

# Urban Traffic Intersection Incident Prediction Using AI Algorithm

Yaguang Kong<sup>1</sup> Huakui Chen<sup>2</sup>

<sup>1</sup>Institute of Information and Control, Hangzhou Dianzi University, 310037, China

Email : ygkong@263.net

<sup>2</sup>College of Medicine, Zhejiang University, 310027, China

## Abstract

Automatic incident detection and characterization is urgently require in the development of advanced technologies used for reducing non-recurrent traffic congestion on urban traffic. This paper presents a new method using data mining to identify automatically freeway incidents. As a component of a real-time traffic adaptive control system for signal control, the algorithm feeds an incident report to the system's optimization manager, which uses the information to determine the appropriate signal control strategy. Off-line tests were conducted to substantiate the performance of the proposed incident detection algorithm based on simulated data. The test results indicate the feasibility of achieving real-time incident detection utilizing the proposed method.

**Keywords:** Incident Detection, Neural Network, Fuzzy Logic

## 1. Introduction

Traffic congestion is a growing problem in urban areas around the world. Much of this congestion is caused by "incidents" events such as accidents or load spills which temporarily reduce highway capacity. Many jurisdictions have implemented incident management systems to reduce the impacts of such events and maintain highway capacity as close as possible to normal levels. Automated Incident Detection(AID). Systems help manage nonrecurring congestion on either a stand alone basis or in combination with Intelligent Transportation Systems(ITS).AID Systems divide the time horizon into multiple intervals or steps, and in each time step they classify samples of link flow measurements into varied traffic states. Most AID Systems classify into two traffic states-the incident free and incident states. The current measures of effectiveness of AID Systems consist of the detection rate(DR),the false alarm rate(FAR),and the mean time to detect(MTTD).The DR is measured as the ratio of detected incidents to total number of incidents present in a data stream. The

FAR represents the ratio of wrongly identified incidents to total number of intervals tested(given an incident-free data stream) or the ratio of unidentified incidents to total number of incidents present in a data stream. Finally, the MTTD represents the mean difference in incident detection and start-up times.

In the past decades, numerous AID algorithm have been proposed. These AID algorithms can be classified into the following categories[4]:

**Comparison algorithms:** Comparison algorithms compare the current traffic conditions such as volume and occupancy to preset thresholds and decide whether or not an incident has occurred. Examples of the comparison algorithms include the California algorithm, TSC algorithm 7, TSC algorithm 8, all purpose incident detection algorithm and pattern recognition algorithms.

**Statistical algorithms:** Statistical algorithms compare the observed traffic data with the predicated traffic data and decide whether or not an accident has occurred based on statistical significance. Examples of the statistical algorithms include the standard normal deviation algorithm and the Bayesian algorithm.

**Time-series algorithms:** Time-series algorithms compute the short-term forecasting values based on observed values and statistical forecasting of traffic data. Significant deviations between observed and forecast values are attributed to incidents. Examples of the time-series algorithms include the high occupancy algorithm, the double exponential smoothing algorithm, the detector logic with smoothing algorithm and the autoregressive integrated moving average algorithm.

**Catastrophe theory algorithms:** Catastrophe theory algorithm such as the McMaster algorithm detect incidents from sudden discrete changes that occur in a single variable of interest while other related variables such as speed, flow, and occupancy are exhibiting a smooth and continuous change.

**Other algorithms:** These algorithms include neural network algorithms, fuzzy set algorithms and video image processing.

This paper presents the development of the neural network and fuzzy logic system for incident detection.

The main objectives for this automated incident detection algorithm are as follows:

- ① Determine incident occurrence using different traffic indices.
- ② Have a high incident-detection rate;
- ③ Have a low FAR;
- ④ Have a low average mean TTD after an actual incident occurrence.

## 2. Prediction of Traffic Information Using GNN

Uncertainty inherent to the distribution of traffic data complicates the traffic parameters forecasting. Stochastic (random) fluctuations in traffic data and unexpected abnormal fluctuations (i.e., if congestion and incidence occurring) significantly influence the future traffic information. Herein, G-RNN models are developed for predicting short-term traffic parameters (volume, speed and occupancy). In grey modeling, assumptions regarding statistical distribution of data are unnecessary. The accumulated generating operation (AGO) is an important feature of grey models, which focuses largely on reducing the randomness of data.

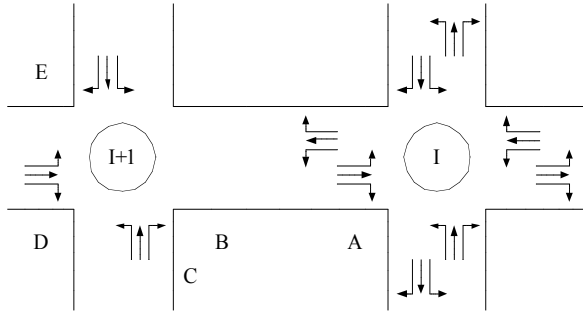


Fig.1: Typical Intersection

Let  $x_d^{(0)}(k)$  be the original traffic information of I lane J segment K time stage.

Let:

$$x_d^{(0)} = [x_d^{(0)}(1), x_d^{(0)}(2), \dots, x_d^{(0)}(n)]$$

$$x_d^{(1)} = [x_d^{(1)}(1), x_d^{(1)}(2), \dots, x_d^{(1)}(n)]$$

Here:

$$x_d^{(1)}(k) = \sum_{l=1}^k x_d^{(0)}(l)$$

According to the GM(1,1) model, has

$$\frac{dx_d^{(1)}}{dk} + ax_d^{(1)} = u$$

We can get the estimation of  $x_d^{(0)}(k)$ :

$$x_d^{(0)}(k) = \left( x_d^{(0)}(1) - \frac{u}{\hat{a}} \right) \left( 1 - e^{\hat{a}} \right) e^{-\hat{a}(k-1)}, k = 2, 3, \dots \quad (1)$$

The parameters  $a, u$  can be estimated by LMS algorithm.

To improve traffic information prediction more precisely, this study introduces the recurrent neural networks and further combined RNN with grey modeling. The recurrent neural network model [7] is a dynamic network, in which it has an internal feedback. Elman [7] demonstrates the capability of the RNN to learn complex spatial-temporal patterns. RNN is a two-layer back propagation network, with the addition of a feedback connection from the output of the hidden layer to its input. This feedback path allows RNN to learn to recognize and generate temporal patterns, as well as spatial patterns. The RNN learns to interpret current inputs in the context of its previous internal states. The RNN consists of two trainable layers of neurons (hidden and output layers), and two layers which simply distribute their activations to the hidden layer (input and context layers). In RNN, the input layer units at time k receive as input not only the input vectors for time k but also hidden layer output of time k-1. The new inputs, i.e., the feedback hidden layer outputs, are called context vectors. The structure of G-RNN for traffic information prediction is depicted in Fig. 2. In Fig. 2, the G-RNN algorithm is constructed by connecting with grey-AGO in the input layer and grey-AGO in the output layer of the RNN architecture. In order to develop the neural network-based forecasting models, the neural networks need to be trained with historical examples of input-output data. The grey-AGOs of traffic volume, speed, and occupancy measurements from the current time point at a given detector station form the input to the G-RNN model. Then output of G-RNN model comprises the traffic volume, speed, and occupancy measurements at the same station at some future time point. Herein,  $x_d^{(1)}(k) \in X$  and  $z_j(k) \in Z$ , respectively, represent the input layer vector and

hidden layer vector, and  $x_d^{(1)}(n)$  is the output unit at time n. Moreover, to compute the hidden node outputs, considering the activation function for  $i^{\text{th}}$  hidden node output at time k can be defined as:

$$z_j(k) = f_H(\gamma_j x_d^{(1)}(k) + \sum_j \delta_j z_j(k-1)), k = 1, 2, \dots, n \quad (2)$$

where  $z_j(k)$  is a output of hidden node i at time k,  $f_H(\cdot)$  is the hidden neuron function for traffic forecasting,  $\gamma_j$  is weight connecting the input layer unit to hidden unit j,  $\delta_j$  is the weight connecting

context layer units to hidden unit  $j$ , and context layer input at time  $k$  then is given as  $z_j(k-1)$ . Then, the output functions of the RNN can be presented as follow:

$$\hat{x}_d^{(1)}(n) = f_o(\beta_0 + \sum_j \beta_j z_j(n)), \quad (3)$$

$$z_j(n) = f_H(\gamma_j x_d^{(1)}(n) + \sum_i \delta_i z_i(n-1)),$$

where  $f_o(\cdot)$  is the output neuron function for traffic information prediction,  $\beta_0$  is a bias unit for the output layer,  $\beta_j$  is the weight connecting the hidden unit  $j$  to the output unit. For computation of  $\delta$ , the derivative chain rule is formulated, then used to update the weight coefficients between context layer and the hidden layer of the training procedure of RNN. The learning algorithm in the RNN model is the same as that in the back propagation networks, using the gradient descent rule, which adjusts the weights based on the derivatives of the error with respect to the weights. Final, the grey-AGO post processes the

output of RNN  $\hat{x}_d^{(1)}(n)$ ; then obtain the prediction

$\hat{x}_d^{(0)}(n)$  at the future time point  $n$ . These output forecasts are then compared with the historical traffic volume and speed data. The entire input sequence is presented to the G-RNN network, and its outputs are calculated and compare with the target sequence to generate an error sequence. Mean squared error (MSE) is used as a performance function in training the G-RNN model. For each time epoch, the error is back propagated to find gradients of errors for each weight and bias. The gradient is actually an approximation since the contributions of weights and biases to errors via the delayed recurrent connection are ignored. The gradient is then used to update the weights with the back propagated training function.

### 3. Incident Prediction using Fuzzy Logic

Fuzzy logic is a human-inference-oriented AI technique that incorporates the uncertainty and abstract nature inherent in human decision-making into intelligent control systems. It captures the approximate and qualitative boundary conditions of system variables by fuzzy sets with membership functions. A fuzzy system implements functions in near-human terms, i.e. IF-THEN linguistic rules, with reasoning by fuzzy logic.

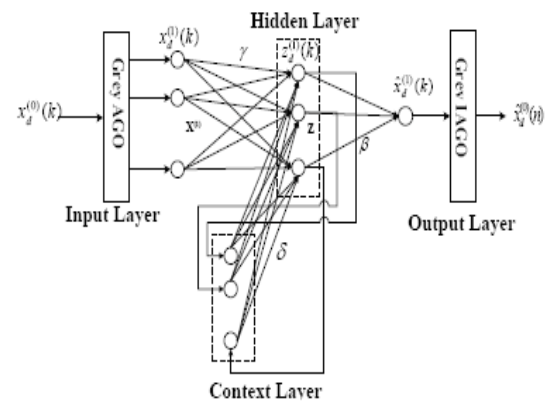


Fig.2. Model structure of G-RNN for traffic parameter Prediction

The fuzzyfication of a crisp set value to a fuzzy terminology is characterized by a scaling factor and a quantization process. After quantization, a degree-of-membership function is applied to derive its membership value, or “belongingness”, in each of the fuzzy linguistic sets. Typically, membership functions are determined by using the average ( $\mu$ ) and standard deviation values ( $\sigma$ ) of observed data. In Fig. 3, average value is assumed the core value of the  $K$  membership function and the boundary of this membership function is determined adding and subtracting of standard deviation value to the core. Other membership functions of the parameter are also established adding the core value of the  $K$  membership function to the multiples of standard deviation value.

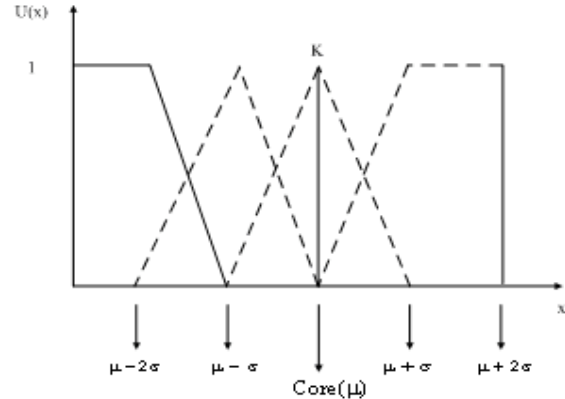


Fig.3 Determination of membership functions

After the input variables have been fuzzyfied, they are fed into a two-dimensional fuzzy decision table to derive the output variable. Using the look-up table, fuzzy control actions (incident state) are computed.

A fuzzy inference routine serves as the main engine of the incident detection algorithm. For each traffic measure, the routine evaluates the past five-minute average and the percent change from the past five-minute average during the most recent minute to determine whether an abrupt change has occurred that constitutes abnormal traffic conditions indicative of the occurrence (or clearance) of an incident. [1]

The proposed approach of this work uses the following variables: speed and occupancy. Use the prediction value from GNN to get the state of incident. For each one of the entrance variables (linguistic variables) is proposed a fuzzy partition with four linguistic values. For example: 1) Average Occupancy: Overall, Heavy, Medium, Light; 2) Percent Increase in Occupancy: Very Large, Large, Medium, Small; 3) Average Volume: Very Slow, Slow, Medium, Quick; 4) Percent Increase in Volume: Very Large, Large, Medium, Small. The linguistic value of Incident State is Very High, High, Medium, Small.

Each fuzzy decision table takes in two inputs, such as if the average occupancy during the past five minutes is heavy and the percent change in volume during the most recent minute is small, then traffic conditions are incident.

The results in a  $4 \times 4$  fuzzy decision table with 16 outputs and degrees of membership. The degree of matching resulting from the inference represents the grade of normality of a location.

#### 4. Example

In order to validate the proposed approach a case study was carried out aiming to have a high incident-detection rate, to have a low FAR, to have a low average mean TTD after an actual incident occurrence. The result is showed in Table 1.

Table 1 Performance of Our Method

Performance measures	Volume cases	This paper
Detection Rate(%)	Light	73
	Medium	92
	Heavy	92
	Overall	87
False alarm rate(%)	Light	0
	Medium	0.86
	Heavy	0.65
	Overall	0.49
Mean time to detect(sec)	Light	258
	Medium	246
	Heavy	226
	Overall	232

#### 5. Conclusion

This research applied fuzzy logic and GNN to incident detection. The inputs to this incident detection algorithm are the average and percent change value of occupancy and volume. The algorithm was tested on a simulation data. The algorithm can be developed as part of a real-time, traffic adaptive, diamond interchange control system. The algorithm captures

overall incident effects on a network using multiple traffic measures for more accurate incident detection.

#### 6. References

- [1] Dipti Srinivasan, Ruey Long Cheu, Young Peng Poh, Albert Kim Chwee Ng, "Development of an intelligent technique for traffic network incident detection", Engineering Application of Artificial Intelligence, Vol.13, 2000, pp311-322.
- [2] Li-Yen Chang \*, Wen-Chieh Chen, Data mining of tree-based models to analyze freeway accident frequency, Journal of Safety Research 36 (2005) 365 – 375
- [3] Evrim Bayam, Jay Liebowitz, William Agresti, Older drivers and accidents: A meta analysis and data mining application on traffic accident data, Expert Systems with Applications 29 (2005) 598–629
- [4] Ng, K.C., Li, Y., 1994. Design of sophisticated fuzzy logic controllers using genetic Algorithms. Proceedings of 3rd IEEE International Conference on Fuzzy Systems 3, 1708-1712.
- [5] Yuh-Horng Wen Tsu-Tian Lee, Fuzzy Data Mining and Grey Recurrent Neural Network Forecasting for Traffic Information Systems, 2005, IEEE, pp356-361.
- [6] C.-K. LIN AND G.-L. CHANG, "Development of a Fuzzy-Expert System for Incident Detection and Classification", Mathl. Comput. Modelling Vol. 27, No. 9-11, pp. 9-25, 1998
- [7] J. L. Elman, "Finding Structure in Time," Cognitive Science, vol. 14, 1990, pp. 179-211.
- [8] Prasad Palacharla, Peter C. Nelson, Virginia P. Sisiopiku, "Data Fusion Using Fuzzy-Valued Logic", IEEE, pp115-119.
- [9] N.D. Belnap, "A useful four-valued logic," in Modern Uses of Multiple-Valued Logics, ed. G. Epstein and M. Dum, Boston, MA: Reidel, 1977.
- [10] Sibok Lee, Raymond A. Krammes, John Yen, "Fuzzy-logic-based incident detection for signalized diamond Interchanges", Transportation Research Part C Vol.6, pp359-377, 1998.
- [11] N.E. Thomas, "Multi-state and multi-sensor incident detection systems for arterial streets", Transportation Research Part C 6 (1998) 337-357
- [12] Jiu-Biing Sheu, "A sequential detection approach to real-time freeway incident detection and characterization", European Journal of Operational Research 157 (2004) 471–485