.2 million deaths and 50 million injuries each year [2]. In 2019, the number of traffic accidents in China was 248,000, and the direct property loss was as high as 1.35 billion yuan (China Statistical Yearbook).

Besides the costs of fatalities and injuries, indirect effects brought by traffic accidents such as congestion and energy consumption also have tremendous impacts on socioeconomic development. Typically, accident congestion can

spread rapidly and even cause a chain breakdown in the entire system, especially in closed roads like highways. In severe cases, the delays of clearance for an accident may increase the occurrence probability of a secondary accident [3]. In 1988, the total extra travel time and fuel consumption caused by traffic congestion, both regular and occasional, in 50 major cities in the United States was estimated at 35 billion U.S. dollars [4].

Traffic accident management (TAM) is of great importance to government and transportation agencies, which can not only improve public security but also reduce economic losses. There are usually two main tasks in TAM: analysis and prediction. By analyzing the features of historical accident records, accident prediction can be realized to foresee the time and place of the accident in advance. Furthermore, traffic accidents post-impact prediction (TAPIP) is also an essential part in TAM but have not caught enough attention. When an accident occurs, if the spatial and temporal impact scope can

be quickly determined, then the transportation agency can induce vehicles to avoid unnecessary congestion, and travelers can also plan their routes in advance.

The lifetime of a traffic accident is usually divided into five phases, namely, detection, verification, response, clearance, and recovery [5]. Most TAPIP-related research studies till now concentrate only on the first four phases, which means the duration from accident occurrence to the reopening of all blocked traffic lanes. The existing research studies use traffic wave theory, regression model, and other methods to analyze the influence of various factors on accident indicators, but have not established a complete relationship between the influencing factors and the spatial-temporal impact of the accidents. Furthermore, the clearance is never the end of an accident, and the process that the traffic gradually recovers to normal conditions cannot be ignored. For the convenience of expression, this work proposes a two-phase accident lifetime. As shown in Figure 1, the four phases before clearance are grouped into clean-up phase, while the recovery phase remains unchanged. The reason for this division is that in the clean-up phase, at least one lane will be blocked, while after that, the whole lanes will be reopened.

As the third-generation neuron model, spiking neural networks (SNNs) [6] are closer to biological principle compared with other artificial neural networks (ANNs). Due to the advantage of being sensitive to spatial-temporal characteristics of information [7], SNNs already improve the traditional neural models on accuracy in many application areas. However, SNNs till now are rarely used in traffic domain and have even never been applied to accident prediction related research.

This study proposes a traffic accident spatial-temporal post-impact prediction model on highways, using the technology of SNNs and CNNs. In the 3-stage model, the duration of clean-up phase (clean-up time) is first estimated, and then, several post-impact variables such as recovery time and accumulative queue length are predicted. Additional online stage is used to make adaptation and update the model. In this regard, the main contributions of this work can be summarized as follows:

- (1) The development of a framework to predict post-impact in the complete accident lifetime, which contains the recovery process.
- (2) The prediction of highway accident post-impact in both spatial and temporal domain, and the outputs are specific and effective variables that can help both transportation agencies and travelers.
- (3) The first use of SNNs in the traffic accident prediction domain and the implementation of encoding the traffic spatial-temporal features into time spikes.
- (4) The processing of a new highway spatial-temporal traffic accident dataset CASTA and the use of it in testing the performance of the proposed model.
- (5) The proposal of an online adaptation and updating method to revise the error and modify the model, for self-improvement during usage.

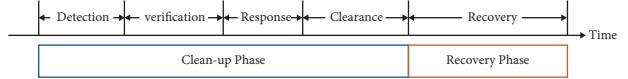


FIGURE 1: Two-phase traffic accident lifetime.

The rest of this study is organized as follows. Section 2 describes literature review on SNNs and TAPIP. Section 3 defines the general framework of the 3-stage prediction model, including the specific input, output, and structure of SNNs and CNNs. The processing of the new dataset is introduced in Section 4, and Section 5 presents the experiment results. Finally, Section 6 provides concluding statements on our work.

## 2. Literature Review

**2.1. Spiking Neural Networks in Traffic Domain.** With the rapid development of artificial intelligence and computer technology, deep learning has been widely applied in traffic domain, including traffic state prediction [8, 9], traffic signal control [1, 10], and driving model development [11].

The works mentioned above are all based on artificial neural networks (ANNs), where the neurons use differentiable, nonlinear activation functions. Although ANNs have achieved great performance in wide areas including classification, pattern recognition, and prediction, they are fundamentally different in structure, neural computations, and learning rule compared to the brain [7]. The way of information propagation between units is one of the most obvious differences. While ANNs rely on 32 bit or 64 bit continuous messages sent between units [12], the neurons in biological brain broadcast trains of action potentials, also known as spike trains to downstream neurons. As the third-generation neuron model, spiking neural networks (SNNs) are close to biological principle by utilizing spikes. When dealing with complex or large datasets, while ANNs face the concern of huge computational consumption, SNNs achieve energy efficiency since output spike trains can be made sparse in time [13]. Besides that, SNNs improve the traditional neural models on accuracy due to the advantage of being sensitive to the spatial-temporal characteristics of information [7]. Although still in the early stages of development, SNNs have become the focus of many application areas and made progress in visual processing [12], speech recognition [14], and medical diagnosis [15].

However, SNNs till now are rarely used in traffic domain, despite their specialized capability in representing spatial-temporal data. Laña et al. [16] presents an approach for spatial-temporal road traffic forecasting that relies on the adoption of the NeuCube architecture based on SNNs. Taking advantages of the NeuCube platform, this work focuses on the spatially-aware traffic variable forecasts and on the exploration of the spatial-temporal relationships among different sensor locations within a traffic network. In another work of Laña et al. [17], an evolving spiking neural networks (eSNNs) based adaptive long-term traffic state estimation model is proposed. By using similarity-based clustering of daily traffic volume data and monitoring them

in real time, the model encodes traffic data into spikes over the time domain effectively and achieves high accuracy in online prediction when new data samples arrive.

**2.2. Post-Impact Prediction of Traffic Accident.** At present, the main research methods in predicting the impact of highway accidents are the traffic wave theory [18, 19], vehicle arriving-departure model [20, 21], regression model [22, 23], decision tree model [5], and ANNs [23, 24]. Yu et al. [18] analyzed the accumulation and dissipation process of the accident location on two-lane highway, using the shockwave theory as a foundation, and finally estimated the spatial-temporal impact of the accident. However, such methods based on theoretical analysis considered only limited influencing factors (flow and density of upstream) and only showed general laws that lead to poor applicability. Zhu et al. [23] established two kinds of forecasting models of spatial-temporal impact for traffic accidents based on nonlinear regression and BP neural networks, where the influence of factors were analyzed such as upstream flowrate, ratio of cargos, and accident handling time. The results showed that the BP neural network achieves higher accuracy. Lin and Li [24] embedded three machine learning algorithms in a hierarchical scheme to perform sequential prediction. The result outperformed others by achieving a MAPE range of 5.5–53.8%. However, it just considered the time duration of an accident rather than spatial impact. In contrast, Lee et al. [25] only focused on estimation of vehicle accident queue length using the ANN model with relatively abundant influencing factors.

The methods above theoretically described and inferred the characteristics of incidental accidents and analyzed the influence of various factors on accident indicators. But they have not established a complete relationship model between the influencing factors and the spatial-temporal impact of the accidents. Furthermore, related works all used one model to output all prediction variables regardless of the characteristics in different phases of accidents. However, the five phases are closely related and can have great influence on each other. Thus, in order to fully capture the spatial-temporal features in the whole process, the prediction should be made separately according to the order of phase.

Since the prediction of accident impact highly relies on feature extraction of spatial and temporal data, SNNs are considered to be a more suitable method due to its characteristics compared with the models mentioned above. However, no such SNNs-related models have been proposed till now.

### 3. Methodology

This work proposes a spatial-temporal post-impact prediction model of traffic accidents. The 3-stage process is shown in Figures 2 and 3, where Figure 2 corresponds to the training process using historic accident records (training set) and Figure 3 corresponds to the predicting process using new accident records (test set). Stage 1 is the prediction of accident clean-up time; it uses clusters to define length

hierarchy of clean-up time and then extract accident features to train the SNNs classifier. Stage 2 concerns with a prediction model of spatial-temporal impact based on CNNs and backpropagation neural networks (BPNNs), feeds with flow and speed data, and outputs the recovery time and accumulative queue length. When a new accident record comes to the model after being processed with the above two stages, another online updating stage will be promoted to correct error and revise model. The notations used frequently in the 3-stage model are given in Table 1.

**3.1. Stage 1: Accident Clean-Up Time Prediction.** According to the two-phase accident lifetime defined in Section 1, phases are not completely independent and can influence each other. Therefore, in order to figure out the spatial-temporal impact in the whole process, we need to predict the duration of clean-up phase first.

Mean-shift clustering is first performed on the clean-up time, which groups the clean-up time into several intervals. Thus, the accidents with similar cleaning difficulty are labeled as a cluster. Furthermore, the accident records with attributes and corresponding class labels are input to train an SNNs classifier. Whenever a new accident record comes in, the accident attributes are extracted and then classified using the trained classifier. The predicting clean-up time is represented by the centroid value of its corresponding cluster.

**3.1.1. Clustering Using Mean-Shift.** According to the statistics in Section 4, the clean-up time of the accidents distributes unevenly throughout the interval. It exists the case where some of the interval lack records. Thus, to improve the prediction accuracy and remove exceptional, clustering the clean-up time into groups and using the centroid value to present the predicted value is better than a direct prediction model.

The mean-shift algorithm [26] is chosen to cluster the clean-up time for two main reasons: it does not require a predefined number of clusters; and such density-based algorithms are less affected by the mean value compared with the K-means algorithm. The key operation of the algorithm is to calculate the drift vector of the centroid through the data density change in the region of data, so as to move the centroid in the next iteration until it reaches the maximum density. Algorithm 1 shows the procedure of clustering the traffic records according to the clean-up time. The bandwidth is set as 10.

**3.1.2. Accident Characterization.** When a new accident record comes to the model, in order to predict the clean-up time, the cluster that the records belongs to needs to be figured out. Thus, a number of attributes that represent the characteristics of accidents are used to train a classifier, which outputs the corresponding cluster.

In a previous work, three kinds of attributes are usually extracted to demonstrate the accident: accident information (e.g. accident type [24, 25, 27–29], blocked area size [5, 25], number of vehicles involved [5, 25, 28], and casualties

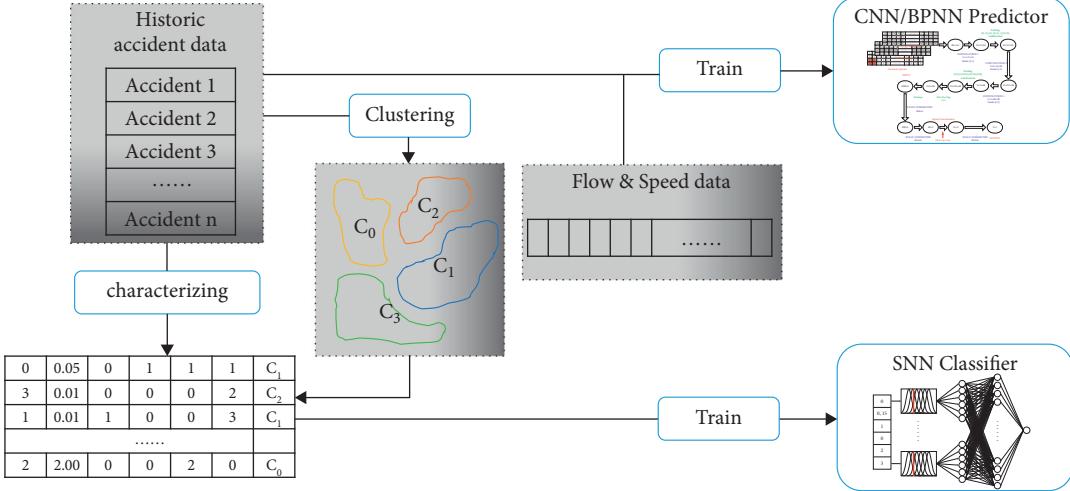


FIGURE 2: 3-stage model for accident post-impact prediction-training process.

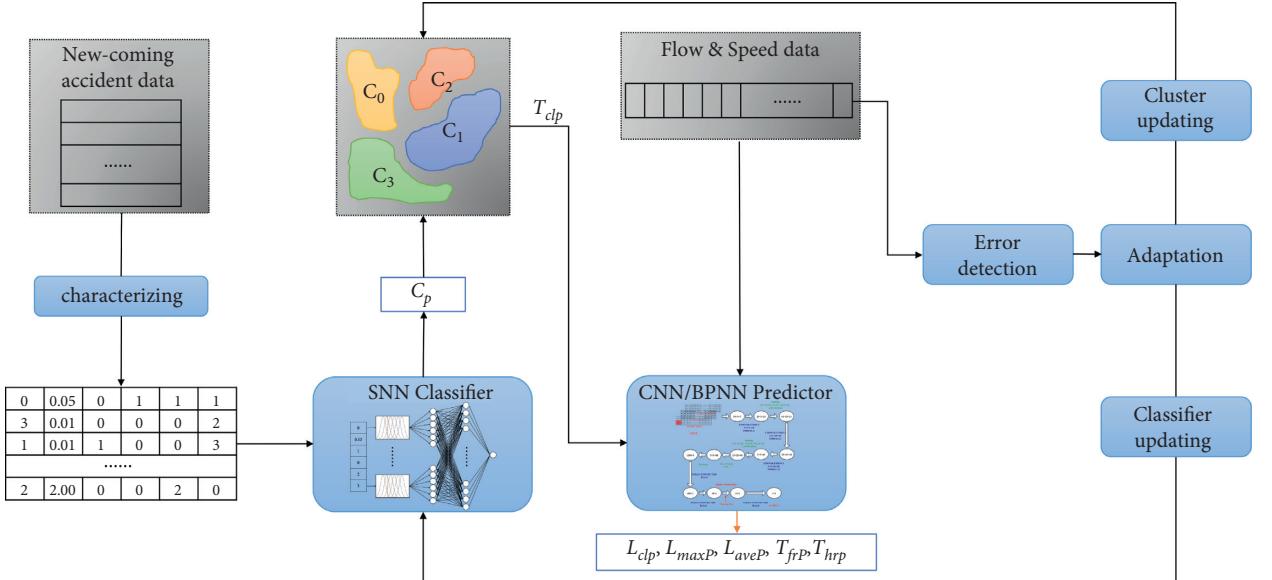


FIGURE 3: 3-stage model for accident post-impact prediction-predicting process.

TABLE 1: Notations.

Notation	Meaning	Units
$C_A$	The actual cluster that the clean-up time of an accident belongs to	—
$C_p$	The predicted cluster that the clean-up time of an accident belongs to	—
$T_{clA}$	The actual clean-up time	Minute
$T_{clP}$	The predicted clean-up time	Minute
$L_{clA}$	The actual accumulative queue length in the clean-up phase	Mile
$L_{clP}$	The predicted accumulative queue length in the clean-up phase	Mile
$L_{maxA}$	The actual maximum accumulative queue length	Mile
$L_{maxP}$	The predicted maximum accumulative queue length	Mile
$L_{aveA}$	The actual average accumulative queue length	Mile
$L_{aveP}$	The predicted average accumulative queue length	Mile
$T_{frA}$	The actual full-recovery time	Minute
$T_{frP}$	The predicted full-recovery time	Minute
$T_{hrA}$	The actual half-recovery time	Minute
$T_{hrP}$	The predicted half-recovery time	Minute

```

(1) Input
(2)  $\{T_c\}$  := dataset of clean-up time over which the procedure is performed
(3) Initialization
(4)  $r$  := the radius of a region
(5)  $d$  := the minimum distance between clusters
(6)  $F(i, C^j)$  := probability that point  $i$  belongs to Cluster  $C^j$ 
(7) Repeat
(8) Randomly select an unvisited point from the dataset as the starting centre point  $o^c$ 
(9) Repeat
(10) Find all the points that appear in the circle  $\odot(o^c, r)$  and group into cluster  $C^{o^c}$ 
(11) For point  $i$  in  $C^{o^c}$ ,  $F(i, C^{o^c}) \leftarrow F(i, C^{o^c}) + 1$ 
(12) Calculate the shift vector of  $o^c$  as  $M_c = 1/K \sum_{x_i \in C} (x_i - o^c)$ 
(13) Update  $o^c$  towards the shift vector as  $o^c \leftarrow o^c + M_c$ 
(14) Until  $M_c$  reaches minimum threshold
(15) If  $\exists C^{o^i}$ ,  $\text{distance}(C^{o^c}, C^{o^i}) < d$ , then merge  $C^{o^c}$  and  $C^{o^i}$ 
(16) Else, label  $C^{o^c}$  as a new cluster
(17) Until all points are visited
(18) For point  $i$  in  $\{T_c\}$ ,  $i$  belongs to cluster  $C$  with maximum probability  $F(i, C) = \max_{C^j} F(i, C^j)$ 
(19) Output
(20)  $\{C^c\}$  := all clusters
(21)  $\{C_{T_c}\}$  := clusters that each point in  $\{T_c\}$  belongs to

```

ALGORITHM 1: Mean-shift clustering of clean-up time.

[5, 25, 29]), spatial-temporal attributes (e.g. lane location [25, 27, 29], road type [24, 25], and time [5, 24, 25, 27–29]), and weather information (e.g. humidity [24, 25, 28], wind [24, 25], and visibility [24, 25]). In this model, considering the dataset used, 6 attributes are selected as the input in this stage:

- (i) Block lanes: a number between 1 and 6 represents the number of lanes blocked by the accident. The value is determined according to TMC event code [30].
- (ii) Block distance: a value represents the length of road blocked by the accident in miles.
- (iii) Day or night: a binary digit, where 0 represents daytime while 1 represents night.
- (iv) Workday or not: a binary digit, where 0 represents workday while 1 represents holiday or weekend.
- (v) Peak hour or not: a number between 0 and 2, where 0 represents morning peak (8 a.m.–10 a.m.), 1 represents evening peak (4 p.m.–6 p.m.), and 2 represents the normal period [31].
- (vi) Weather condition: a number between 0 and 4, which represents sunny/rain/snow/thunderstorm/fog, respectively, when accident happens.

More detailed information of data processing can be found in Section 4.

**3.1.3. Classification Using SNNs.** A classifier is trained to match the accident attributes to the cluster it belongs to. Due to the characteristics of being sensitive to the spatial-temporal features, SNNs are applied in building the classifier. It is the first time of applying SNNs in traffic accident prediction domain.

After being first proposed in 1997 by Maass [6], SNNs have attracted researchers to explore different research directions. The main technique directions of SNNs can be concluded in four main categories: encoding and decoding of spiritual information, neuron models and network simulation strategy, similarity measurement of spiking sequence, and synaptic weight learning rules [32].

The classification problem in this work concerns with the supervised learning in SNNs, which refers to finding a suitable synaptic weight matrix for a series of given input spike trains and target spike trains, so that the output of the neurons is as close as the corresponding target spike trains, that is, the error evaluation function of the two reaches minimum [33].

(1) *Leaky Integrate-and-Fire Neural Model.* Neural computing relies on the construction of the spiking neuron model. In order to do numerical analysis, an efficient mathematical neural model should be built based on the geometry and electrical properties of neuron. Leaky Integrate-and-Fire neural (LIF) model [34], as the single compartment model with fixed threshold which has been widely used in neural computing, can directly describe the relationship between membrane potential  $V$  and input electric current  $I$ . When  $V$  is between the resting potential  $V_{\text{rest}}$  and the threshold  $V_{\text{thre}}$ , it is given by the following equation:

$$\tau_m \frac{dV}{dt} = -(V - V_{\text{rest}}) + R_m I, \quad (1)$$

where  $\tau_m$  is a time constant, and  $I$  is the sum of current released by presynaptic neurons. Once  $V$  crosses  $V_{\text{thre}}$  from below, a spike is generated and  $V$  is reset to  $V_{\text{rest}}$ . The LIF model, with the advantages of using simple structure to realize accurate simulation of SNNs by event-driven simulation strategy, is chosen as the basic neuron model in our work.

(2) *Gaussian Tuning Curve Coding.* In the biological neural system, spikes are transmitted between neurons with different combinations. The corresponding rules and mechanisms of conversion between the stimulation signal and the spike trains are called neural coding [35]. When neuron feels external stimuli, encoding module encodes the stimulation signal to specific spike trains and responses; in turn, decoding module can estimate the stimulation signal from spike trains.

Researchers put forward various coding methods, which can be divided into two main categories: frequency-based coding [36] and time-based coding [37]. Evidence confirms that time-based coding is the more effective, since the time structure of the spike trains carries stimulus signals on a millisecond or even smaller scale, not just the average firing frequency [38]. With the introduction of more complex systems, single-neuron coding cannot effectively present the huge amount of information. Population coding [39] is thus proposed to solve complex information coding problem, where each neuron has a unique spike response distribution for a given stimulus, and the responses of the neuron population are combined to represent the overall information.

The population coding method used in this model is Gaussian tuning curve coding [40], as shown in Figure 4. A neuron covers a certain range of analog quantity in the form of Gaussian function, while the height of the corresponding Gaussian function to a certain value of the analog quantity determines the time for the neuron to emit spike. In this work, a population containing six neurons is used to encode the six accident attributes mentioned above. Figure 4 shows an example of how “block distance” of 0.18 miles is encoded into spike trains {0.79, 0.04, 0.52, 0.96, 1.00, 1.00} (ms).

(3) *Multi-ReSuMe Training Algorithm.* The supervised learning algorithm in SNNs can be divided into three categories based on the learning rules of synaptic weight: gradient-descent based method, synaptic plasticity-based method, and convolution-based method [41]. Gradient-descent based methods such as SpikeProp [40] learn from the error back propagation algorithm of traditional ANNs, where the state variables of the neuron model must have analytical expressions and lack biological realism. From a biological perspective, a training algorithm should update synaptic weights based on the temporal correlation of presynaptic and postsynaptic spikes, in keeping with the spike timing-dependent plasticity (STDP) theory [42]. In this work, a synaptic plasticity-based method called the multilayer remote supervised method (Multi-ReSuMe) [43] is used to train the classifier. Multi-ReSuMe extends the single-layer impulse neural network ReSuMe algorithm [41] to the multilayer network and overcomes some limitations as it can be applied to neurons firing multiple spikes, and it can in principle be applied to any linear neuron model.

The main contribution of Multi-ReSuMe is the combination of the STDP and anti-STDP mechanism in one process, where synaptic weight update between hidden layer  $H$  and output layer  $O$  satisfies the following equation:

$$\begin{aligned} \frac{d}{dt}w_{oh}(t) = & \frac{1}{n_h}S_h(t)\left[\int_0^\infty a^{pre}[S_o^d(t) - S_o^a(t)]ds\right] \\ & + \frac{1}{n_h}\left[S_o^d(t) - S_o^a(t)\right]\left[a + \int_0^\infty a^{post}(s)S_h(t-s)ds\right]. \end{aligned} \quad (2)$$

For any excitatory synaptic connection from hidden neuron  $h$  to output neuron  $o$ , a synaptic strength  $w_{oh}$  is potentiated whenever a target spike  $S_o^d(t)$  is observed and depressed whenever the trained neuron fires  $S_o^a(t)$ . Here,  $s$  denotes a delay between the presynaptic and postsynaptic firing times, where  $s = t_o^f - t_h^f$ . The role of the noncorrelative factor  $a$  in equation (2) is to adjust the average strength of the synaptic inputs so as to impose on a neuron a desired level of activity. The kernels  $a^{pre}$  and  $a^{post}$  are STDP-related terms that define the shape of a learning window  $W(s)$  [44],

$$W(s) = \begin{cases} a^{pre}(-s) = -A_- \cdot \exp\left(\frac{s}{\tau_-}\right), & s \leq 0, \\ a^{post}(s) = +A_+ \cdot \exp\left(\frac{-s}{\tau_+}\right), & s > 0, \end{cases} \quad (3)$$

where parameters  $A_+, A_- > 0$  are the amplitudes and  $\tau_+, \tau_- > 0$  are the time constants of the learning process.  $a^{pre}(-s)$  gives the weight change if the presynaptic spike (the spike of the hidden neuron occurs) comes after the postsynaptic spike (the spike of the output and target neurons), while  $a^{post}(s)$  gives the weight change if the presynaptic spike comes before the postsynaptic spike [43].

Similarly, synaptic weight update between input layer  $I$  and hidden layer  $H$  satisfies the following equation:

$$\begin{aligned} \frac{d}{dt}w_{hi}(t) = & \frac{1}{n_i n_h}S_i(t)\sum_{o \in O}\left[\int_0^\infty a^{pre}[S_o^d(t) - S_o^a(t)]\right]w_{oh} \\ & + \frac{1}{n_i n_h}\sum_{o \in O}\left[S_o^d(t) - S_o^a(t)\right] \\ & \left[a + \int_0^\infty a^{post}(s)S_i(t-s)ds\right]w_{oh}. \end{aligned} \quad (4)$$

As Figure 5 shows, the network has 36 input neurons corresponding to six accident attributes, and an output neuron representing the cluster that the input sample belongs to. The hidden layer contains 80 neurons which are fully connected with other layers. The output neuron's spike train contains a single spike, where the timing differs from each other, as given in Table 2. The process of building and training a Multi-ReSuMe-based classifier is given in Algorithm 2. The parameters are initialized as follows:  $A_+ = 1.2$ ,  $A_- = 0.5$ ,  $\tau_+ = \tau_- = 0.5$ , and  $a = 0.05$  [45].

**3.2. Stage 2: Accident Spatial-Temporal Post-Impact Prediction.** After the clean-up phase of an accident, the traffic flow will begin to recover from congestion to normal condition. Recovery phase is an important process which is often

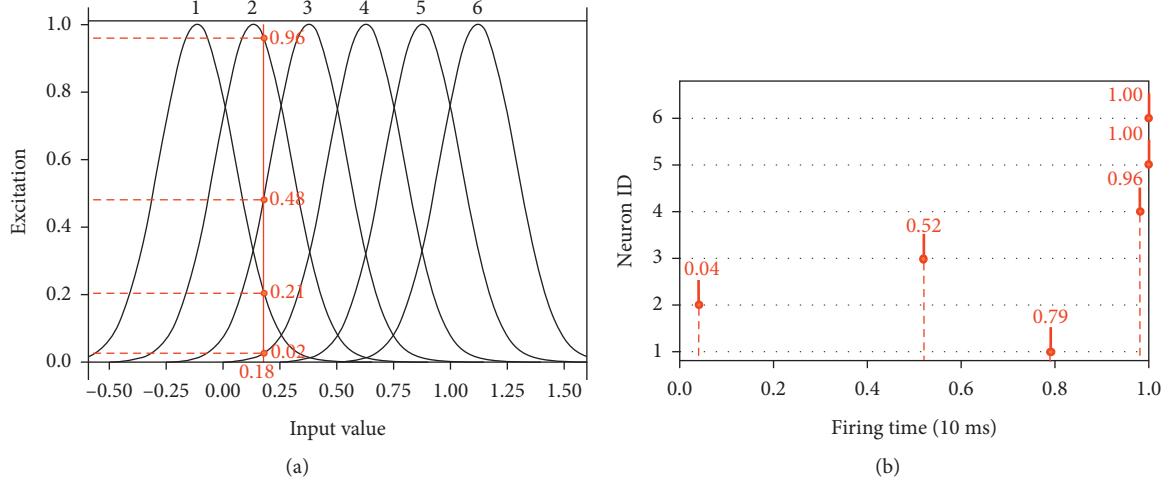


FIGURE 4: Accident attribute coding with Gaussian tuning curve. (a) The corresponding Gaussian curve of six neurons. (b) Firing time of six neurons.

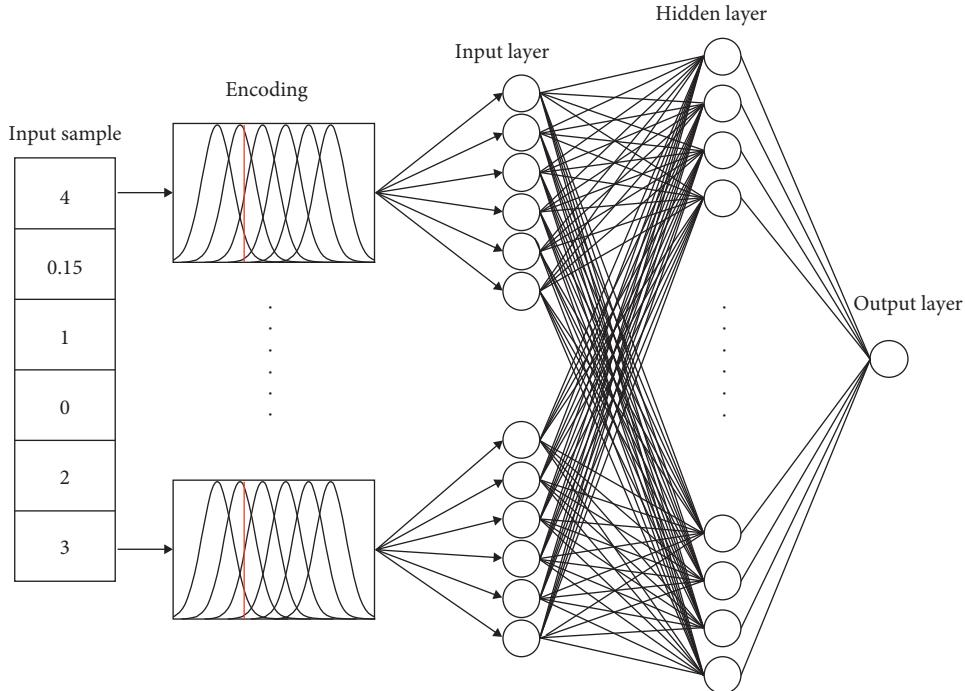


FIGURE 5: A Multi-ReSuMe-based SNNs classifier.

TABLE 2: The target spike-time of output neuron corresponding to each cluster.

Cluster no.	Output spike-time (ms)
0	0
1	1
2	2
3	3
:	:
$n$	$n$

```

(1) Definition
(2)  $\{A\}$  := Historic accident dataset
(3)  $A_i$  := accident data in  $\{A\}$  where  $A_i = (a_1, a_2, a_3, a_4, a_5, a_6, c)$ 
(4)  $e_{\text{target}}$  := the minimum value of network error to reach when learning is considered converged
(5)  $W$  := synaptic weight between all neurons
(6)  $M$  := maximum number of iterations
(7) Initialization
(8)  $W \leftarrow$  Initial synaptic weight with random value  $\theta \sim U(0.2, 0.8)$ 
(9) For iteration = 1,  $M$  do
(10)   For  $A_i \in \{A\}$  do
(11)     Encode  $A_i = (a_1, a_2, a_3, a_4, a_5, a_6, c)$  to a series of spike trains  $A_i = (S_i^1, S_i^2, S_i^3, S_i^4, S_i^5, S_i^6, S_o^d)$ 
(12)     Set membrane potential of all neurons to the resting potential (set to 0)
(13)     Pass input spike trains to the network and find actual output spike  $S_o^a$ 
(14)     Calculate network error  $e_{A_i} = 1/2(S_o^d - S_o^a)^2$ 
(15)     Compute weight modifications  $\Delta W$  for all layers according to equation (2) and equation (4)
(16)     Update synaptic weight  $W \leftarrow W + \Delta W$ 
(17)   End for
(18)   Calculate summed network error  $e = \sum e_{A_i}$ 
(19) While  $e \geq e_{\text{target}}$ 

```

ALGORITHM 2: Multi-ReSuMe-based classifier.

neglected by the travelers and traffic management department, since people always regard clearance as the end of an accident, while the evacuation process afterwards plays a vital role in reasonable path planning. Thus, in this phase, several important values in the recovery phase will be predicted.

First, the predicted clean-up time, matrixes of traffic volume, and speed at the occurrence of the accident are used to estimate the queue length at the end of the clean-up phase (or the beginning of recovery phase). A CNNs-based predictor is used in order to better capture the spatial features of the congested road. Second, the analysis of the recovery phase is based on a backpropagation neural networks (BPNNs), where the estimated queue length together with the historical average traffic condition are used as input. The final output of this stage can include full-recovery time, half-recovery time, maximum accumulative queue length, and average accumulative queue length. Whenever a new accident record comes in, stage 2 will start when it receives the output clean-up time of stage 1.

**3.2.1. Queue Length Estimation at the End of Clean-Up Phase Using CNNs.** During the clean-up phase, one or several lanes of the highway will be affected which leads to the reduction of capacity. According to the specific number of block lanes and block distance, there will be varying degrees of vehicle accumulation. If the block area is relatively small or the traffic volume is low, the traffic flow itself will absorb disturbance and cause little congestion. However, if high traffic volume meets the large block area, the accumulated queue will be very long, and it will be necessary for the travelers to change the travel plan.

Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural network, which has been successfully employed to analyze visual imagery. Since the variables that influence the queue length in our model include

traffic flow, speed, and the geometry of the block area, they can all be better presented by pictures. Thus, CNNs are the best choice since they behave well in extracting spatial features from images so as to fully understand the spatial characteristics around the block area. A CNN consists of an input and an output layer, multiple convolutional layers, and optional hidden layers such as pooling layers, fully connected layers, and normalization layers. Figure 6 shows the demonstration of how these layers can be combined to build a CNN according to the requirement [1]. Convolutional layers apply a convolution operation to the input and pass the result to the next layer, so as to achieve feature extraction [46].

**(1) Traffic Environment Preprocessing.** The definitions and representations of the traffic environment are very important, as the accuracy of prediction is dependent on the effectiveness of the information received about the environment. To take advantage of the CNNs, the environment is processed as three pictures in the model: a picture of geometry showing the block area, a picture of the vehicle speed, and a picture of the traffic flow. A representation of this process is shown in Figure 7, where an accident blocks two lanes with 0.23 miles on a four-lane highway. Noticing that the red arrow shows the location of sensors (spread unevenly), and the grey dotted lines in Figure 7(a) represents how the picture is divided into grids that is long in 0.1 miles and wide in lane width. Figure 7(b) shows the trafficability of each grid, where the block area is set to 0, and other normal area is set to 1. The corresponding average flow (vehicles/hour) and average speeds (miles/hour) are shown in Figures 7(c) and 7(d). The data in each grid are always accessed from the nearest sensor and later normalized. The total length is set as 5 miles, which means  $4 \times 50$  grids. These settings ensure that the environment is accurately and sufficiently represented and also not too complex for the CNNs to understand.

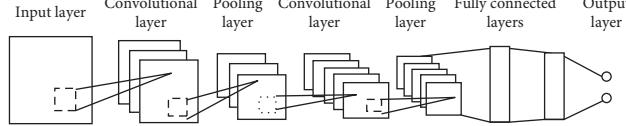
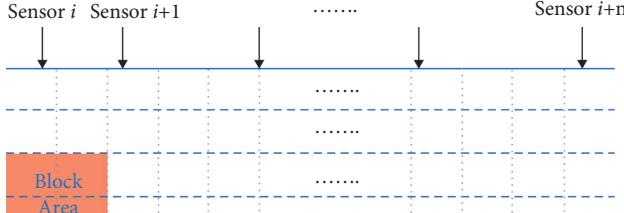


FIGURE 6: Architecture of CNNs.



(a)

134	134	109	109	99	.....	167	167	160	160
292	292	286	286	270	.....	190	190	173	173
490	490	482	482	442	.....	334	334	298	298
396	396	255	255	260	.....	288	288	277	277

(c)

1	1	1	1	1	.....	1	1	1	1
1	1	1	1	1	.....	1	1	1	1
0	0	1	1	1	.....	1	1	1	1
0	0	1	1	1	.....	1	1	1	1

(b)

75	75	76	76	75	.....	77	77	76	76
70	70	73	73	71	.....	70	70	68	68
68	68	72	72	67	.....	79	79	77	77
63	63	66	66	67	.....	66	66	66	66

(d)

FIGURE 7: Example of processing traffic environment pictures. (a) Four-lane highway map. (b) Geometry picture. (c) Traffic flow (vehicles/hour). (d) Average speed (miles/hour).

(2) *Network Structure.* For highways with different number of lanes in the dataset, we separately trained three CNNs, namely, the 4-lane model, 5-lane model, and 6-lane model. Each CNN receives three traffic environment pictures mentioned in 3.2.1.1 together with the predicted cleanup time as input, and after processing through six layers (three convolutional layers and three fully connected layers), it outputs a number representing the predicted queue length  $L_{clp}$ . The structure of the 5-lane CNN model, including the processing method in each layer and the picture size before and after each layer, is shown in Figure 8. The network structure of the 4-lane model and 6-lane model is not presented here. Learning rate is set as 0.005, and the training ends when mean square error reaches 0.001.

**3.2.2. Prediction of Spatial-Temporal Impact in Recovery Phase Using BPNNs.** At the beginning of the recovery phase, the block area will be removed and the traffic will gradually return to normal condition. Thus, the environment in this phase can be described as a simple congestion propagation model in the straight closed road. Shock wave theory [47], proposed in 1955, regards traffic flow as a continuous fluid with a linear flow-density relationship and has been applied widely in the congestion propagation analysis. However, it only concerns constant flow and density and neglects stochastic effects. This work fully considers the uneven distribution of traffic in the upstream and downstream and uses a BPNNs-based predictor to capture the nonlinear relationship between model input and output that estimate spatial-temporal impact variables such as full-recovery time,

half-recovery time, maximum accumulative queue length, and average accumulative queue length.

Similar to Section 3.2.1, we also train three BPNNs models corresponding to 4-lane case, 5-lane case, and 6-lane case. Each BPNN receives inputs including the speed and flow of each lane at the beginning of recovery phase together with the predicted queue length, and after processing through a hidden layer, it outputs a number representing the spatial-temporal impact variable. Note that a separate BPNN is trained for each variable. It is worth mentioning that in the predicting process shown in Figure 3, the instant traffic data are represented by the historical average value to improve accuracy. For example, if the predicted recovery phase of an accident starts at 2021/6/20, 8:00 (Sunday), the speed and flow data will be accessed through calculating the average value at the four most recent “Sunday 8:00.” Figure 9 shows the structure of the 4-lane BPNN model, which has 17 input neurons and 9 hidden neurons. The 5-lane model has 21 input neurons and 11 hidden neurons, while the 6-lane model has 25 and 13. Learning rate is set as 0.005, and the training ends when mean square error reaches 0.001.

The process of acquiring actual recovery time and accumulative queue length is detailed in Section 4.

**3.3. Stage 3: Online Updating.** After stage 1 and stage 2, the whole lifetime of an accident has already been predicted. It is expected that most accidents will be classified accurately, and the predicted spatial-temporal variables will match the actual one within a fault tolerance. However, some of the predictions can have long time span using the information at

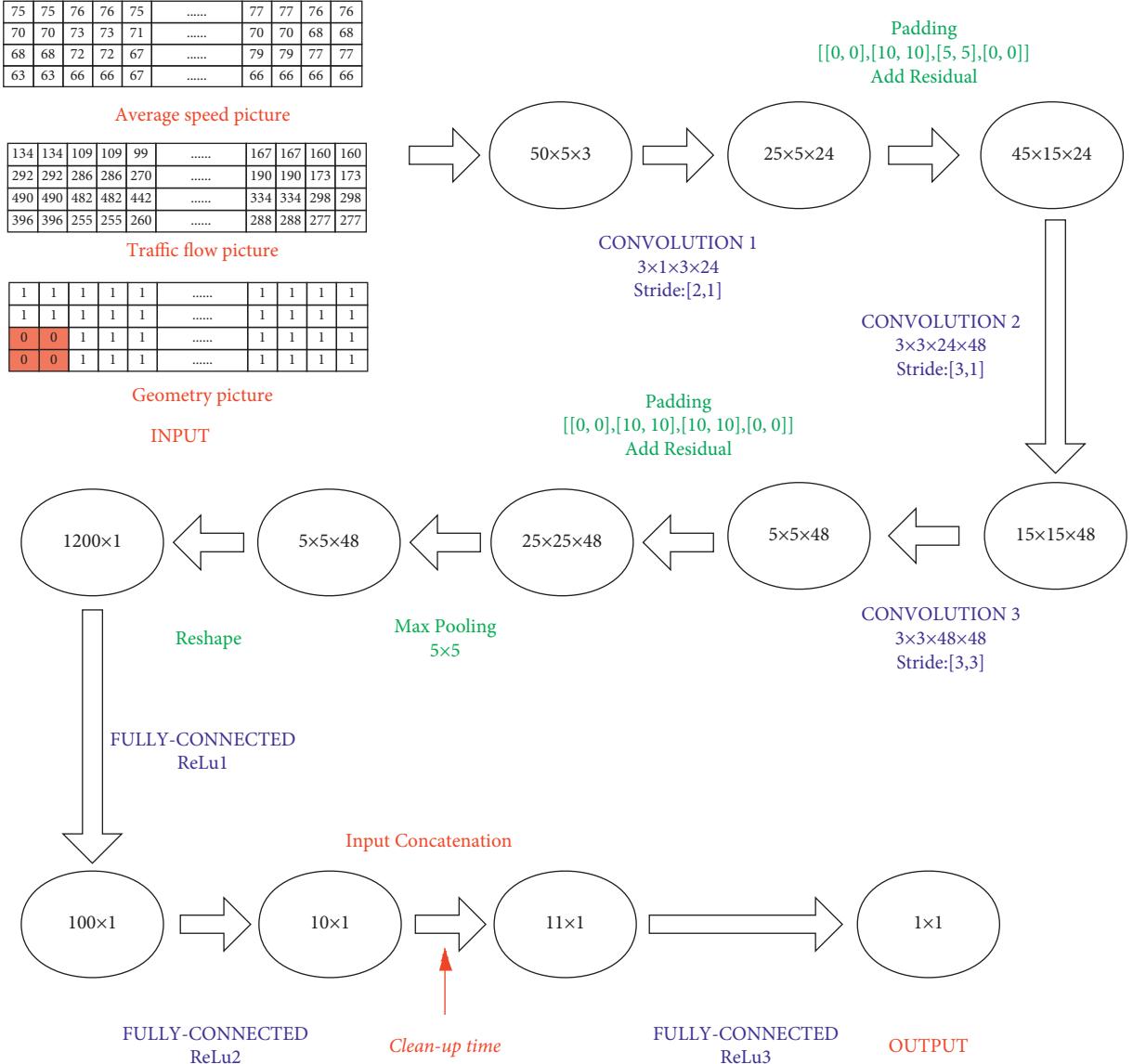


FIGURE 8: Structure of the CNNs in the 5-lane model.

the first beginning. Long-term prediction has always been a difficulty and errors must exist. The most influencing part would be the clustering and classification accuracy, since the error will continue to propagate and cause more serious errors in the following stages. That is the reason for proposing stage 3: detecting the error during the development of the accident and making timely adaptation and revision. Furthermore, the revised values can also help to modify and update the models in stage 1, namely, mean-shift-based clustering and SNNs-based classification.

**3.3.1. Error Detection and Adaptation.** The first step in the online updating stage is to detect if the predicted clean-up time is excessively deviating from the true value. As Figure 10 shows, the detection process is carried out every 5 minutes starting from the occurrence of an accident. The cause of errors can be divided into two situations:

(i)  $T_{clA} > T_{clP}$ : the accident has not been totally cleaned in the predicted clean-up time; then, the stage 2 is repeated every 5 minutes until the accident is cleaned.

(ii)  $T_{clA} < T_{clP}$ : the accident is cleaned earlier than predicted; then, the current time is set as the actual clean-up time and is used to repredict in stage 2.

This process makes up for the shortcomings of long-term prediction and helps the model to predict more accurately.

**3.3.2. Clustering and Classification Updating.** Another benefit of online stage is the modification of the clustering and prediction model when new knowledge is found along with the incoming accidents. As Figure 11 shows, the process starts with finding the actual closest cluster  $C_A$  for  $T_{clA}$ . Then, it can be divided into the following two situations:

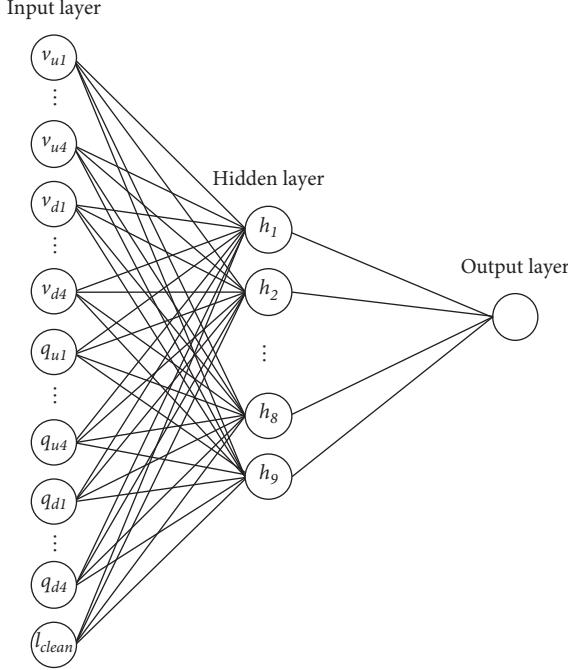


FIGURE 9: Structure of the BPNNs in the 4-lane model.

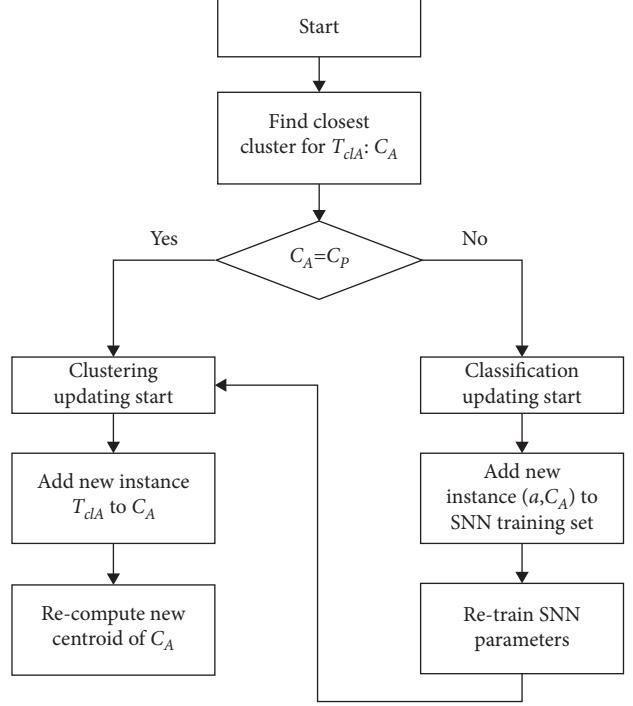


FIGURE 11: Clustering and classification updating process.

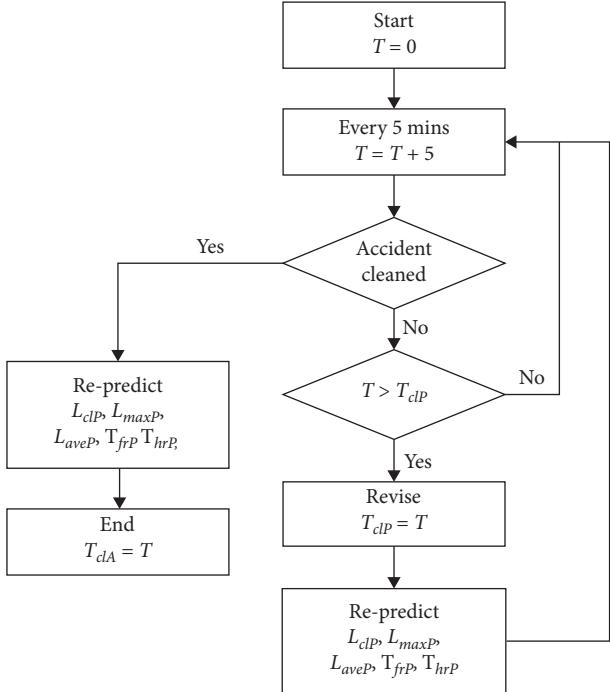


FIGURE 10: Error detection and adaptation process.

- (i)  $C_A = C_P$ : the accident is classified into the proper cluster and the error is caused by the difference from the centroid value. In this case, only a clustering updating is carried out to add a new instance and recompute new centroid of  $C_A$ ;

(ii)  $C_A \neq C_P$ : the accident is not classified into the proper cluster and the error is caused by the misclassification of the SNNs classifier. Then, another classification updating is needed where the accident attributes and its actual cluster  $C_A$  is added to the SNNs training set and later used to retrain the SNNs model.

This process helps to describe accidents that are not available in the training dataset and further improve the overall accuracy of the model.

#### 4. Dataset

This section describes the process of constructing a California statewide spatial-temporal traffic accident dataset (CASTA), using two datasets named US accident [48] and California Department of Transportation (Caltrans) Performance Measurement System (PeMS) [49].

US accident is a countrywide traffic accident dataset, which covers 49 states of the United States. The data are continuously being collected from February 2016, containing about 3.5 million accident records currently.

PeMS provides access to real-time and historical performance data in many useful formats and presentation styles, including a consolidated database of traffic data collected by Caltrans placed on state highways throughout California, as well as other Caltrans and partner agency datasets.

To figure out the spatial-temporal impact of each accident, we use the process shown in Figure 12 to match and

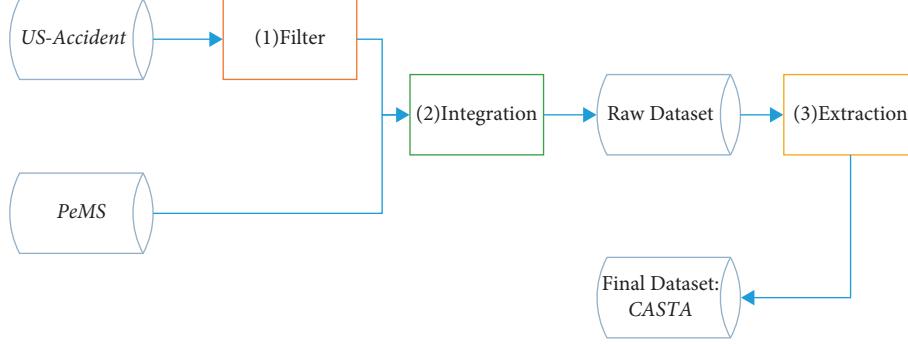


FIGURE 12: Process of creating spatial-temporal accident dataset CASTA.

merge the two datasets—US accident and PeMS. It includes three important steps: data filter, integration, and extraction.

**4.1. Data Filter.** 12,000 traffic accident records in California between February 2016 and February 2020 are filtered from the US accident datasets. The locations of the samples are all state motorways in California, and their distributions are given in Table 3. For better representing the spatial characteristics of accidents, the statistical distribution of clean-up time (sum of detection, verification, response, and clearance time) of the selected records is shown in Figure 13, from 16 min to 8 h 40 min. It can be figured out that the clean-up time is not distributed evenly in the range, where more than 96% of the records fall in (25, 120], and (25, 30] is the interval with the highest amount. However, the records with clean-up time less than 25 min or higher than 240 min are not representative and not enough to train the model. Thus, we only include 11940 pieces of records with clean-up time fallen in (25, 240] minutes in the experiment.

**4.2. Data Integration.** The data integration process is about matching the traffic flow and speed data with traffic accident records. First, we extract the latitude and longitude coordinates and time of each accident record, denoted as  $\langle \text{lat}, \text{lng}, t \rangle$ . Second, we find the nearest sensor to that coordination in PeMS and extract the flow and speed data after  $t_e$ . Third, concatenate the PeMS and US accident to form the new raw dataset. Note that disabled detectors have already been filtered out.

**4.3. Data Extraction.** By comparing with the upstream flow/speed after  $t_e$ , figure out the accident full-recovery time  $T_{\text{frA}}$ , half-recovery time  $T_{\text{hrA}}$ , maximum accumulative queue length  $L_{\text{maxA}}$ , and average accumulative queue length  $L_{\text{aveA}}$  according to Algorithm 3. The final dataset CASTA consists of 20 attributes that fall in 6 main types as given in Table 4.

## 5. Experiments and Results

Methods proposed in Section 3 have been tested with the CASTA dataset, and the results are discussed in this section. Among the 11940 pieces of records, 9940 of them are randomly chosen as the training set, while the rest 2000 are

used as the test set. Since no similar previous work that predicts the whole accident process has been proposed yet, the comparison is carried out in each stage, respectively. First, the outcomes of the first two stages without the online adaptation are presented. Second, the online stage is added to test how the outcomes have been improved. Experiments have been implemented on the server with four CPU (Intel Xeon (R) CPU E5-2650 v2 @ 2.60 GHz), 32 GB RAM.

### 5.1. Offline Prediction Analysis

**5.1.1. Clean-Up Time Analysis.** The efficiency of the initial clean-up time prediction is essential to the following stages, since the error will be further propagated and lead to larger mistakes. According to the cluster results, the clean-up time has been divided into 13 groups, where group 13 is a noise cluster. Table 5 presents the interval of each group and its corresponding centroid value.

When comparing the performance of classifiers, the classification accuracy which means the proportion that predicted class is equal to the real class is usually calculated. However, in this case, since the specific value of clean-up time is more important than the class, the classification accuracy is replaced by the MAPE and RMSE in the following equations, respectively, by comparing the predicted clean-up time  $T_{\text{clP}}$  and actual clean-up time  $T_{\text{clA}}$ .

$$\text{MAPE} = \sum_{i=1}^n \left| \frac{T_{\text{clA}} - T_{\text{clP}}}{T_{\text{clA}}} \right| \times \frac{100}{n}, \quad (5)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (T_{\text{clA}} - T_{\text{clP}})^2}. \quad (6)$$

In order to verify the suitability of our proposed model (mean-shift clustering and SNNs classifier), several methods have been selected as comparison benchmarks: nonlinear regression [50], KNN [51], ANNs, and mean-shift clustering and ANNs classifier. Noticing that the first three benchmarks directly build the relationship between accident attributes and clean-up time without the clustering process, while the fourth replaces the SNNs classifier by ANNs. The experiment results are given in Table 6. Our proposed method outperforms all other benchmarks in accuracy and

TABLE 3: Locations of traffic accident records.

County	Motorway											Total
	I-5	I-805	I-8	I-15	I-405	I-605	I-10	I-880	I-80	I-280	I-680	
San Diego	694	484	391	258	0	0	0	0	0	0	0	1827
Orange	523	0	0	0	410	25	0	0	0	0	0	958
Riverside	0	0	0	284	0	0	178	0	0	0	0	462
Los Angeles	708	0	0	0	1320	690	1741	0	0	0	0	4459
San Bernardino	0	0	0	259	0	0	863	0	0	0	0	1122
Santa Clara	0	0	0	0	0	0	0	307	0	317	216	840
Alameda	0	0	0	0	0	0	0	818	242	0	379	1439
Solano	0	0	0	0	0	0	0	0	605	0	57	662
San Francisco	0	0	0	0	0	0	0	0	137	94	0	231
Total	1925	484	391	801	1730	715	2782	1125	984	411	652	12000

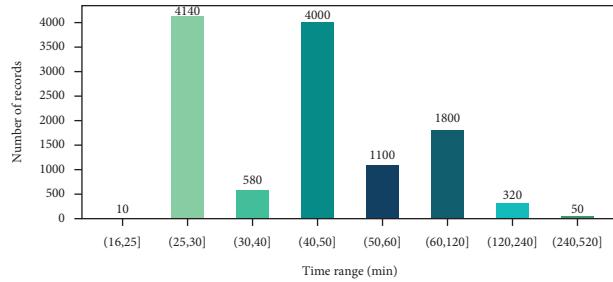


FIGURE 13: Distribution of accident clean-up time.

TABLE 4: Attributes in CASTA.

Attribute category	Attributes	Value and description
Accident attribute	Id TMC event code Block lanes Block distance	Unique identifier of the accident record 202 serious accidents 203 multivehicle accidents 245 two-lane blocked 246 three-lane blocked ..... The number of lanes affected by the accident The length of the road extent blocked by the accident
Spatial attribute	Latitude Longitude County Motorway ID Sensor ID	Latitude in GPS coordinate of the accident point Longitude in GPS coordinate of the accident point County in address field Motorway name in address field Unique identifier of the nearest sensor to the accident point
Temporal attribute	Day or night Workday or not Peak hour or not	0 day 1 night 0 workday 1 holiday or weekend 0 morning peak hour 1 evening peak hour 2 normal period
Weather attribute		0 sunny 1 rain 2 snow 3 thunderstorm 4 fog
Traffic flow characteristic attribute	Flow data series Speed data series	A matrix of flowrate of upstream sensors after accident happens A matrix of speed of upstream sensors after accident happens

TABLE 4: Continued.

Attribute category	Attributes	Value and description
Impact attribute	Accident clean-up time	Sum of detection, verification, response, and clearance time
	Full-recovery time	Time duration from clearance to the moment when no vehicle is accumulated
	Half-recovery time	Time duration from clearance to the moment when queue length is half of maximum accumulative queue length
	Maximum accumulative queue length	—
	Average accumulative queue length	—

```

(1) Input
(2)  $s_e$  := the nearest sensor in PeMS to the location of traffic accident  $e$ 
(3)  $S$  := upstream sensors set of  $s_e$  denoted by  $S = \{s_1, s_2, \dots, s_n\}$ 
(4)  $t_e$  := the closest time point in PeMS to the start time of traffic accident  $e$ 
(5)  $T$  := time point set in PeMS after  $t_e$  denoted by  $T = \{t_1, t_2, \dots, t_n\}$ , where  $t_i - t_{i-1} = 5$  min
(6)  $V_i$  := speed of sensor  $s_i$  in  $T$  denoted by  $V_i = \{v_{s_i, t_1}, v_{s_i, t_2}, \dots, v_{s_i, t_n}\}$ 
(7)  $T_c$  := sum of detection, verification, response, and clearance time of traffic accident  $e$ 
(8) Initialization
(9)  $F_{dissipate} \leftarrow \text{false}$  means if the traffic flow is in dissipating process
(10)  $v_{\text{threshold}} \leftarrow 10$  mile/h means the vehicle is low-speed driving or waiting
(11) For  $t_i$  in  $T$  do
(12)   For  $s_j$  in  $S$  do
(13)     If  $v_{s_j, t_i} > v_{\text{threshold}}$ 
(14)       Then If  $L_{\text{maxA}} < \text{distance}(s_e, s_j)$ 
(15)         Then update  $L_{\text{maxA}} = \text{distance}(s_e, s_j)$ 
(16)       Else
(17)          $F_{\text{dissipate}} = \text{true}$ 
(18)         If  $\text{distance}(s_e, s_j) < L_{\text{maxA}}/2$ 
(19)           Then  $T_{\text{hrA}} = t_i - t_e$ 
(20)         End if
(21)       End if
(22)     Update  $L_{\text{aveA}} \leftarrow ((j-1) * L_{\text{aveA}} + \text{distance}(s_e, s_j))/j$ 
(23)     Break
(24)   End for
(25)   If  $j = 1$  and  $F_{\text{dissipate}} = \text{true}$ 
(26)     Then update  $T_{\text{total}} = t_i - t_e$ 
(27)   Break
(28) End for
(29)  $T_{\text{frA}} = T_{\text{total}} - T_c$ 
(30) Output
(31)  $T_{\text{frA}}$  := full-recovery time of traffic accident  $e$ 
(32)  $T_{\text{hrA}}$  := half-recovery time of traffic accident  $e$ 
(33)  $L_{\text{maxA}}$  := maximum accumulative queue length
(34)  $L_{\text{aveA}}$  := average accumulative queue length

```

ALGORITHM 3: Data extraction from dataset.

TABLE 5: Clustering results.

Cluster no.	Clean-up time interval	Centroid value
1	[25, 30)	28.73
2	[30, 34)	30.89
3	[34, 40)	36.65
4	[40, 48)	43.02
5	[48, 53)	49.22
6	[53, 58)	56.87
7	[58, 62)	60.11

TABLE 5: Continued.

Cluster no.	Clean-up time interval	Centroid value
8	[62, 76)	68.35
9	[76, 91)	84.25
10	[91, 108)	98.37
11	[108, 116)	113.03
12	[116, 125)	118.08
13 (noise cluster)	[125, 240)	130.90

TABLE 6: Prediction performance comparison of clean-up time.

Method	MAPE (%)	RMSE	Training time (h)
Nonlinear regression	29.33	30.97	—
KNN	30.02	28.05	—
ANNs	28.58	23.64	5.50
Mean-shift clustering and ANNs classifier	14.30	11.19	4.88
Mean-shift clustering and SNNs classifier (ours)	12.07	8.13	3.20

training time. Mean-shift clustering and ANNs classifier ranks first in the comparing methods, but still needs more running time and higher MAPE and RMSE. The other three benchmarks achieves unsatisfied prediction accuracy due to the lacks of combination of clustering and classification. The results prove that clustering the records into classes and using the centroid value to present the predicted value is better than direct prediction model when the data are dispersive distributed, as well as the suitability of applying SNNs in capturing spatial and temporal features.

For better analysis of the model performance, the accuracy in each interval is calculated, respectively, as given in Table 7. It can be figured out that MAPE and RMSE are relatively low in intervals with abundant records, e.g., [25, 30) and [40, 50). In contrast, the prediction difficulty of intervals that clustered as noise (e.g., [120, 240)) is relatively high and the accuracy is not satisfied due to insufficient training. Fortunately, the probability of such severe accidents is low enough and will not cause frequent impact on traffic. Furthermore, with the increase of historic accident data, those outliers can be removed from the noise cluster and the accuracy can be further improved.

**5.1.2. Clean-Up Phase Queue Length Analysis.** In this section, the performance of CNNs based queue length prediction is analyzed, using the output of the previous stage. Comparison benchmarks include the shockwave theory [18], nonlinear regression, and ANNs. Shockwave theory assumes that the traffic flow and speed is constant at the accident site and upstream. It takes traffic flow as a continuous fluid with a flow ( $q$ )-density ( $k$ ) relationship and uses equation (7) to calculate the shockwave when the state of a traffic stream changes from  $(q_i, k_i)$  to  $(q_j, k_j)$  [18]. The length of block area is neglected in the shockwave theory.

$$W_{ij} = \frac{q_j - q_i}{k_j - k_i}. \quad (7)$$

Due to inability of capturing picture input, other two benchmarks set up relationships between constant flow, speed, block length, and queue length.

In Table 8, MAPE and RMSE of the proposed model and other benchmarks are listed based on number of lanes. 4-lane cases, 5-lane cases, and 6-lane cases account for 19%, 27%, and 54%, respectively. The proposed CNNs based model still outperforms other benchmarks in MAPE and RMSE in all three cases. Shockwave theory is unable to make efficient prediction due to unrealistic assumption, while nonlinear regression and ANNs make similar performance but are still not satisfied due to constant traffic information. Considered together with the distribution of queue length shown in Figure 14, the 4-lane model and 5-lane model have simple environment that leads to high prediction accuracy. With the increase of lanes, the traffic itself can absorb the disturbance better; thus, the accumulated queue length becomes shorter. That explains the high MAPE but low RMSE in 6-lane case.

The MAPE of 6-lane cases grouped according to the length of the queue is also calculated and shown in Figure 15. It can be concluded that the MAPE of queue length in [0.4, 2.2] miles is below 20%, while the model performs relatively poor when the queue length is below 2 miles and above 2.2 miles. The cause of error in low queue length is the quality of the PeMS dataset, where the distance between sensors is not small enough to give accurate queue length. For example, if the actual queue length of an accident is 0.06 miles while the distance between the closest sensors is 0.1 miles, then the queue length will be regarded as 0. That kind of errors will greatly affect the model performance, especially in low queue length cases. Meanwhile, when the accumulative queue length is greater than 2.2 miles, the model needs larger input to capture the environment, and thus, 5-mile is not enough to make accurate prediction.

Though the proposed model has already achieved better performance than other methods, the accuracy would still improve if trained by a more complete dataset. The merging

TABLE 7: MAPE and RMSE in clean-up time intervals.

Clean-up time interval (min)	MAPE (%)	RMSE
[25, 30)	3.62	2.64
[30, 40)	9.06	4.32
[40, 50)	5.30	3.08
[50, 60)	10.87	6.37
[60, 120)	9.01	18.26
[120, 240)	43.90	50.20

TABLE 8: Prediction performance comparison of queue length.

Method	4-lane (19%)		5-lane (27%)		6-lane (54%)		Average	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
Shockwave theory	30.03	0.17	28.98	0.15	36.79	0.15	33.40	0.15
Nonlinear regression	22.30	0.13	21.07	0.13	27.09	0.12	24.55	0.12
ANNs	22.09	0.12	23.50	0.10	26.85	0.08	25.04	0.09
CNNs (ours)	16.81	0.07	15.06	0.06	20.77	0.05	18.48	0.06

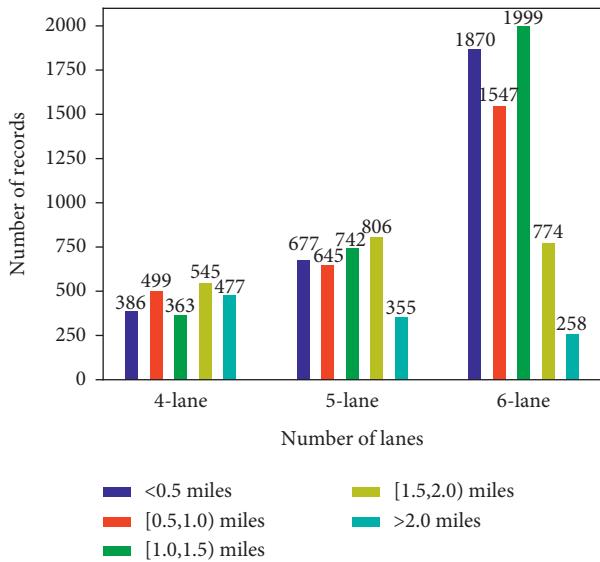


FIGURE 14: Distribution of clean-up phase queue length.

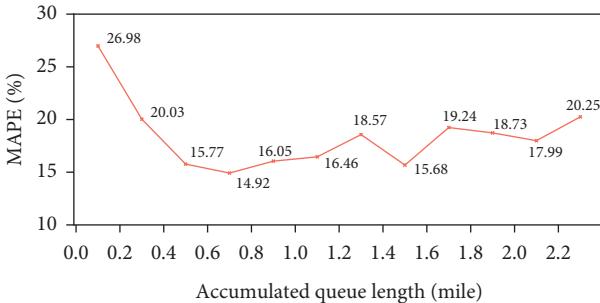


FIGURE 15: MAPE and RMSE in queue length intervals of 6-lane cases. The cross-point represents the average value in that interval.

process of US accident and PeMS can expand the error in some situation. For example, the coordination bias errors in US accident will result in an offset when matching the closest sensor in PeMS.

**5.1.3. Recovery Phase Spatial-Temporal Impact Analysis.** In the last stage of the offline prediction, the model will give out the final predicted spatial-temporal impact during the recovery process. Comparison benchmarks include shockwave theory, nonlinear regression, and BPNNs (base). The proposed model will use the output data of the CNNs model in the previous stage, while shockwave theory and nonlinear regression will still be based on the result using the same method, which means the performance analysis already includes the error of previous stages. Comparing to our proposed model that uses historical average flow/speed as input, the BPNN (base) only uses the value at the beginning of the recovery phase. According to the results given in Table 9, the error of the shockwave theory and nonlinear regression continues to expand. The MAPE of maximum accumulative queue length even reaches 40.47%, which loses its reference value. The BPNN (base) stably ranks the second under all assessment criteria, just behind our proposed method. Due to the comprehensive representation of traffic environment and the use of historical average value, the proposed model achieves satisfying performance, where the MAPE of four impact variables are all below 20%.

Although there is no existing work considering the whole accident process that can be used as comparing benchmark, the above separate comparison in each offline stage still illustrates the effectiveness of our proposed method. The offline part can be regarded as the combination of three different models developed for the characteristics in each stage, and the advantages are further expanded after the combination.

**5.2. Online Prediction Analysis.** Although the accuracy of offline prediction is already much higher than other comparing benchmarks, it is still not satisfying enough for providing precise information and guidance for governments and travelers. Especially for accidents with long time span, as shown in Figure 16, the error can be as large as 40%.

TABLE 9: Prediction performance comparison of spatial-temporal impact.

Method	4-lane (19%)		5-lane (27%)		6-lane (54%)		Average	
	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE	MAPE (%)	RMSE
<b>Full-recovery time (minute)</b>								
Shockwave theory	30.88	33.96	30.05	32.77	27.28	29.88	28.71	31.44
Nonlinear regression	29.07	31.54	27.96	31.08	26.03	26.03	27.13	28.44
BPNNs (base)	24.55	26.05	25.37	25.73	24.07	22.36	24.51	23.97
BPNNs (ours)	19.30	18.03	20.05	17.99	17.93	15.02	18.76	16.39
<b>Half-recovery time (min)</b>								
Shockwave theory	36.52	22.58	32.95	20.30	33.84	20.17	34.11	20.66
Nonlinear regression	27.33	18.07	27.74	17.94	26.32	17.72	26.90	17.85
BPNNs (base)	21.80	16.33	22.93	15.02	21.01	13.22	21.68	14.30
BPNNs (ours)	18.71	11.26	17.97	10.38	18.02	9.96	18.14	10.32
<b>Maximum accumulative queue length (mile)</b>								
Shockwave theory	38.74	0.52	38.55	0.50	42.03	0.47	40.47	0.49
Nonlinear regression	29.75	0.37	28.03	0.33	32.55	0.34	30.80	0.34
BPNNs (base)	21.74	0.31	23.09	0.31	24.11	0.27	23.38	0.29
BPNNs (ours)	18.24	0.24	18.01	0.21	21.39	0.19	19.88	0.20
<b>Average accumulative queue length (mile)</b>								
Shockwave theory	35.52	0.43	36.04	0.36	40.54	0.35	38.37	0.37
Nonlinear regression	26.22	0.26	22.34	0.19	25.62	0.25	24.85	0.23
BPNNs (base)	18.41	0.20	20.03	0.16	22.57	0.19	21.09	0.18
BPNNs (ours)	16.28	0.19	17.26	0.12	20.09	0.13	18.60	0.14

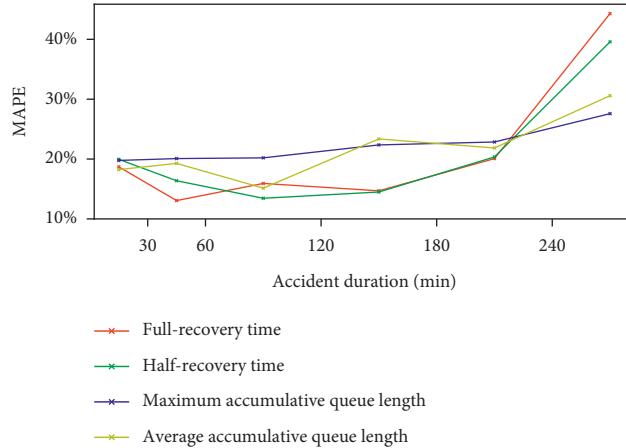


FIGURE 16: MAPE in spatial-temporal impact in accident duration intervals of 6-lane cases. The cross-point represents the average value in that interval.

To compensate for the difficulty in long-term prediction, the online prediction phase is carried out and analyzed to see how much improvement it can bring.

Taking the 6-lane cases as examples, 10 accidents in each clean-up time interval given in Table 10 are chosen as the test set (60 accidents in total) in this stage. According to statistical results given in Table 10, the actual clean-up time of 29 accidents is larger than the predicted ones, while 28 is smaller, and the rest 3 are equal. Apply error detection and adaptation mechanism to these test cases, and an averaged result per interval is given in Table 11. Reduction of MAPE is visible from any interval and impact variable when the final adaptation is done.

To further find out how the improvement is made in each test case, the detailed MAPE difference in full-recovery

time of 60 cases is shown in Figure 17, where values above 0 represent cases with less error, while negative values represent cases with more error. It can be concluded that most of the cases except for one accident (in red) experienced lower MAPE, and the differences range from -0.28% to 32.98%. Along with the increase of accident clean-up time, the improvement becomes more evident. That is reasonable since cases with longer clean-up time always experience more times of adaptation.

In Figure 18, 6 accidents are randomly picked to show how MAPE of full-recovery time changes along with the increase in adaptation times, where the adaptation interval is 5 minutes. The overall trend for all cases is gradually decreasing, where the least of them has experienced one adaptation, while the most has experienced 8 times of

TABLE 10: Clean-up time differences distribution in intervals.

Clean-up time interval	[25, 60)	[60, 90)	[90, 120)	[120, 150)	[150, 180)	[180, 240)	Total
$T_{\text{clA}} > T_{\text{clP}}$	5	4	6	5	2	7	29
$T_{\text{clA}} = T_{\text{clP}}$	2	0	1	0	0	0	3
$T_{\text{clA}} < T_{\text{clP}}$	3	6	3	5	8	3	28

TABLE 11: Average MAPE in clean-up time intervals before and after adaptation.

Clean-up time interval	[25, 60) (%)	[60, 90) (%)	[90, 120) (%)	[120, 150) (%)	[150, 180) (%)	[180, 240) (%)	Average (%)
$T_{\text{fr}}$	Offline	18.40	13.36	15.62	14.39	21.07	44.80
	Online	16.33	10.52	10.07	12.35	17.09	24.37
$T_{\text{hr}}$	Offline	20.95	16.07	12.45	14.89	20.05	38.58
	Online	17.86	15.01	10.99	11.37	17.29	20.55
$L_{\text{max}}$	Offline	19.37	20.28	20.80	22.36	22.55	27.64
	Online	17.70	17.25	16.83	18.77	17.05	18.42
$L_{\text{ave}}$	Offline	18.14	19.47	16.14	23.76	21.65	30.48
	Online	15.17	17.93	14.82	19.90	18.95	21.07

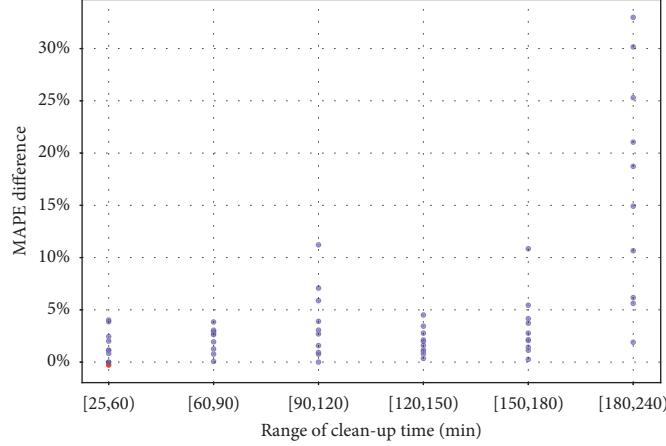


FIGURE 17: MAPE difference distribution of full-recovery time.

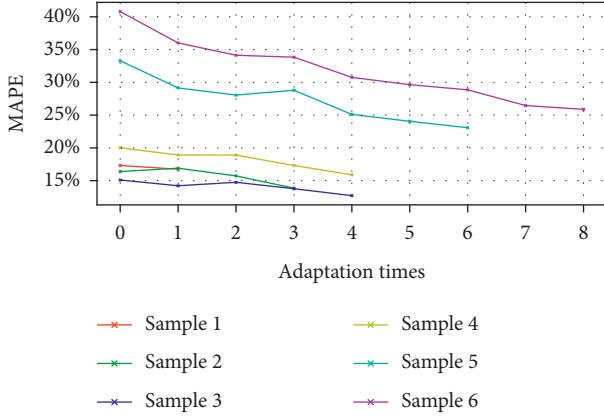


FIGURE 18: MAPE of full-recovery time along with the increase in adaptation times.

adaptation. Even after a few times of adaptation, the model can achieve a much better accuracy, which conveys the effectiveness of this mechanism.

In the second step of online stage, the clustering and prediction model will be modified and updated to improve the future prediction. To reduce workload, this process should be carried out until a batch of data has been gathered. Due to the dataset size limit, there is no enough cases to support to show the obvious change in model parameters and improvements after retraining. Thus, the result analysis of this step is not represented in this work, but can be supplemented when applied to more datasets in the future.

All of the results and discussion above point to the same conclusion: the introduction of online stage can greatly improve the model performance.

## 6. Conclusions

Traffic accidents post-impact prediction (TAPIP) plays an important role in traffic management. In this study, a 3-stage TAPIP model on highway is proposed and tested by the dataset CASTA. Using the advantages of SNNs, the model builds a comprehensive relationship between the spatial-temporal features of accidents and clean-up time. The

combination of clustering and classification improves the prediction accuracy. Due to the capability of CNNs to fully capture the spatial characteristics of traffic environment and the use of historical average data, the model achieves effective prediction of post-impact variables (e.g., fully-recovery time, half-recovery time, maximum accumulative queue length, and average accumulative queue length). A new dataset CASTA that describes California statewide spatial-temporal traffic accidents is constructed and used to test the model. Experiment results prove that our model performs better in accuracy and running time than several existing benchmarks, including nonlinear regression and KNN. The introduction of online adaptation and updating the mechanism further expands the advantages of the model.

This work implements a framework that describes the whole process of a traffic accident and captures the spatial-temporal characteristics in each phase. When applied in real world, whenever an accident takes place, the whole process can be predicted immediately. It proves the reliability and efficiency in applying SNNs in traffic domain, especially in accident prediction research. With regards to future work, we acknowledge that this work is not perfect and that there are still many aspects that can be improved upon. First, the dataset with more detailed accident information and higher quality traffic data can be used to improve the model performance. Second, accidents with clean-up time longer than 240 or shorter than 25 minutes can be further added into the model when enough records have been gathered. Third, clustering and classification updating in online stage needs further experimental demonstration. Last, the model can be extended to city roads rather than limited to highways.

## Data Availability

All data and program files included in this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported in part by the NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1709212) and “Research on frontiers of intelligent transport system” funded by China Association for Science and Technology, National Natural Science Foundation of China (U1509205).

## References

- [1] D. Li, J. Wu, M. Xu, Z. Wang, and K. Hu, “Adaptive traffic signal control model on intersections based on deep reinforcement learning,” *Journal of Advanced Transportation*, vol. 2020, Article ID 6505893, 14 pages, 2020.
- [2] T. Beshah and S. Hill, “Mining road traffic accident data to improve safety: role of road-related factors on accident severity in Ethiopia,” in *Proceedings of the 2010 AAAI Spring Symposium Series*, Stanford, CA, USA, March 2010, <https://www.aaai.org/ocs/index.php/SSS/SSS10/paper/view/1173>.
- [3] J. Tang, L. Zheng, C. Han et al., “Statistical and machine-learning methods for clearance time prediction of road incidents: a methodology review,” *Analytic Methods in Accident Research*, vol. 27, Article ID 100123, 2020.
- [4] K. A. Small, “Economics and urban transportation policy in the United States,” *Regional Science and Urban Economics*, vol. 27, no. 6, pp. 671–691, 1997.
- [5] C. Zhan, A. Gan, and M. Hadi, “Prediction of lane clearance time of freeway incidents using the M5P tree algorithm,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1549–1557, 2011.
- [6] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.
- [7] A. Tavanaei, M. Ghodrati, S. R. Kheradpisheh, T. Masquelier, and A. Maida, “Deep learning in spiking neural networks,” *Neural Networks*, vol. 111, pp. 47–63, 2019.
- [8] F. Su, H. Dong, L. Jia, Y. Qin, and Z. Tian, “Long-term forecasting oriented to urban expressway traffic situation,” *Advances in Mechanical Engineering*, vol. 8, no. 1, 2016.
- [9] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, “Short-term traffic prediction based on dynamic tensor completion,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2123–2133, 2016.
- [10] L. Zhihui, C. Qian, Z. Yonghua, and Z. Rui, “Signal cooperative control with traffic supply and demand on a single intersection,” *IEEE Access*, vol. 6, pp. 54407–54416, 2018.
- [11] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, “A data-driven lane-changing model based on deep learning,” *Transportation Research Part C: Emerging Technologies*, vol. 106, pp. 41–60, 2019.
- [12] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers in Computational Neuroscience*, vol. 9, 2015.
- [13] Q. Yu, H. Tang, K. C. Tan, and H. Yu, “A brain-inspired spiking neural network model with temporal encoding and learning,” *Neurocomputing*, vol. 138, pp. 3–13, 2014.
- [14] A. Tavanaei and A. Maida, “Bio-inspired multi-layer spiking neural network extracts discriminative features from speech signals,” in *Neural Information Processing*, pp. 899–908, Springer, Cham, Switzerland, 2017.
- [15] N. Kasabov, V. Feigin, Z.-G. Hou et al., “Evolving spiking neural networks for personalised modelling, classification and prediction of spatio-temporal patterns with a case study on stroke,” *Neurocomputing*, vol. 134, pp. 269–279, 2014.
- [16] I. Laña, E. Capecci, J. Del Ser, J. L. Lobo, and N. Kasabov, “Road traffic forecasting using NeuCube and dynamic evolving spiking neural networks,” *Intelligent Distributed Computing XII*, Springer, Cham, Switzerland, pp. 192–203, 2018.
- [17] I. Laña, J. L. Lobo, E. Capecci, J. Del Ser, and N. Kasabov, “Adaptive long-term traffic state estimation with evolving spiking neural networks,” *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 126–144, 2019.
- [18] G. Yu, Y. Liu, Y. Wang, and X. Zhu, “Analysis of traffic accident temporal and spatial impact based on shockwave theory,” in *Proceedings of the 12th International Conference of Transportation Professionals (CICTP 2012)*, pp. 557–567, Beijing, China, August 2012.
- [19] S. Chandana Wirasinghe, “Determination of traffic delays from shock-wave analysis,” *Transportation Research*, vol. 12, no. 5, pp. 343–348, 1978.

- [20] Y. He, Z. Liu, and S. Du, "Prediction of recovery time of urban traffic accident based on active flow-split," in *Proceedings of the 2016 International Conference on Smart City and Systems Engineering (ICSCSE)*, pp. 7–9, Hunan, China, November 2016.
- [21] E. C. Sullivan, "New model for predicting freeway incidents and incident delays," *Journal of Transportation Engineering*, vol. 123, no. 4, pp. 267–275, 1997.
- [22] S. Cohen and C. Nouveliere, "Modelling incident duration on an urban expressway," *IFAC Proceedings Volumes*, vol. 30, no. 8, pp. 297–301, 1997.
- [23] B. Zhu, X. Fu, and S. Yang, "Prediction model of space-time impact of traffic accidents based on nonlinear regression and BP neural network," *Highway Engineer*, vol. 43, no. 6, pp. 134–139, 2019.
- [24] Y. Lin and R. Li, "Real-time traffic accidents post-impact prediction: based on crowdsourcing data," *Accident Analysis & Prevention*, vol. 145, Article ID 105696, 2020.
- [25] Y. Lee, C.-H. Wei, and K.-C. Chao, "Evaluating the effects of highway traffic accidents in the development of a vehicle accident queue length estimation model," *International Journal of Intelligent Transportation Systems Research*, vol. 16, no. 1, pp. 26–38, 2018.
- [26] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 8, pp. 790–799, 1995.
- [27] C.-H. Wei and Y. Lee, "Sequential forecast of incident duration using artificial neural network models," *Accident Analysis & Prevention*, vol. 39, no. 5, pp. 944–954, 2007.
- [28] L. Kuang, H. Yan, Y. Zhu, S. Tu, and X. Fan, "Predicting duration of traffic accidents based on cost-sensitive Bayesian network and weighted K-nearest neighbor," *Journal of Intelligent Transportation Systems*, vol. 23, no. 2, pp. 161–174, 2019.
- [29] B. Yu, Y. T. Wang, J. B. Yao, and J. Y. Wang, "A comparison of the performance of ANN and SVM for the prediction of traffic accident duration," *Neural Network World*, vol. 26, no. 3, pp. 271–287, 2016.
- [30] "TMC/Event Code List-OpenStreetMap Wiki," 2021, [https://wiki.openstreetmap.org/wiki/TMC/Event\\_Code\\_List](https://wiki.openstreetmap.org/wiki/TMC/Event_Code_List).
- [31] V. Punzo, M. T. Borzacchiello, and B. Ciuffo, "On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1243–1262, 2011.
- [32] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *International Journal of Neural Systems*, vol. 19, no. 4, pp. 295–308, 2009.
- [33] X. Lin, X. Wang, and Z. Hao, "Supervised learning in multilayer spiking neural networks with inner products of spike trains," *Neurocomputing*, vol. 237, pp. 59–70, 2017.
- [34] J. Feng, "Is the integrate-and-fire model good enough?—A review," *Neural Networks*, vol. 14, no. 6-7, pp. 955–975, 2001.
- [35] C. F. Stevens and A. Zador, "Neural coding: the enigma of the brain," *Current Biology*, vol. 5, no. 12, pp. 1370–1371, 1995.
- [36] G. D. Lewen, W. Bialek, and R. R. d. R. v. Steveninck, "Neural coding of naturalistic motion stimuli," *Network: Computation in Neural Systems*, vol. 12, no. 3, pp. 317–329, 2001.
- [37] G. Buzsaki, R. Llinas, W. Singer, A. Berthoz, and Y. Christen, *Temporal Coding in the Brain*, Springer Science & Business Media, Berlin, Germany, 2012.
- [38] D. A. Butts, C. Weng, J. Jin et al., "Temporal precision in the neural code and the timescales of natural vision," *Nature*, vol. 449, no. 7158, pp. 92–95, 2007.
- [39] A. E. Orhan and W. J. Ma, "Neural population coding of multiple stimuli," *Journal of Neuroscience*, vol. 35, no. 9, pp. 3825–3841, 2015.
- [40] S. M. Bohte, J. N. Kok, and H. La Poutré, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1, pp. 17–37, 2002.
- [41] A. Kasiński and F. Ponulak, "Comparison of supervised learning methods for spike time coding in spiking neural networks," *International Journal of Applied Mathematics and Computer Science*, vol. 16, pp. 101–113, 2005.
- [42] N. Caporale and Y. Dan, "Spike timing-dependent plasticity: a Hebbian learning rule," *Annual Review of Neuroscience*, vol. 31, no. 1, pp. 25–46, 2008.
- [43] I. Sporea and A. Grüning, "Supervised learning in multilayer spiking neural networks," *Neural Computation*, vol. 25, no. 2, pp. 473–509, 2013.
- [44] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, UK, 2002.
- [45] I. Sporea and A. Grüning, "Classification of distorted patterns by feed-forward spiking neural networks," in *Artificial Neural Networks and Machine Learning-ICANN 2012*, pp. 264–271, Springer, Berlin, Germany, 2012.
- [46] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: integrated recognition, localization and detection using convolutional networks," 2013, <https://arxiv.org/abs/1312.6229>.
- [47] M. J. Lighthill and G. B. Whitham, "On kinematic waves II. A theory of traffic flow on long crowded roads," *Proceedings of the Royal Society of London-Series A: Mathematical and Physical Sciences*, vol. 229, no. 1178, pp. 317–345, 1955.
- [48] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, "Accident risk prediction based on heterogeneous sparse data," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 33–42, New York, NY, USA, November 2019.
- [49] Caltrans PeMS," 2021, [https://pems.dot.ca.gov/?controller\\_id=714907&content=detector\\_health&tab=&s\\_time\\_id=1625443200&dnode=VDS&station\\_id=717033&st\\_cd=on&st\\_ch=on&st\\_ff=on&st\\_hv=on&st\\_ml=on&st\\_fr=on&st\\_or=on](https://pems.dot.ca.gov/?controller_id=714907&content=detector_health&tab=&s_time_id=1625443200&dnode=VDS&station_id=717033&st_cd=on&st_ch=on&st_ff=on&st_hv=on&st_ml=on&st_fr=on&st_or=on).
- [50] T. Amemiya, "Chapter 6 non-linear regression models," in *Handbook of Econometrics*, vol. 1, pp. 333–389, Elsevier, Amsterdam, Netherlands, 1983.
- [51] J. H. Friedman, F. Baskett, and L. J. Shustek, "An algorithm for finding nearest neighbors," *IEEE Transactions on Computers*, vol. 24, no. 10, pp. 1000–1006, 1975.