Artificial Neural Networks for Freeway Incident Detection

YORGOS J. STEPHANEDES AND XIAO LIU

A freeway incident detection algorithm is developed using back propagation neural networks. Based on real-time occupancy and volume counts from pairs of adjacent loop detector stations, the network is trained with actual data, including 31 incidents from a typical freeway in the Twin Cities Metropolitan Area over the afternoon peak period during 72 days. Results indicate that the neural network, with about 1,000 connections, can learn the main characteristics of a variety of incidents. Algorithm performance, in terms of detection and false alarm rates, is superior to most of the best algorithms that have been tested with this data set.

Fast and accurate detection of incidents is vital for the successful operation of incident management systems. With incidents accounting for many of the vehicle hours lost to nonrecurring freeway congestion, prompt and reliable detection (critical for assuring effective response and clearance) can substantially contribute to improving freeway traffic flow.

Low reliability is the major shortcoming of existing automated incident detection methods for freeway operations. Because of the high number of false alarms generated by such methods, traffic engineers generally do not rely on them for automated detection of incidents.

Because incident management is critical in reducing the total delay to drivers in urban freeways, traffic planners continue to develop methods that can be used to reliably identify an incident. Interest in such methods has increased as transportation officials realize that prompt and reliable detection of incidents is critical to advanced traffic management systems (*1–2*), which seek to provide optimal control of freeway and arterial networks.

Recent research has focused on assessing how existing and new incident detection systems perform (*3*). This assessment involved developing and testing a new algorithm and comparing it with existing ones. This study represents an effort to improve incident detection systems by designing an incident detection algorithm based on artificial neural networks.

BACKGROUND

Artificial neural networks, whose structures are based on the present understanding of biological nervous systems, have been studied for many years in the hope of achieving human-like performance in various practical applications (*4*). These models comprise a large number of simple, nonlinear computational elements operating within a parallel distributed information processing architecture and arranged in patterns reminiscent of a biological neural network. Computational elements or nodes are connected to other elements via weights that are typically adapted during use to improve performance. Through each connection, each element may receive information from another, weighted by the corresponding weight of that connection. Research has demonstrated that artificial neural networks offer high computation rate, memory, learning, and fault tolerance.

Neural network classifiers are nonparametric and make weaker assumptions concerning the shapes of underlying distributions than traditional statistical classifiers. They may thus prove more robust when distributions are generated by a nonlinear process and are strongly non-Gaussian. In particular, they can identify (*a*) which class best represents an input pattern, and (*b*) where it is assumed that inputs have been corrupted by noise or some other process. Several neural network models can be used as classifiers, including the Hopfield net (*5*), the Carpenter-Grossberg classifier (*6*), Kohonen's self-organizing model (*7*), and Multi-layer network (*8*). Multi-layer networks are feedforward nets with one or more hidden layers of nodes between the input and output nodes. Since the backpropagation training algorithm was proposed, Multi-layer network has become the most popular model, particularly in the pattern recognition field. The feasibility of neural network models for freeway incident detection has been demonstrated (*9*).

Incident detection is a typical pattern recognition problem and can benefit from the application of neural network methods. In particular, in incident detection human-like performance is sought in detecting unusual events in the traffic stream and in reducing false alarms by differentiating incidents from other events, such as compression waves, traffic pulses, and equipment malfunction. Methods should be robust under the assumption that the traffic distributions are generated by nonlinear, non-Gaussian processes. Although new incident detection algorithms are promising, methods that can be adapted during use are expected to improve performance. In addition, high computation rate and fault tolerance are needed, and these are characteristics of neural networks. A neural network algorithm is developed that can be trained to recognize traffic patterns through time, and classify such patterns as having an incident or being incident-free. The sensitivity of classification performance on training methods is also investigated. Finally, the performance of the neural network is compared with recent findings from well-performing methods, with encouraging results.

Brief Description of Back Propagation Algorithm

The back propagation neural network is a feedforward, multilayer perceptron with one or more layers of nodes hidden between the input and output nodes, which can be trained using the back propagation algorithm. Although it cannot be proven that this algorithm

Y. J. Stephanedes, Department of Civil and Mineral Engineering, and X. Liu, Department of Civil and Mineral Engineering and Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, Minn. 55455.

converges, its applications have been shown to be successful in a variety of problems. A three-layer network with one layer hidden is shown in Figure 1, where

$$Y_m = f\left(\sum_{k=0}^{K-1} W_{km} Z_k - \Theta_m\right), \qquad m = 0, 1, \ldots, M-1 \qquad (1)$$

$$Z_k = f\left(\sum_{i=0}^{N-1} W_{ik} X_i - \Theta_k\right), \qquad k = 0, 1, \ldots, K-1 \qquad (2)$$

and W_{km}, W_{ik} are the connections between nodes k and m, and i and k, respectively, usually called weights.

The back propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output and the desired output of a multilayer feedforward perceptron. Each node in the hidden and output layers adds weighted inputs from the previous layer and passes the result through a nonlinear function that must be continuously differentiable; the capabilities of this network stem from the use of this function. The following logistic nonlinearity, familiar to transportation engineers from trip demand analysis and other applications, is most commonly used.

$$f(\alpha - \Theta) = 1/(1 + e^{(\alpha-\Theta)}) \qquad (3)$$

where Θ is an internal threshold or offset, and the value of f varies from 0 to 1.

The training algorithm is initialized by setting all weights at small random values from -1 to 1 and specifying an input vector X and a desired output vector D. The actual output vector Y is calculated from Equations 1, 2, and 3. Based on the error between desired and actual output, the weights are updated using a recursive algorithm that begins at the output nodes and works back to the hidden and input layers.

DATA DESCRIPTION

The neural network detection method was developed with data collected from Interstate 35W, a heavily traveled and often congested freeway in Minneapolis, Minnesota. The study was confined to the afternoon peak period (4:00 p.m. to 6:00 p.m. because incident detection under moderate-to-heavy traffic conditions is of greatest importance for advanced freeway management.

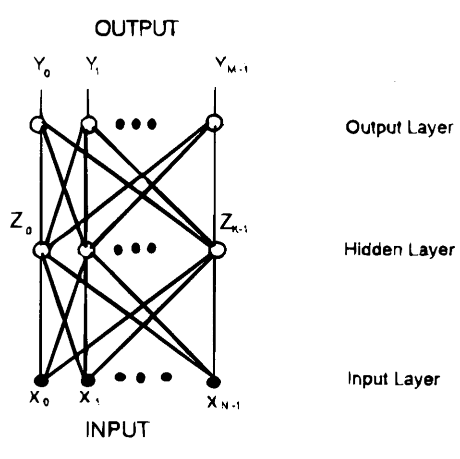

FIGURE 1 Three-layer feedforward neural network.

The selected 5.5 mi freeway segment shown in Figure 2 is fully covered by TV cameras, allowing detailed traffic information to be gathered. This segment includes most types of geometric configurations usually found on a freeway, such as entrance and exit ramps with or without exclusive lanes, bottlenecks; ramps carrying heavy volumes, etc. The data, which were obtained from 14 detector stations imbedded 0.3 to 0.7 mi apart in this segment, consist of 1-minute volume and occupancy updated every 30 sec and averaged over all lanes (see Table 1). A total of 140 hr of traffic data from 72 afternoon peak periods were used.

In the time period of this study, 31 incidents were reported by the Traffic Management Center of the Minnesota Department of Transportation. Confirmation of these incidents is made mainly through television cameras and recorded daily in incident logs by the TMC engineer. Incident logs include time and location of incident occurrence, incident type, duration, severity, impact on traffic, roadway condition, and other information.

Incidents in the data set, to be detected by the algorithm, include incidents blocking one lane, one or both shoulders, or a combination of lanes, shoulders, and freeway entrance-exit. There were no incidents blocking two or more lanes. Although the proposed algorithm is based on observable changes in the traffic flow all, incidents recorded by the traffic engineer were included, even if they had minimal or no impact on traffic. Detection of these incidents by the neural network algorithm proved to be most challenging.

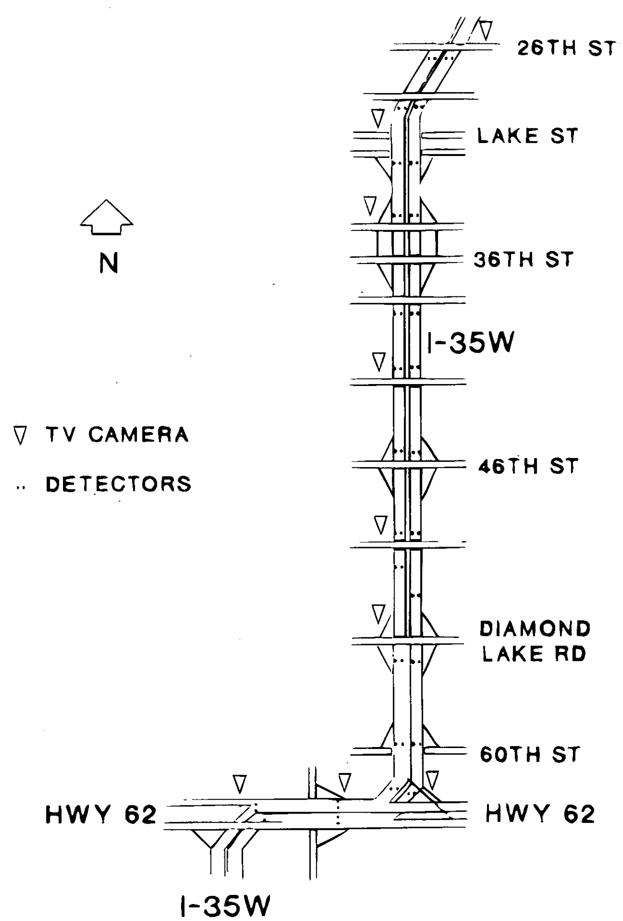

FIGURE 2 Study site in Minneapolis, 1–35W.

TABLE 1 Original Volume and Occupancy Traffic Data

Date: 12/06/1989 southbound, afternoon

Station / Time	042S		046S		050S		051S		055S		060S		061S		062S		063S	
16:10:30	22†	37‡	29	24	32	24	29	17	29	24	33	23	32	41	35	10	19	8
16:11:00	24	30	31	28	33	24	33	19	26	24	33	28	31	35	32	9	20	9
16:11:30	29	27	28	31	32	23	33	20	30	22	25	26	32	32	27	7	20	9
16:12:00	29	24	28	29	35	23	34	21	33	19	25	23	31	35	29	8	20	8
16:12:30	27	21	32	26	35	23	36	22	32	20	31	25	28	36	28	8	17	7
16:13:00	27	25	31	23	35	23	33	25	32	22	31	29	27	37	27	8	16	7
16:13:30	24	28	32	22	37	25	31	30	31	26	26	32	30	35	30	9	14	6
16:14:00	22	26	34	23	34	27	33	29	28	29	26	30	30	31	33	10	16	7
16:14:30	25	25	32	20	31	35	34	25	29	26	27	23	31	34	31	9	17	8
16:15.00	27	23	30	18	31	36	33	24	30	26	29	23	32	34	31	10	19	9
16:15:30	29	22	31	19	31	29	34	25	23	36	33	25	30	33	30	9	21	9
16:16:00	28	21	32	21	31	25	35	25	22	33	33	24	31	39	29	8	17	7
16:16:30	26	22	32	24	32	22	32	24	27	22	31	23	30	40	37	12	17	8
16:17:00	28	25	30	28	32	20	29	26	29	22	27	25	29	35	40	13	20	9
16:17:30	27	24	29	28	29	26	24	35	32	22	28	25	30	33	33	9	20	9
16:18:00	28	22	32	25	22	40	17	32	33	23	30	22	29	32	28	8	16	7
16:18:30	29	22	32	22	21	46	20	22	30	22	31	24	33	33	28	8	19	9
16:19:00	27	23	31	22	24	42	26	17	28	20	33	27	36	32	27	8	21	10
16:19:30	23	33	31	25	25	39	26	13	29	19	31	28	34	32	28	8	20	10
16:20:00	21	36	26	34	25	41	26	12	29	17	27	25	32	29	29	9	19	10
16:20·30	24	31	20	48	25	39	27	13	29	18	29	24	31	31	30	9	18	9
16:21:00	27	34	20	52	28	31	28	14	29	18	32	22	31	33	32	10	21	10
16:21:30	27	28	22	50	28	32	28	14	28	18	30	18	32	32	29	8	21	10
16:22·00	24	22	22	47	26	35	28	14	29	19	26	15	30	30	28	8	24	13
16:22:30	19	31	24	43	28	33	28	15	32	18	28	17	27	24	36	11	24	12
16:23:00	17	42	26	39	29	32	28	15	30	16	31	20	33	27	36	11	22	10
16:23:30	21	42	26	40	29	32	28	13	27	15	30	22	37	32	29	8	23	10
16:24:00	23	41	23	43	28	35	28	15	29	15	31	25	35	32	27	7	26	11
16:24:30	22	38	24	39	27	37	28	14	28	12	29	22	31	31	28	7	25	10
16:25:00	22	31	26	35	28	35	28	13	27	11	27	21	31	30	30	7	24	9
16:25:30	22	31	27	32	26	29	29	14	26	11	33	26	33	31	30	7	19	8
16.26:00	22	32	27	29	28	31	28	14	28	12	32	24	31	31	28	7	16	8
16:26:30	22	33	26	29	28	36	28	14	26	11	27	20	30	28	30	7	24	11
16:27:00	22	34	28	32	27	33	28	15	25	10	28	21	30	26	34	10	23	9
16.27:30	23	33	27	29	29	30	26	14	28	12	29	22	30	27	28	8	22	10

† Column 1 indicates volume (veh/min);
‡ Column 2 indicates occupancy (%).

INCIDENT DETECTION ALGORITHM

Reducing the number of false alarms that can result from short-term traffic inhomogeneities is a major objective of the incident detection algorithm developed in this study. To increase the transferability potential of the algorithm, it is kept as simple as possible. Following results from earlier work, the algorithm operates based on real-time traffic measurement at pairs of adjacent stations.

Earlier work has sought to achieve detection performance by exploiting the smoothed, normalized spatial occupancy difference between adjacent stations through time (*3*). To simplify the detection process, only raw data are employed in this work. Further, to take full advantage of all possible patterns presented by such data in real time, both occupancy and volume are presented to the neural network. These data are routinely available at traffic management centers across the United States and in other countries. Figures 3 and 4 show occupancy and volume data from a typical incident in our data set, occurring on December 6, 1989, at 16:18:00.

Several neural networks were investigated, all with 41 elements in the input layer and one in the output layer. The network that performed best had 30 nodes in the hidden layer and was selected for the remainder of this work. The number of training iterations was set at two levels; the first level, at which error was adequately reduced, was 1,500, and the second, at which the effects of over-learning are clear, was 4,000. The performance of the neural network for the data set was evaluated with respect to detection rate, false alarm rate, and time-to-detect. The best neural network algorithm was compared with incident detection algorithms previously found to perform best (i.e., DELOS, California, and Algorithm 7) (*10*).

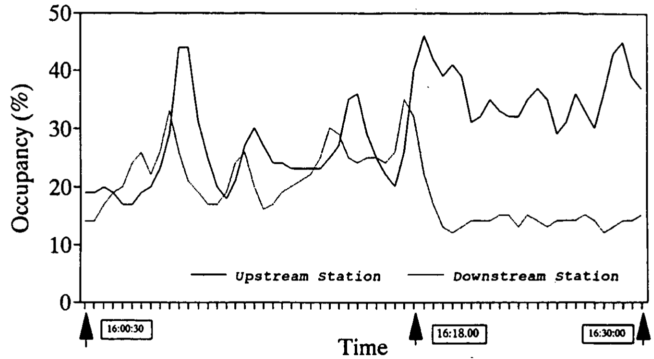

FIGURE 3 Detection occupancy data.

Training

The training of the neural network involves training with freeway data that include 31 reported incidents and with incident-free data randomly acquired from 14 of the 72 days in the data set. These are 30-sec station lane-average volume and occupancy data from 14 stations along the 5.5 mi freeway section described previously. For each incident, one or more 5-min patterns is introduced to the network depending on the duration of the incident, for a total of 89 training incident patterns.

Because the size of the dataset is limited, each consecutive incident pattern is placed at a 2.5-min overlap with its preceding pattern so the number of patterns with which the network is trained increases. The selection of 5-min pattern length and 2.5-min pattern overlap reflects findings from preliminary analysis of the data and is a function of station location, incident duration, and size of data set. Sensitivity analysis could be performed to determine the best pattern length and overlap in terms of algorithm performance. Training could be extended with additional pattern combinations from the data set. Because volume and occupancy data are collected every 30 sec, each sample includes 10 volume and 10 occupancy measurements from each of the two detector stations. As a result, each sample contains a total of 5 min $\times$ 2 measurements/min $\times$ 2 variables $\times$ 2 stations = 40 measurements.

The input vector of the neural network has 41 elements, including one element of constant 1 for adjusting the internal threshold, Θ_k, in Equation 2. The output of the neural network is one element and can vary between 0 and 1. During classification, an output value greater than a user-specified threshold (see later discussion on the

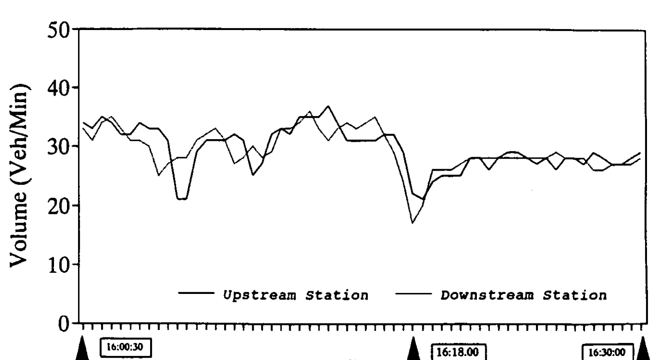

FIGURE 4 Detection volume data.

sensitivity of algorithm performance on the value of this threshold; default value is 0.5) indicates an incident, otherwise an incident-free traffic state is indicated. The desired output of an incident pattern is 1 and that of an incident-free pattern is 0. To illustrate, for an accident occurring between Stations 50S and 51S on December 6, 1989, at 16:18:00 (see original traffic data in Table 1), the first three incident patterns selected are listed in Table 2.

For training the network with incident-free patterns, 14 days in the data set were randomly selected, 336 5-min incident-free patterns were randomly acquired such that they average one per peak hour from each adjacent-station pair, for a total of 14 days × 2 hr/day × 12 station pairs. Because of the large number of available patterns, no pattern overlap was employed. In acquiring these patterns more weight was placed in areas in which previous work (*3*) had indicated a higher number of false alarms. For instance, the first three incident-free patterns between Stations 31S and 35S on December 6, 1989, beginning at 16:00:30, are listed in Table 3.

In each iteration of the training process, all 425 training patterns were presented to the network in random order. Because of the large number of patterns, a small gain was used for updating the network weights; as a result, a high number (above 1,000) of iterations were employed.

TABLE 3 Incident-Free Patterns

	31S		35S			31S		35S			31S		35S	
	V	O	V	O		V	O	V	O		V	O	V	O
16:00:30	23	10	24	09	16:05:30	27	12	24	09	16:10:30	28	12	25	09
16:01:00	25	12	21	08	16:06:00	29	12	25	09	16:11:00	28	12	24	09
16:01:30	24	11	24	09	16:06:30	27	11	28	10	16:11:30	28	12	26	10
16:02:00	24	11	23	08	16:07:00	28	12	26	10	16:12:00	28	12	26	10
16:02:30	28	12	21	08	16:07:30	33	15	24	09	16:12:30	29	13	24	09
16:03:00	29	12	25	09	16:08:00	29	12	30	11	16:13:00	29	13	27	10
16:03:30	30	13	27	10	16:08:30	26	11	28	10	16:13:30	32	14	27	11
16:04:00	31	13	26	10	16:09:00	26	11	24	08	16:14:00	32	14	28	11
16:04:30	30	12	28	10	16:09:30	24	10	24	09	16:14:30	30	13	28	11
16:05:00	27	11	28	10	16:10:00	27	11	23	08	16:15:00	31	13	27	10

The output of the test module is illustrated in Table 5, which reflects part of the testing on the December 6, 1989, data. The output indicates a correctly detected incident and a number of false alarms. The incident began between Stations 51S and 55S at 16:18:00 and was continuously detected from 16:19:30 (time-to-detect = 1.5 min) until 16:54:30. Further, four false alarms at zero persistence were indicated at 16:07:30 (Stations 46S-50S), 16:10:30 (Stations 50S-51S), 16:16:30 (Stations 61S-62S), and 16:18:00 (Stations 55S-60S). For every continuous false alarm series, one false alarm is recorded.

Testing

The neural network was tested through application to the complete data set over the 72-day period. Although the data set is the same as the one used for training, the testing procedure involved a substantially larger number of patterns, with both patterns containing incidents and incident-free patterns. For every two adjacent stations, a new pattern was defined in the data every 30 sec for a total of 211,536 test patterns. For example, the first three input patterns of station pair 50S and 51S on December 6, 1989, are shown in Table 4.

Every 30 sec the neural network classifies the state of traffic as either incident or incident-free based on the threshold defined by the user. After a persistence test, an incident alarm is declared. Four types of detection were tested based on persistence values of $P = 0$, 1, 2, and 3. For instance, if $P = 0$ following an incident classification at t, an incident alarm is declared. If $P = 3$ following an incident classification at time t, an incident alarm is declared only if an incident classification is also recorded at $t + 30$ sec, $t + 60$ sec, and $t + 90$ sec.

RESULTS

Results from testing indicate the neural network's sensitivity to the number of iterations, the user-specified threshold, and the persistence value. Higher threshold and persistence values reduce the false alarm rate, but also reduce the detection rate and increase the average time-to-detection. For instance, Table 6 indicates that at 1,500 iterations with zero persistence, increasing the user-specified threshold from 0.5 to 0.8 reduces the false alarm rate from 1.4 to 0.40 percent, but also reduces detection rate from 94 to 81 percent and increases average time-to-detection from 2.5 to 4.1 min. Similarly, if user-specified threshold value is kept constant at 0.5, increasing persistence from 0 to 3 reduces false alarm rate from 1.4

TABLE 2 Incident Patterns

	50S		51S			50S		51S			50S		51S	
	V	O	V	O		V	O	V	O		V	O	V	O
16:15:30	31	29	34	25	16:18:00	22	40	17	32	16:20:30	25	39	27	13
16:16:00	31	25	35	25	16:18:30	21	46	20	22	16:21:00	28	31	28	14
16:16:30	32	22	32	24	16:19:00	24	42	26	17	16:21:30	28	32	28	14
16:17:00	32	20	29	26	16:19:30	25	39	26	13	16:22:00	26	35	28	14
16:17:30	29	26	24	35	16:20:00	25	41	26	12	16:22:30	28	33	28	15
16:18:00	22	40	17	32	16:20:30	25	39	27	13	16:23:00	29	32	28	15
16:18:30	21	46	20	22	16:21:00	28	31	28	14	16:23:30	29	32	28	13
16:19:00	24	42	26	17	16:21:30	28	32	28	14	16:24:00	28	35	28	15
16:19:30	25	39	26	13	16:22:00	26	35	28	14	16:24:30	27	37	28	14
16:20:00	25	41	26	12	16:22:30	28	33	28	15	16:25:00	28	35	28	13

TABLE 4 Testing Input Patterns

	50S		51S			50S		51S			50S		51S	
	V	O	V	O		V	O	V	O		V	O	V	O
16:00:30	34	19	33	14	16:01:00	33	19	31	14	16:01:30	35	20	34	17
16:01:00	33	19	31	14	16:01:30	35	20	34	17	16:02:00	34	19	35	19
16:01:30	35	20	34	17	16:02:00	34	19	35	19	16:02:30	32	17	33	20
16:02:00	34	19	35	19	16:02:30	32	17	33	20	16:03:00	32	17	31	24
16:02:30	32	17	33	20	16:03:00	32	17	31	24	16:03:30	34	19	31	26
16:03:00	32	17	31	24	16:03:30	34	19	31	26	16:04:00	33	20	30	22
16:03:30	34	19	31	26	16:04:00	33	20	30	22	16:04:30	33	23	25	26
16:04:00	33	20	30	22	16:04:30	33	23	25	26	16:05:00	31	29	27	33
16:04:30	33	23	25	26	16:05:00	31	29	27	33	16:05:30	21	44	28	26
16:05:00	31	29	27	33	16:05:30	21	44	28	26	16:06:00	21	44	28	21

TABLE 5 Output of Neural Network

Date: 12/06/1989 southbound, afternoon									
Station Time	042S	046S	050S	051S	055S	060S	061S	062S	063S
16:05.30	-	-	-	-	-	-	-	-	
16·06:00	-	-	-	-	-	-	-	-	
16·06:30	-	-	-	-	-	-	-	-	
16:07·00	-	-	-	-	-	-	-	-	
16:07:30		Inc	-	-			-	-	
16.08·00	-	Inc	-	-	-	-	-	-	
16·08:30	-	-	-	-	-	-	-	-	
16:09:00	-	-	-	-	-	-	-	-	
16:09:30	-	-	-	-	-	-	-	-	
16:10:00	-	-	-	-	-	-	-	-	
16·10:30	-	-	Inc	-	-	-	-	-	
16:11.00	-	-	Inc	-	-	-	-	-	
16.11.30		-	-	-	-	-	-	-	
16:12:00	-	-	-	-	-	-	-	-	
16:12:30	-	-	-	-	-	-	-	-	
16:13:00	-	-	-	-	-	-	-	-	
16:13:30	-	-	-	-	-	-	-	-	
16:14:00	-	-	-	-	-	-	-	-	
16:14:30		-	-	-	-	-	-	-	
16·15:00	-	-	-	-	-	-	-	-	
16:15:30	-	-	-	-	-	-	-	-	
16·16:00	-	-	-	-	-	-	-	-	
16:16·30	-	-	-	-	-	-	Inc	-	
16:17:00	-	-	-	-	-	-	Inc	-	
16:17:30	-	-	-	-	-	-	Inc	-	
16:18.00	-	-	-	-	Inc	-	Inc	-	
16.18:30	-	-	-	-	-	-	Inc	-	
16:19:00	-	-	-		-	-	Inc	-	
16:19:30	-	-	-	Inc	-	-	-	-	
16:20:00	-	-	-	Inc	-	-	-	-	
16:20:30	-	-	-	-	-	-	-	-	
16:21:00	-	-	-	-	-	-	-	-	
16:21:30	-	-	-	-	-	-	-	-	
16:22:00	-	-	-	Inc*	-	-	-	-	
16:22:30	-	-	-	Inc	-	-	-	-	
...	.	...	...	...	...	...	...	...	
16.54:30	-	-	-	Inc	-	-	-	-	

* Neural network detects incident.

to 0.40 percent, but also reduces detection rate from 94 to 84 percent and increases time-to-detect from 2.5 to 4.7 min. These effects are illustrated in Figure 5, in which the performance envelope of the neural network is also demonstrated.

The performance of the neural network algorithm was evaluated at different numbers of iterations. Two of these, 1,500 and 4,000 iterations, are reported in this study. A comparison of Tables 6 and 7 or Figures 5 and 6 indicates that the performance of the network at 4,000 iterations is worse than that at 1,500 iterations. Although additional sensitivity analysis may further specify the range of iterations for best performance of the neural network, the results indicate that at least 1,500 iterations are required for the network to be adequately trained and that 4,000 iterations will result in overtraining, which reduces the network's performance.

The performance of the neural network algorithm was compared with the best of the existing algorithms that have been calibrated and extensively tested and evaluated for this data set (*10*). These include Minnesota Algorithm DELOS 3.3 (0.05,6). Minnesota Algorithm DELOS 1.1 (10,6), Algorithm 7, and the California algorithm, in order of decreasing performance.

The California algorithm consists of three comparison tests to preset thresholds. An incident is detected (*a*) when upstream occupancy is significantly higher than downstream occupancy both in absolute value and relative to upstream occupancy and (*b*) when downstream occupancy has adequately decreased during the past 2 min. The last test distinguishes an incident from a bottleneck by indicating that a reduction in downstream occupancy has occurred over a short period of time as a result of the incident.

TABLE 6 Performance Results at 1,500 iterations

Persistence	Threshold	Detection Rate (%)	False Alarm Rate (%)	Average Detection Time (Min.)
0	0 5	94	1.4	2 5
	0.6	90	1.1	2.9
	0.7	81	0.70	3.4
	0.8	81	0.40	4.1
1	0.5	94	0.92	3.2
	0.6	87	0.68	3.9
	0.7	81	0.36	4.4
	0.8	74	0.19	4.8
2	0 4	87	0.75	3.3
	0.5	87	0.58	4.1
	0.6	84	0.40	4.7
	0.7	74	0.19	5.2
	0.8	58	0.10	5.6
3	0.4	87	0.53	3.9
	0.5	84	0.40	4.7
	0.6	81	0.26	5.4
	0.7	71	0.12	5.7
	0.8	48	0.069	6.2

Algorithm 7 is similar to the California algorithm, but it replaces the temporal downstream occupancy difference in the third test with the present downstream occupancy measurement. This replacement seeks to reduce the false alarms produced by compression waves.

The Minnesota algorithms use low-pass filtering of the occupancy measurements to distinguish short-term traffic inhomogeneities from incidents. Further, the algorithms attempt to distinguish recurrent congestion from incident congestion based on slow

TABLE 7 Performance Results at 4,000 Iterations

Persistence	Threshold	Detection Rate (%)	False Alarm Rate (%)	Average Detection Time (Min.)
0	0.5	94	2.9	1.9
	0.6	94	2.5	2.1
	0.7	94	2.2	2.3
	0 8	90	1.8	2.2
1	0.5	94	1.4	3.1
	0.6	94	1.2	3.2
	0.7	94	0.99	3.3
	0.8	84	0.78	3.2
2	0.4	90	0.86	3.7
	0.5	90	0.73	3.7
	0.6	90	0.62	4.2
	0.7	81	0.49	4.6
	0.8	65	0.38	4.5
	0.9	39	0.25	4.4
3	0.4	87	0.55	4.5
	0.5	81	0.47	4.0
	0.6	74	0.37	4.4
	0.7	61	0.29	4.9
	0.8	52	0 21	4.8
	0.9	26	0.11	4.0

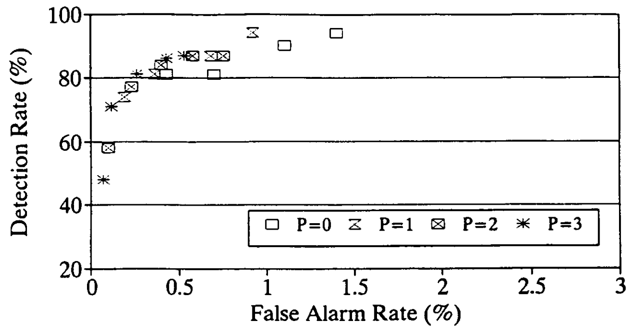

FIGURE 5 Neural network performance at 1,500 iterations.

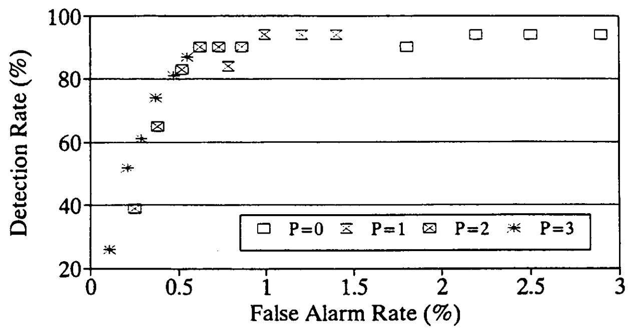

FIGURE 6 Neural network performance at 4,000 iterations.

or fast evolution of the congestion, respectively. The distinguishing logic of the two tests used is based on a temporal comparison of spatial occupancy difference between adjacent stations. Assuming an incident occurs at t, the congestion test considers the smoothed spatial occupancy difference from k time increments after t, normalized by the highest value of the smoothed upstream and downstream occupancies from n increments before t. The incident test compares the smoothed spatial occupancy difference for the period after t with the corresponding value from the past period. In particular, DELOS 1.1 (10,6) uses a moving average to smooth 10, 30-sec past occupancy values, and 6 present values. DELOS 3.3 (0.05,6) uses exponential smoothing with a smoothing factor of 0.05, and a time lag of 6 between the periods before and after the incident.

The evaluation results indicate that the neural network performs better than all algorithms in the set and, at a detection rate of 70 percent or higher, performs as well as the best algorithm, DELOS 3.3. This performance is noteworthy because the neural network represents the initial results in its class, developed in Minnesota with real data, whereas DELOS 3.3 was developed after considerable research. A more fair comparison would be between the neural network and DELOS 1.1, which also represents the initial findings in its class *(3)*. To illustrate, as Figure 7 suggests, at 70 percent detection rate, the false alarm rate of the neural network is 0.12 percent and that of DELOS 3.3 is 0.13 percent. The false alarm rate of DELOS 1.1 is 0.25 percent, or twice the number of false alarms of the neural network; that of Algorithm 7 is 0.34 percent, or approximately three times as many false alarms; and the false alarm rate of the California algorithm is 0.52 percent, or more than four times the number of false alarms produced by the neural network. Future versions of the neural network are expected to (*a*) improve time-to-detection by using a more appropriate pattern size, and (*b*) further decrease the number of false alarms by preprocessing and normalizing the field data before analysis.

CONCLUSION

A neural network algorithm that can be used to improve automated incident detection in freeways was discussed. Based on real-time occupancy and volume counts from pairs of adjacent loop detector stations, the three-layer feedforward network with approximately 1,000 nodes was trained with actual data, including 31 incidents from I-35W (a typical freeway in the Twin Cities Metropolitan

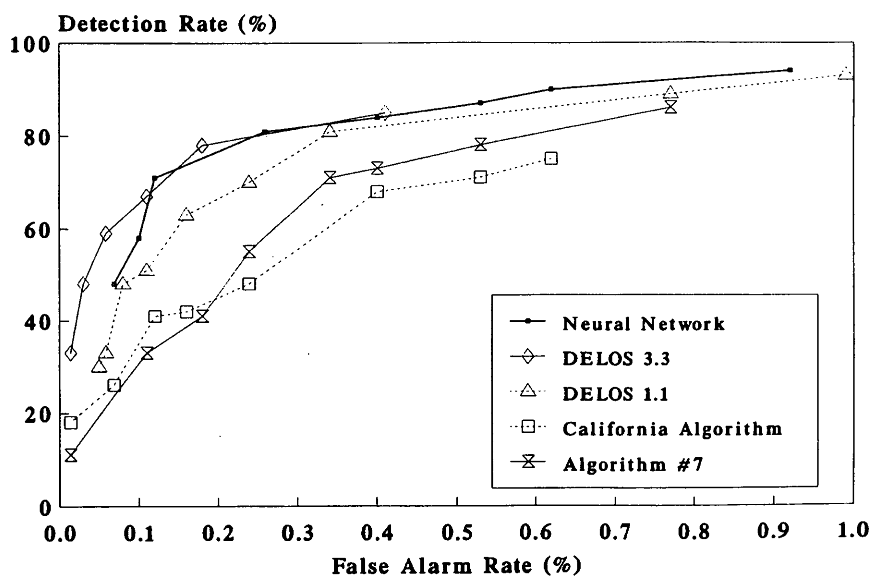

FIGURE 7 Algorithm performance comparison.

Area) over the afternoon peak period. Training with data from 72 days yielded promising test results and indicated that the algorithm was able to learn and classify incident and incident-free patterns effectively. Methods for improving the time-to-detect incidents are currently being developed by the authors.

Test results also indicated the sensitivity of algorithm performance to values of user-specified threshold, persistence, and the number of iterations used for training. Algorithm performance in terms of detection and false alarm rates was superior to most of the best algorithms that have been tested with this data-set. At a detection rate of 70 to 80 percent, the trained network has a false alarm rate of 0.12 percent to 0.26 percent. The computation time for one test is 4 msec on IBM-486/33MHz, indicating that it is practical to conduct incident detection in real time. Algorithm testing is continuing with the collection of additional incident data in the Metropolitan Area of Minneapolis-St. Paul.

ACKNOWLEDGMENT

The authors gratefully acknowledge Athanasios Chassiakos, Department of Civil-Mineral Engineering, University of Minnesota, for his helpful remarks. This work was supported in part by the National Science Foundation and the Minnesota Supercomputer Institute. The Center for Transportation Studies, Department of Civil-Mineral Engineering, University of Minnesota, is also acknowledged for its support. The Traffic Management Center, Minnesota Department of Transportation, cooperated in the study by providing the necessary data.

REFERENCES

1. Judicky, D., and J. Robinson. Managing Traffic During Nonrecurring Congestion. *Institute of Transportation Engineers Journal,* Vol. 62, No. 3, March 1992, pp. 21–26.
2. Kay, J. Intelligent Vehicle-Highway Systems and Incident Management. *Institute of Transportation Engineers Journal,* Vol. 62, No. 3, March 1992, pp. 55–57.
3. Stephanedes, Y., and A. Chassiakos. Application of Filtering Techniques for Incident Detection. *ASCE Journal of Transportation Engineering,* Vol. 119, No. 1, Jan./Feb. 1993, pp. 13–26.
4. Lippmann, R. An Introduction to Computing with Neural Nets. *ASSP Magazine,* Vol. 4, No. 2, IEEE, April 1987, pp. 4–22.
5. Hopfield, J., and D. Tank. Computing with Neural Circuits: A Model. *Science,* Vol. 233, Aug. 1986, pp. 625–633.
6. Carpenter, G., and S. Grossberg. The Art of Adaptive Pattern Recognition by a Self-Organizing Neural Network. *Computer,* Vol. 21, 1987, pp. 77–88.
7. Kohonen, T. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin, 1989.
8. Rumelhart, E., G. Hinton, and R. Williams. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations* (D. Rumelhart and J. McClelland, eds), MIT Press, 1986, pp. 318–362.
9. Cheu, R., S. Ritchie, W. Recker, and B. Bavarian. Investigation of a Neural Network Model for Freeway Incident Detection. In *Artificial Intelligence and Civil and Structural Engineering* (B. Topping, ed), Civil-Comp Press pp. 267–274.
10. Chassiakos, A., and Y. Stephanedes. Smoothing Algorithms for Incident Detection. In *Transportation Research Record 1394*, TRB, National Research Council, Washington, D.C., 1993, pp 8–16.

Publication of this paper sponsored by Committee on Freeway Operations.