

1999

## Incorporating neural network traffic prediction into freeway incident detection

Benjamin Todd Taggart  
*West Virginia University*

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

---

### Recommended Citation

Taggart, Benjamin Todd, "Incorporating neural network traffic prediction into freeway incident detection" (1999). *Graduate Theses, Dissertations, and Problem Reports*. 958.  
<https://researchrepository.wvu.edu/etd/958>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

**INCORPORATING NEURAL NETWORK TRAFFIC  
PREDICTION INTO FREEWAY INCIDENT DETECTION**

Benjamin T. Taggart

Thesis Submitted to the College of Engineering and Mineral Resources at West Virginia  
University in Partial Fulfillment of the Requirements for the  
degree of

Masters of Science  
In  
Civil Engineering

David R. Martinelli, Ph.D, Chair  
Ronald W. Eck, Ph.D.,  
Darrell Dean, Jr., Ph.D.

Department of Civil and Environmental Engineering

Morgantown, WV  
1999

Keywords: Incident Detection, Traffic Prediction, Neural Networks

## ABSTRACT

Incorporating Neural Network Traffic Prediction into Freeway Incident Detection

Master's Thesis

Benjamin T. Taggart

The efficient operation of an incident management system depend

Neural network models have been applied to traffic prediction frequently and even repeatedly because of its superior capability in emulating nonlinear systems. However, these traffic prediction models have not been utilized for incident detection. On the other hand, it is expected that the performance of an incident detection algorithm can be improved if an advanced prediction model is incorporated into. Therefore, this study developed several traffic prediction models that were then integrated into incident detection algorithms. The traffic prediction models were developed based on three different choices of independent variables, while the incident detection algorithms employed different decision functions. The results show that a good prediction model can improve the performance of an incident detection algorithm only when the decision function of the algorithm is appropriately chosen.

## Acknowledgements

I would like to express my grateful appreciation to my co-advisors Dr. David Martinelli and Dr. Ronald Eck for making it possible for me to attend graduate school. You both have shown a tremendous amount of confidence in my abilities by providing me the unique opportunity to explore the entire realm of transportation engineering. Dr. Martinelli served as my academic and research advisor. Under his patience and guidance I have been able to acquire vast amount of knowledge in the realms of artificial intelligence, and operations research. Dr. Eck served as my advisor while I worked in the West Virginia Transportation Technology Transfer Center. During the course of my college career, Dr. Eck has provided me with excellent advice and instruction both in and out of the classroom.

I would extend my appreciation to Dr. Darrell Dean, Jr. for taking time out of his busy schedule to serve on my committee, and also for serving as my undergraduate advisor.

I would also like to extend acknowledgement to my mentors, Dr. Hualiang Teng, and Michael Blankenship. Dr. Teng provided the immediate supervision for this research, while Mike served as my immediate supervisor in the T<sup>2</sup> Center. Both of you have been extremely patient, and have helped to further my professional development as well as provided me with excellent guidance. You both were very understanding and flexible in letting me split my time between work and research.

I would like to thank Robin Altobello, Denise Brooks, Kim Carr, Crystal May, David Turner, and Sandy Wotring for all of the assistance they have provided me throughout my college career.

I would like to thank Jennifer Peterson for all of her help and assistance in filing this document electronically.

Finally, I would like to extend a special acknowledgement to my parents, Jeff and Lou Ann Taggart. They have provided me with the care, support, and encouragement to succeed in all of my academic endeavors. Thank you for everything.

## Vita

The author was born in Uniontown, Pennsylvania on July 4, 1975. He graduated in the top 25% of his class at Laurel Highlands Senior High School in Uniontown, Pennsylvania in May of 1993. He then obtained a Bachelor's degree of Science in Civil and Environmental Engineering from West Virginia University in December of 1997.

The author then joined the Transportation Engineering graduate program at West Virginia University in January of 1998. While in the graduate program the author has conducted research under the advisement of David Martinelli, Ph.D. and Hualiang Teng, Ph.D. The author has also served as a Technical Assistant in the West Virginia Transportation Technology Transfer Center while working under the guidance of Ronald Eck, Ph.D., P.E. and Michael Blankenship, P.E. He anticipates taking a traffic engineering position with the firm of Barakos-Landino, Inc. in June of 1999.

The author is a candidate for the degree of Master of Science in Civil Engineering in May 1999.

## TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
VITA	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1 INTRODUCTION	1
1.1 Incident Management Systems	1
1.2 Incident Detection	2
1.3 Detection Algorithms	3
1.4 Problem Statement	4
1.5 Objectives	6
1.6 Thesis Overview	
CHAPTER 2 LITERATURE REVIEW	8
2.1 Traffic Prediction	8
2.2 Introduction to Neural Networks	9
2.3 Neural Network Traffic Prediction Models	13
2.4 Detection Algorithms	16
2.5 Neural Network Incident Detection Algorithms	21
CHAPTER 3 METHODOLOGY	23
3.1 Traffic Data	23
3.2 Neural Network Traffic Prediction	27
3.3 Incident Detection Algorithm	35
CHAPTER 4 RESULTS	39
4.1 Performance of the Neural Network Traffic Prediction Model	39
4.2 Performance of the Incident Detection Algorithm	44
CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS	50
REFERENCES	52

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
2.1	Comparison of Prediction Models (Smith and Demetsky)	14
2.2	Input-Output Specifications for BPN Predictor (Kwon and Stephanedes)	15
2.3	Estimated Performance Results of Different Detection Algorithms (Black)	18
3.1	Incident Classification on I-880 (Skarbodinis)	26
3.2	Sample Data Sheet for the Time Only NNTP Model	28
3.3	Sample Data Sheet for the Occupancy Only NNTP Model	28
3.4	Sample Data Sheet for the Time and Occupancy NNTP Model	29

## LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
2.1	Figure of a Neural Network	12
3.1	Map of I-880 by Petty	25
3.2	Mean of the Occupancy Data	30
3.3	Standard Deviation of the Occupancy Data	31
3.4	Diagram of the Occupancy Only NNTP Model	34
3.5	TransGuide Map	37
4.1	Training Curves for the Southbound NNTP Models	41
4.2	Training Curves for the Northbound NNTP Models	42
4.3	Comparison of the Raw Data to the Output of the NNTP Models	43
4.4	Incident Detection Performance Curves	49

# Chapter 1

## Introduction

The capacity of a freeway is determined by the number of lanes and the environment in which it operates. Incidents inhibit the efficiency at which a freeway operates. The Federal Highway Administration (FHWA) defines an incident as any non-recurring event, which causes reduction of capacity or abnormal increases in demand. According to the FHWA (1), incidents cause more than 60% of all delays on urban freeways, and cost society an estimated 34 billion dollars in lost revenues every year. The FHWA also found that there are approximately 10-25 incidents per million-vehicle miles (1). For every one minute of incident there will be four minutes of congestion after the incident is cleared.

The traditional method of increasing capacity is to add more lanes. This method is costly, and in many places impossible because of a lack of resources. An operational approach to managing traffic is needed. As a result, incident management systems have been developed to efficiently manage traffic, minimize the reduction of capacity upstream from the incident, as well as minimize the duration of the incident.

### 1.1 Incident Management Systems

Incident management systems consist of four basic components: detection, verification, response, and recovery. The first step, detection, determines that an incident has occurred and conveys this message to a central dispatcher. The next step, verification, determines the precise location and nature of the incident. After the incident is verified the next step, response, organizes and sends the appropriate resources to the scene of the incident. The final step, recovery, clears up the incident and returns the freeway to normal traffic conditions.

The benefits that incident management systems can provide to the freeway system include decreased incident duration and a reduction in the number of secondary incidents as well as improved overall safety of the freeway. Incident management systems can significantly decrease the duration of an incident by providing quick, precise reactions to problems that may arise on the freeway. These systems can also help reduce the number of secondary incidents. Secondary incidents (20% of all incidents) typically occur upstream from the incident and are usually the result of overheating, running out of fuel, and minor braking accidents (1). By managing the traffic around incidents and minimizing the duration, the frequency of secondary incidents will be lowered. Improved safety will be obtained by shortening the response time for emergency vehicles to arrive at the scene of an accident.

## **1.2 Incident Detection**

Incident detection is the most crucial step in an incident management system because it initiates the process. Therefore, rapid detection is necessary to minimize the duration in which capacity is reduced.

There are many types of incidents, such as vehicle breakdowns, multi-car accidents, road construction, and special events. However, all incidents can be classified into two groups, predictable and unpredictable. Predictable incidents are a result of maintenance and construction activities or special events such as sporting events or parades. Unpredictable incidents occur from things such as accidents, break downs, weather, and other acts of nature. The magnitude of accidents can also be broken down into two types, major and minor. Major incidents last thirty minutes or longer and block at least one lane of traffic. Minor incidents have a duration of less than thirty minutes and block only the shoulder (1).

There are many different methods for detecting incidents. These methods include motorists using CB radios and cellular phones, patrol vehicles, aerial surveillance, closed circuit television, and use of electronic sensors. All of these techniques have certain advantages and disadvantages. This study will focus mainly on using electronic sensors, such as loop detectors, which are beneficial because they provide automatic detection. Loop detectors can provide automatic detection because they are continuously collecting real-time data such as speed, volume and occupancy. The raw data are then sent back to a computer that can detect changes in traffic conditions and detect incidents using a detection algorithm. Electronic sensors are also very consistent in collecting data, and work well in any weather condition. Other benefits of electronic sensors include the ability to detect incidents independently without the aid of conducting regular patrols of the area or relying on motorists to report the incident.

Automatic incident detection is a complex problem because of the diverse characteristics related to non-recurring congestion. Although an average number of incidents can be expected to occur, the exact time, type, and duration of each incident is random and unpredictable. These events are non-recurring and non-linear in nature. The events also require the classification of data. The most important classification the algorithm has to make is whether traffic is in free flow or incident condition. This can be challenging during the beginning and ending of peak traffic flow conditions.

### **1.3 Detection Algorithms**

The purpose of a detection algorithm is to evaluate the variances in traffic conditions to determine if an incident has occurred. There are two major parts of a detection algorithm, the decision function and the threshold. The decision function operates by comparing current traffic conditions to a predicted value of the conditions that the algorithm forecasts. The

difference between the actual conditions and the predicted conditions is then compared to the threshold.

The threshold is a set value at which the decision function must be greater than to set off an incident alarm. If the threshold is “low” the detection rate will be “high,” and the algorithm could generate many of false alarms. If the threshold is “high,” the false alarm rate will be relatively “low,” but the algorithm will not detect as many incidents. To improve the algorithm performance, a precise decision function must be created to extract the best performance of the threshold value.

#### **1.4 Problem Statement**

Incident detection algorithms have deficiencies that, if improved, can enhance the overall performance. The goal in developing a good incident detection algorithm is to optimize the detection rate while keeping the false alarm rate as low as possible. As stated earlier in this chapter, the relationship between the detection rate and the false alarm rate is that if one value is high the other one will be high as well. To improve the detection rate without increasing the false alarm rate, weaknesses that cause false alarms must be strengthened. One particular weakness occurs when traffic patterns change between peak and off peak periods. Most detection algorithms have a traffic prediction model that forecasts traffic conditions and updates the decision function in the detection algorithm. These functions are based primarily on historical data or moving averages of real time data. Recent research in traffic prediction has applied neural networks in order to build more accurate forecasting models. These models have been very accurate and have been able to work from real time data.

Although neural networks have been successfully applied to both traffic prediction and incident detection, they have yet to be applied as a traffic prediction model to an incident

detection algorithm. Neural network traffic prediction (NNTP) models have also never used time as an input to construct a time series model. Since the possibilities of these methods have not been explored (although they have been successfully applied independently), the development is worthy of study.

The expected benefits from integrating a neural network traffic prediction model into an incident detection algorithm will be to improve the overall performance of the incident detection algorithm by increasing the detection rate without substantially raising the false alarm rate. The possible increased performance would occur as a result of outputting a more accurate traffic variable into the incident detection algorithm decision function. The potential improvements in the decision function will stem from the possible improvement in prediction accuracy that is provided by the NNTP model. The NNTP forecasted values such as occupancy should be closer to actual traffic conditions than those determined previously by historical data and moving averages. This increase in accuracy should carry over and improve the precision of the decision function, which in turn will maximize the efficient range of the threshold. These improvements should translate into higher detection rates and lower false alarm rates for the incident detection algorithm.

### **1.5 Objectives**

To investigate the applications of integrating a neural network traffic prediction model (NNTP) into an incident detection algorithm, the following tasks were developed:

1. To obtain highway traffic data that contains incidents.
  - Evaluate the traffic data to identify patterns and to determine the variance of the data between segments.
  - Determine the time and locations of the incidents in the data.

- Separate the data that has incidents from the days that do not.
2. To create an accurate NNTP.
    - Determine the inputs and outputs of the NNTP models.
    - Determine the most efficient structure of the NNTP models.
    - Create incident free data training data sets.
    - Train and test the NNTP models.
  3. To integrate the NNTP model into an incident detection algorithm.
    - Create data sets that include incidents.
    - Select a detection algorithm for evaluation.
    - Incorporate the output of the NNTP into the decision function of the NNTP.
    - Test the modified and original detection algorithms using the incident data sets.
    - Evaluate the performance of the detection algorithms.
  4. To develop conclusions and recommendations of the work conducted in this project.

## **1.6 Overview**

This study will investigate the prospects of incorporating a time series based neural network traffic prediction model into an incident detection algorithm. Chapter Two presents a thorough literature review to examine previous research in the areas of traffic prediction and incident detection. From this literature review, a solid knowledge base will be developed to encompass the successes and deficiencies of previous research. Chapter Three defines the methodology of how this research is conducted. Included are data collection, the training and construction of the traffic prediction model, as well as the selection and the process of integrating the NNTP into the incident detection algorithm. Chapter Four discusses the results of the NNTP

and the incident detection algorithm. Finally, Chapter Five discusses the results and makes conclusions and recommendations for further research.

## Chapter 2

### Literature Review

A literature review was conducted to explore the published research that has been conducted on traffic prediction and incident detection. The goals of this review were to assess current incident detection algorithms and traffic prediction models, and to see how neural networks have been applied to both traffic prediction and incident detection.

#### **2.1 Traffic Prediction**

Traffic prediction is an essential function of any traffic management system. Over the years, four types of prediction models have been developed; data based algorithms, time series models, simulations, and neural networks.

Data based algorithms are based on the premise that traffic patterns are typical, and a function of historical data. The historical data from a segment of freeway is collected over a period of time, and then compiled to provide the basis for prediction. The problem that arises from using historical data is that there is no way to monitor the current traffic situation for any events or incidents that may cause irregularities. The original Urban Traffic Control System (UTCS) functions off of historical data. The second UTCS (2) model (UTCS-2) was developed to incorporate real time data to make corrections for deviations from the “average historical pattern.” UTCS-3 (3) utilizes real time data only to predict traffic conditions. However this model has problem because of time lag. The weakness of data based algorithms is that they assume the projection ratio is constant which creates large errors.

The time series model uses real time data to statistically estimate traffic in the next interval. The most noted time series model has been the Box and Jenkins method. It has been shown (4) to yield accurate results in many different applications. The auto regressive integrated

moving average (ARIMA) method is the most refined version of Box Jenkins that has been applied to traffic applications. ARIMA models are linear estimators regressed on past values of the modeled time series. (4) The ARIMA model requires little computational time which makes it favorable for traffic applications. However, Stephanedes (5) found that in trying to apply ARIMA to UTCS that it generated high errors. Other statistical models include the exponential smoothing method, the double exponential smoothing method, and the Kalman filter method.

A variety of simulation models have been developed to represent how traffic systems operate under different conditions. These models are extremely useful for traffic engineers because of the ability to change different traffic variables to learn how a system will function in different scenarios. On-line simulation provides the ability to project future traffic patterns while considering any previously implemented strategies in real time (6). Unfortunately there is no way to incorporate real time data which renders them useless for applications in the field.

Although data-based algorithms, time series models and simulation models all serve a purpose in traffic management, there is still a need for an accurate real time traffic prediction model. Traffic prediction is a complex problem, and involves the classification of data, which tends to be non-linear, and have a high degree of uncertainty. Neural network models are excellent in solving these types of problems, and have been applied to traffic prediction.

## **2.2 Introduction to Neural Networks**

Neural networks were developed based on the objective of emulating the human brain for computation. As introduced in Lawrence (7), the human brain is a complex biological network of hundreds of billions of special cells called neurons. These neurons send information back and forth to each other through connections; the result is an intelligent system capable of learning,

analysis, prediction, and recognition. Artificial neural networks are formed a series of simulated neurons that are connected in a manner similar to the brain's neurons and are thus able to learn in a manner similar to people.

In Lawrence (7), neural network models are first classified based on their network architecture: some networks contain feedback, and some do not. Next, neural network models are further categorized by their neuron transfer function, i.e., linear and nonlinear models. In addition, neural networks can be classified again by their learning algorithm, which is usually either supervised or unsupervised. A supervised network has its output compared to known good answers during training. In neural network traffic prediction (NNTP) and neural network incident detection (NNID) models, the most widely applied neural networks are back propagation networks (BPN), which belong to the class of supervised nonlinear feed forward network. Other applied neural networks are probabilistic neural networks and its close relative, the radial basis function, which are new unsupervised feed forward models, and the adaptive resonance theory network, which is a feedback network.

Basically, back propagation is a supervised learning scheme by which a layered feed forward network with continuously valued neurons is trained to become a pattern-matching machine. The network learns by making corrections to the connections, based on the error at the output. Correction signals propagate back through the network during training. As training progresses, the amount of error is minimized (7). Figure 2.1 presents a BP neural network with a structure of three layers. The input layer consists of  $m$  neurons and a neuron for bias; the hidden layer includes  $r$  neurons and one neuron for bias; and the output layer has  $n$  neurons. For the connection between neurons on different layers, weights are assigned and changed during training. At the beginning, the weights are initialized to small random numbers:

$-0.1 \leq w_{ij}^m \leq 0.1$ . The next step is to choose an input vector  $X$  and let  $O_i^1 = x_i$  in  $X$ , where  $O_i^1$  is the output  $i$  in layer 1, i.e., the input layer. Then, the signal can be propagated forwards for  $m = 2, 3, \dots, M$  by:

$$O_j^m = g(\text{net}_j^m) = g(\sum_i w_{ij}^m O_i^{m-1}). \quad (1)$$

After that, error  $\delta_i^m$  of output in layer  $M$  is computed as:

$$\delta_i^m = g'(\text{net}_i^m)(d_i - O_i^m), \quad (2)$$

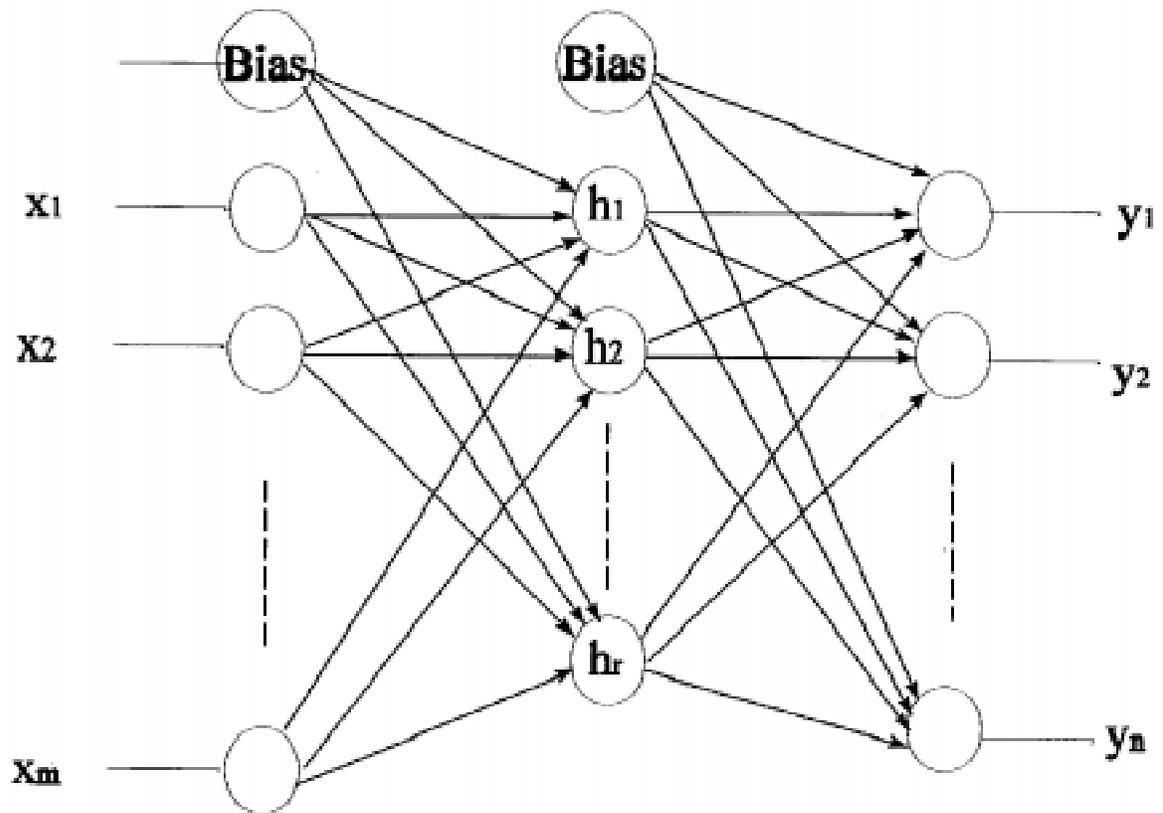
where  $d_i$  is the desired output,  $O_i^M$  is the output from neural network models. Error in hidden layer's  $\delta_i^m$  can also be computed for  $m = M, M-1, M-2, \dots, 2$ :

$$\delta_i^{m-1} = g'(\text{net}_i^{m-1}) \sum_j w_{ij}^m \delta_j^m. \quad (3)$$

Given the errors, the weights can be adjusted by:

$$\Delta w_{ij}^m = \alpha O_i^{m-1} \delta_j^m. \quad (4)$$

The process of weight adjustment is carried out repeatedly until the errors fall within a certain range.



**Figure 2.1 An Example of Neural Network Structure**

### **2.3 Neural Network Traffic Prediction Models**

Neural networks have been applied to traffic prediction models and been able to accurately forecast traffic. The NNTP models have been used for traffic signal controls, freeway volume prediction, and ramp volume prediction. Most of the current NNTP models have been used to forecast traffic volumes.

Dougherty, et al. (8) applied two approaches to neural networks to three different problems, one of which is traffic flow prediction. The input consists of flow from the current and two previous time intervals for one link in one approach and for two consecutive links in another approach. This results of this research showed promise that neural networks could be used to predict traffic.

Zhang, et al. (9) performed an exercise of emulating a nonlinear, dynamic relationship for speed at the downstream end of a freeway segment for subsequent time intervals given speed and density at both the upstream and downstream ends and the associated ramp entry rate in the freeway segment in the current time interval. This research concluded that neural networks were able to accurately model traffic flow, and that the networks were computationally efficient enough that they could be implemented into real time. Unfortunately, the data used to train the neural network model was from a simulation model, thus reducing the reliability of the conclusions.

Neural networks have been shown to outperform other types of prediction models. Smith and Demetsky (10) conducted research using neural networks to predict volume. The input of the training patterns consisted of the volume of both the current and one previous time interval, historical volume, historical volume in the next time interval, average speed and wet pavement

conditions in the current interval. They compared the NNTP model to the ARIMA model, and a historical average. The results of this research, shown in Table 2.1, clearly shows that the NNTP produced fewer errors than the other two models.

**Table 2.1 Comparison of Prediction Models (Smith and Demetsky)**

Model	Root Square Mean Error	Average Absolute Error
Historical Average	2730	146
ARIMA	3490	195
BPN	2620	144

Chang and Huarang (11) explored two methods to predict volume. The first model used an average traffic volume over five lanes, while the second model used only the volume from the fifth lane. Each model was constructed with two different structures. The first structure used the volumes from the past three intervals, while the second structure used the volumes from the past 5 intervals. The output of the NNTP was then processed through an error correction filter to adjust the data. The study concluded that:

- NNTP could quickly model traffic patterns
- The NNTP was unable to capture demand patterns
- There was no difference or advantage in using 3 or 5 previous intervals.

The research also recommended that for future studies a large amount of data should be used and the data should be collected from a freeway with heavily traffic.

Kwon and Stephanedes (12) developed a BPN to forecast traffic volumes on on-ramps. The BPN used volumes from the previous intervals at the ramp in question, volumes from the previous interval the previous day, volumes at upstream ramps as well as volumes from upstream and downstream loops. The input and output variables used in this model can be seen in Table 2.2. The results of the BPN were compared against the UTCS-2 and an adaptive parameter

predictor using a Kalman Filter. It was found that using the same amount of historical data, the BPN requires less time and effort and provides comparable results to the adaptive parameter predictor. The BPN was found to have a mean absolute error range (MAE) between 3.7 to 9.1, while the adaptive predictor had a MAE range of 3.0 to 11.4. Both of these models outperformed the UTCS-2. (12) Kwon and Stephanedes (12) also found that the BPN is less adaptive to demand fluctuations than the adaptive parameter predictor because the BPN is not reflected in the prediction in the next interval unless the network is retrained with the new data.

**Table 2.2 Input-Output Specifications for BPN Predictor (Kwon and Stephanedes)**

		Ramp in question	Upstream 3 entrance ramps 3 exit ramps	Upstream 3 mainline locations	Downstream 2 mainline locations
Input	Current Day	Volume at t-1, cumulative volume at t, t-1, t-2	Volume at t-1, cumulative volume at t, t-1, t-2	Volume at t-1	Volume at t-1
Input	Previous day	Volume at t-1, cumulative volume at t, t-1, t-2	Volume at t-1, cumulative volume at t, t-1, t-2	Volume at t, t-1, cumulative volume at t	Volume at t, t-1, cumulative volume at t
Output	Current Day	Volume at t			

Park, et al. (13) presented a study involving the application a radial basis function (RBF) neural network to forecast short-term freeway traffic volumes. RBF requires less computational time than back propagation network (BPN) model does. For simplicity, only downstream traffic volumes were used. For comparison with the RBF model, a Taylor series model, an exponential smoothing model, a double exponential smoothing model, and two BPN models were developed. Of the two BPN models, one utilized downstream station traffic volume data, while the other processed both upstream and downstream traffic data. The RBF model trained in 20-30 minutes, while the BPN took two to three hours. All of the traffic prediction models were evaluated using the same data from two different sites. The results from the BPN model, including both upstream

and downstream data, had the lowest mean absolute percentage error (MAPE) at Site 1, while the RBF model had the lowest MAPE at Site 2.

In summary, most of the NNTP models are for predicting traffic volume. Usually, traffic data for more than three previous time intervals are considered for input. Some applications only considered one location which was their concern, while others also combined data from locations which are viewed as influential to prediction.

## **2.4 Detection Algorithms**

Electronic sensors use a detection algorithm or decision-making algorithm to distinguish between normal and incident conditions. There are two categories of incident detection algorithms: single and double detector algorithms.

As defined in Ahmed and Cook (14), single detector algorithms (or smoothing algorithms) are those which employ short-term forecasts of the state variables and a set of calibrated thresholds to detect the sudden perturbations in traffic stream behavior generated by incidents. The state variables employed in these algorithms are usually obtained from a single detector that is upstream with respect to possible occurring incidents. The design of the algorithms is based on the notion that recurrent congestion can take more time to accumulate than non-recurrent congestion. Therefore, to capture the temporal trend of traffic variables, prediction models such as time series models have been frequently employed. Existing algorithms of this category include Double Exponential Moving Average (DEMA) algorithm (15), the Standard Normalized Deviation (SND) algorithm (16), the Autoregressive Integrated Moving Average (ARIMA) algorithm (14), the McMaster algorithm (17), and the combined

fuzzy logic, artificial neural network (ANN) algorithm (18), and the Cumulative Sum (CUSUM) algorithm (19).

The second category of algorithms is those which attempt to distinguish between incident and incident-free conditions as measured by differences in upstream and downstream traffic conditions. Thus, traffic variables are obtained from two consecutive detectors to capture possible spatial differences. Typical algorithms belonging to this category are the California algorithm, the low pass filtering algorithm (20), and the artificial neural network (ANN) algorithm (21). There are also some ANN algorithms such as Ritchie and Cheu (22), and Stephanedes and Liu (23) which incorporate both temporal and spatial variables.

It should be noted that there is no documented result to show, a priori, which category of algorithms is better. The underlying reason is that the traffic processes involved in incidents are too complicated to be simply described by one or two experiencing rules used in detection algorithms. Most researchers expect double detector algorithms to have better performance than the single detector algorithms. As usually assumed in double detector algorithms, upstream occupancies should increase while downstream occupancies should decrease after occurrence of an incident. This assumption may not be true in some cases. As pointed out in Cook and Cleveland (15), there are some exceptions (7 out of 50 cases) where downstream volume increases rather than decreases. In these cases, the double detector algorithm will fail to detect the incidents.

Detection algorithms function using three types of data; speed, volume, and occupancy. Table 2.3 illustrates the data required for the various types of detection algorithms. The table shows that the value of occupancy is heavily utilized for most detection algorithms.

**Table 2.3 Incident Detection Algorithm Data Requirements (Black )**

Algorithm	Occupancy	Volume	Speed
California Basic	Required		
California #8	Required		
Standard Normal Deviate	Required		
Exponential Smoothing	Required		
Modified McMaster	Required	Required	
Time Series ARIMA	Required	Required	
Low Pass Filter	Required		
Neural Network	Required	Required	Required

To gain an understanding of how an incident detection algorithm works, an overview of the CUSUM algorithm (19) is provided. The decision function of this algorithm can be written as:

$$g_k = \frac{m}{2\hat{\sigma}_0^2} - [v - (\hat{m}_t - \hat{m}_0)]^2 + (\hat{m}_t - \hat{m}_0)^2, \quad (1.1)$$

where  $g_k$  = the threshold,

$m$  = the number of occupancies used,

$v$  = the difference between the means of errors,

$\sigma$  = the estimated standard deviation in normal conditions

$\mu$  = the occupancy.

The decision function is driven by statistically evaluating the difference in the occupancy values. Once this difference is obtained, it is then compared against the threshold,  $g_k$ . If the difference is greater than the threshold, there is an incident. If the difference is smaller than the threshold, traffic is in a free flow condition.

The common criteria used to evaluate algorithm performance are detection rate, false alarm rate, and detection time. The false alarm rate conveys the percentage of time intervals that were identified as incidents and those that were not. The detection rate refers to the percent of

incidents that were detected. These deficiencies can be traced to the decision function and the threshold value. Detection time is defined by the amount of time required for the algorithm to detect that an incident has occurred.

The hardware for detecting incidents currently outperforms the available software. The deficiencies associated with detection algorithms are low detection rates, high false alarm rates, and long detection times. The California PATH Research program recently compiled an evaluation of various detection algorithms (24). Unfortunately, the algorithms were not all evaluated using the same data, therefore the results from this study can only provide a rough estimate of the capability of each algorithm. A portion of the results of this evaluation can be seen in Table 2.4.

**Table 2.4 Estimated Performance Results of Different Detection Algorithms**

**(Source PATH)**

Algorithm	Detection Rate (%)	False Alarm Rate (%)	Avg Detection Time (minutes)
Modified McMaster	68	0.01	2.2
SND	92	1.30	1.1
California Basic	82	1.73	0.85
California #8	68	0.18	3.04
ARIMA	100	1.50	0.40
Bayesian	100	0	3.9

Table 2.4 shows how the different algorithms perform. It can be seen that there does not seem to be a perfect algorithm. For instance, the Bayesian algorithm has a 100% detection rate and a false alarm rate of 0%, however, it takes almost four minutes to detect an incident. The ARIMA algorithm also has a detection rate of 100% and a low detection time of only 0.4 minutes. Unfortunately, it has a high false alarm rate of 1.5%. It is clear from Table 2.4 that there are definite tradeoffs involved when trying to optimize one specific criterion.

The decision function is the part of the algorithm that evaluates the current traffic conditions. It typically employs a traffic prediction model to forecast what the traffic conditions will be in the next time interval. The forecasted values are generated by a prediction model that can be based on historical data or a moving average of real time data. These predicted values are then compared to the actual conditions. The difference between the two values is statistically evaluated (different statistical methods are used for each algorithm) and then compared to the threshold.

The threshold is the value with which a current condition is compared to decide whether or not there is an incident. If the threshold is “low,” the detection rate will be “high,” and the algorithm could generate many of false alarms. If the threshold is “high,” the false alarm rate will be relatively “low,” but the algorithm will not detect as many incidents. To improve the algorithm performance, a precise decision function must be created to extract the best performance of the threshold value.

The threshold is normally a fixed value based on the amount of variation that the decision function may experience without being an incident. Improving the traffic prediction model to reflect values that are closer to current conditions will decrease the amount of error generated in the decision function. This will then improve the tolerance of the threshold. By accomplishing these tasks, the overall performance of the algorithm could be improved.

An interesting observation of recent studies on models employed for traffic prediction and incident detection is that neural networks have been applied frequently. Neural networks are extremely useful for solving problems that have complex relationships, are non-linear, have a high degree of uncertainty, and require classification, all of which are characteristics of the incident detection and traffic prediction problem.

## **2.5 Neural Network Incident Detection Models**

Chang (11) was one of the first to apply neural network models to freeway incident detection. Two approaches were employed. In the first approach, a neural network model was developed based on traffic volume from several detectors. The second approach integrated two neural network models, one of which filtered out abnormal information, and the other was for incident detection. Unfortunately, the details of the input and output pair for training these neural network models were not provided.

Neural network models applied by Dougherty, et al. (8) classify congestion states, where no distinction was made as to either recurrent or non-recurrent congestion. The inputs consisted of one, two, or all three of current congestion parameters from 20 neighboring detector sites: vehicle flow, queue length, and occupancy. The number of previous intervals was not specified for the data. The output was a binary variable indicating whether the sub-region was congested or uncongested. The data sets used for model development were from a traffic control system.

Ritchie and Cheu (22) also developed their neural network model based on simulation data. The input pattern included upstream occupancy and volume at the current and the previous two-time intervals and downstream occupancy and volume at current and previous four time intervals. Extending the work of Ritchie and Cheu (22), Cheu and Ritchie (25) again tried two different neural network models: the self-organizing feature map and adaptive resonance theory. While simulation data was again used for algorithm development, field data was used for testing.

In Hsiao, et al. (18), a neural network model was used in combination with fuzzy logic. Through neural network training, the membership function for each traffic variable was determined and the decision rule for incident detection was also derived in a form of a numerical

matrix. Three traffic variables are used: speed, flow, and occupancy only for the current time interval from a single detector location.

Based on actual traffic data, Stephanedes and Liu (23) applied the back propagation neural network model. The inputs included volume and occupancy from upstream and downstream detectors for the current and nine previous time intervals.

Dia and Rose (21) discussed a multi-layer feed-forward (MLF) neural network incident detection model that was developed and evaluated using field data. The input of training pattern pairs consisted of speed, flow, and occupancy obtained at only the current interval for two consecutive detectors.

In general, most of the NNID models utilize data from multiple locations, i.e., the upstream and downstream detector stations. However, there are applications that only used data from a single location. The predominant traffic variable used for incident detection algorithms is occupancy, although volume and speed were also used in some applications. Some studies also evaluated information from previous time intervals. The ultimate output from NNID algorithms is an indicator of a predicted traffic state. The direct output, however, is a real value, which must be used by a threshold value to derive traffic states.

The literature review has shown that neural networks have been applied to traffic prediction models and incident detection algorithms, and have enhanced the performance of both systems. However, no research has been conducted that integrated a neural network traffic prediction model into an incident detection algorithm. The existing neural network traffic prediction models have not been constructed using the variable of time, although research conducted by Williams (11) recommend that a time variable be incorporated. The objective of this research will be to improve the overall performance of the incident detection algorithm.

## Chapter 3

### Methodology

The purpose of this study is to investigate the effects of integrating a neural network traffic prediction (NNTP) model into an incident detection algorithm. It is speculated that a NNTP can improve the performance of an incident detection algorithm. This study will be broken into three phases, data collection, construction of the NNTP, and integration of the NNTP into the detection algorithm.

#### 3.1 Traffic Data

This study requires a large quantity of freeway traffic data to train the NNTP models and evaluate the detection algorithms. Although traffic data can be created by using a simulator that generates any conditions the programmer inputs, real freeway data are more desirable. Real traffic data are desirable because they are an accurate representation of what the system would undergo in a real world application. They also lend more credibility to the research. These real data will need to come from a freeway with a high average daily traffic (ADT). The high ADT is desirable because heavier traffic increases the occurrence and frequency of incidents. The high volume also makes an incident more pronounced and easier to detect.

Data collected from an outside source is sought because:

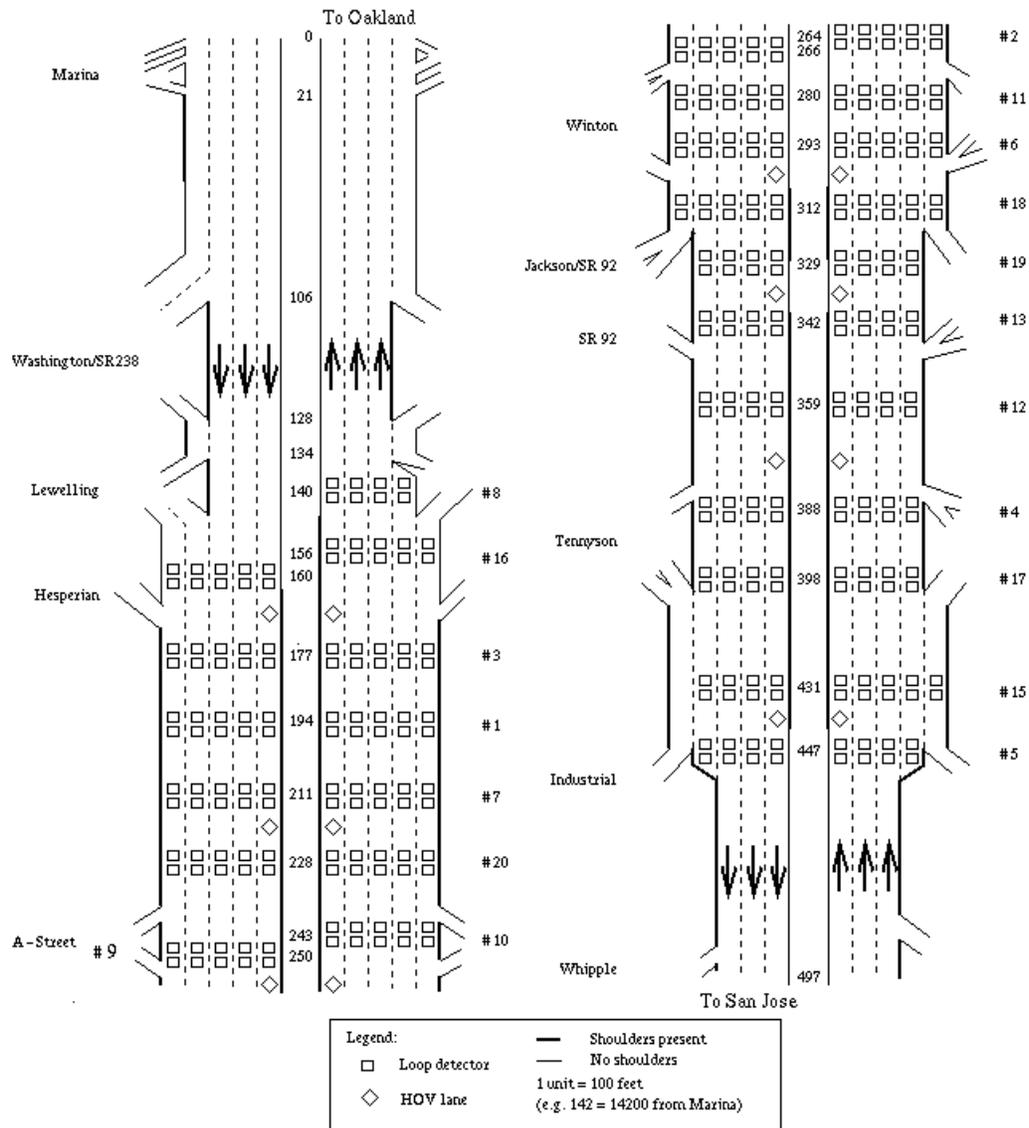
- A freeway section with the high ADT needed does not exist in West Virginia
- The collection of such a large data base would require a tremendous amount of time
- The resources required to collect and filter the raw data are unavailable.

Once these data are obtained, they are filtered if necessary, and formatted for use by the NNTP and the incident detection algorithm.

The data used in this study was collected by the California PATH Program during the Freeway Service Patrol (FSP) project (28). The FSP project was an incident management program that was conducted on freeway I-880 in the city of Hayward, Alameda County, California. The data was collected in two periods: February 16 through March 19, 1993 and September 27 through October 29, 1993. The FSP project produced the largest, most comprehensive, and up to date freeway operations database. This site has one of the highest frequencies of incidents in the San Francisco Bay Area. The data are available on the Internet for use by the research community and have been used for various studies (26, 18).

The freeway section is 9.2 miles long and varies from 3 to 5 lanes with an HOV lane covering approximately 3.5 miles of the study section. Several of the sections lacked right-hand shoulders and/or left-hand shoulders. The ADT ranged between 160,000 to 200,000 vpd (24). The layout of the monitored section of I-880 can be seen in Figure 3.1.

Field observations and loop detectors are used for obtaining information about freeway traffic conditions. The field observations were made by probe vehicles, California Highway Patrol (CHP), tow-truck logbooks, call boxes, and motorists using



**Figure 3.1 Map of I-880 in Hayword, CA (Source Petty)**

cellular phones. The loop detectors are placed in pairs on the main lanes, and located in approximately 0.33-mi intervals. There are single detectors imbedded in the on and off ramps. Loop detector data was collected for two peak periods: 5:00 to 10:00 AM and from 2:00 to 8:00 PM (28).

A total of 2,181 incidents were recorded during the study period. There were 1,210 incidents in the March study and 971 incidents in the September study. The estimated

incident rate for this section of I-880 was 104 incidents per million vehicle-miles, which is consistent with the national average incident rate. (27). As seen in Table 3.1, accidents accounted for 10 percent of the total incidents while vehicle breakdowns accounted for almost 89 percent.

**TABLE 3.1 Incident Classification** (Source Skabardonis 1997)

INCIDENT TYPE	LOCATION			TOTAL	%
	Median	In-Lane	Rt. Shoulder		
Accident	29	35	103	167	10.3
Breakdown	60	24	1347	1431	88.6
Debris/Pedestrian	1	15	2	18	1
<b>Total</b>	<b>90</b>	<b>74</b>	<b>1452</b>	<b>1616</b>	<b>100</b>

This study utilizes the data that was well formatted by PATH for an international transportation symposium incident detection contest held in July of 1998. These data include 15 weekdays from March 1, 1993 to March 19, 1993. In total, 197 incidents were identified in the studied freeway area during the March time period.

This study uses three consecutive segments in each direction. The segments will be between loop detectors #3 and #20. These segments are to eliminate interference from the on/off ramps. Roadway geometry is believed to affect traffic patterns during incidents (29). Since various detection algorithms process traffic patterns differently, it became important in this study to select a homogeneous section of roadway. This allows the algorithm development and evaluation to be based on a common set of traffic patterns.

The loop detectors provide speed, volume, and occupancy data at 30-second intervals. In this study, however, only occupancy data is utilized, because it is the prevalent variable used in incident detection algorithms. It is understood that an incident would cause the upstream occupancy to increase; therefore, only occupancies from the upstream detector

station were used. Specifically, only occupancy data from detectors #20, #7 and # 1 were used for northbound traffic, and those from #3, #1 and # 7 for southbound traffic.

### **3.2 Neural Network Traffic Prediction**

The NNTP models will be developed using the above mentioned data. Most of the data will be used for training purposes; a small portion will be reserved for testing. The first step of developing the NNTP will be to determine what the desired output should be. The appropriate inputs will be configured to predict the desired output. Three different NNTP models will be developed to evaluate different configurations of inputs to get the desired output. Initially, test models will be constructed to evaluate different training configurations such as network structure, learning rate, and training time. Once these factors are “optimized,” the final training of the models will be done. The performance of these models will then be evaluated on the root square mean error and the training curves.

The data from the FSP project was used to develop and train the NNTP. The initial decision in the development process of the NNTP was to determine the inputs and outputs of the model. The literature review reveals that many detection algorithms employ occupancy. This fact coupled with the ample supply of occupancy data from the FSP project determined that the output from the NNTP should be occupancy. Occupancy will also be used as one of the inputs. The literature review also reveals that none of the existing NNTP models explicitly considered a time component. However, Williams (4) strongly recommends that the cyclic pattern be modeled. A cyclic pattern shows that traffic patterns occur on a time cycle. Obviously, prediction can be improved at places where a change in traffic variables occurs when conditions change to and from normal and peak conditions by incorporating a time component. This change in flow conditions is an area where many of false alarms are

generated. So, by improving the prediction model the detection model should improve as well.

Three models are developed using time and/or occupancy to predict occupancy. The first model uses time (t) as the only input. The second model uses only occupancy (o) for the input. This model uses the occupancies from the previous three intervals to predict occupancy in the next interval. The third model incorporates time and occupancy at the three previous intervals as well as the time of the next interval to predict the next occupancy. Note that only one location is considered here. Examples of how this data looks during training of the three NNTP models (time only, occupancy only, and time and occupancy) can be seen in Tables 3.2, 3.3, and 3.4. The first column(s) are the inputs for the training. The last column in the tables contains the pattern values, which provide the model a comparison to see if the model is outputting correct answers.

**Table 3.2 Sample Data Sheet for time only method**

Time (t)	Occupancy (o)
5:00:30	1.6
5:01:00	1.4
5:01:30	2.1
5:02:00	0.8
5:02:30	1

**Table 3.3 Sample Data Sheet for occupancy only method using occupancy from the past three intervals**

Occupancy			
o-3	o-2	o-1	o
2.6	1.6	1.3	1
1.6	1.3	1	1.5
1.3	1	1.5	1.6
1	1.5	1.6	2.6
1.5	1.6	2.6	1.7

**Table 3.4 Sample Data Sheet for time and occupancy method using time and occupancy from the past three intervals and time of next interval to predict the next occupancy**

Time	Occ	Time	Occ	Time	Occ	Time	Occ
t-3	o-3	t-2	o-2	t-1	o-1	t	o
5:00:30	2.7	5:01:00	2.2	5:01:30	2.3	5:02:00	1.9
5:01:00	2.2	5:01:30	2.3	5:02:00	1.9	5:02:30	3.6
5:01:30	2.3	5:02:00	1.9	5:02:30	3.6	5:03:00	1
5:02:00	1.9	5:02:30	3.6	5:03:00	1	5:03:30	2.1
5:02:30	3.6	5:03:00	1	5:03:30	2.1	5:04:00	2.3

Training data sets, as shown in Tables 3.2, 3.3, and 3.4, were compiled and edited for each occupancy prediction model given the specifications of input components. While compiling the combined training sets, traffic characteristics were first investigated for southbound and northbound lanes respectively. In the investigation, traffic data for each location was filtered to exclude incident data. Any days' data that was not incident-free was deleted. Each data set contained ten days worth of data. The mean and the standard deviation of the occupancies were calculated for each time interval of the ten-day incident free data sets. Figures 3.2 and 3.3 present the values of the mean and standard deviation respectively for each direction. The difference in means appear to begin at time 14:35, while the difference in standard deviation begins at 14:20. The standard deviation shows a particularly dramatic change when the time is close to peak time. With the existence of such a difference, a decision was made to train the neural network model for these two directions separately. The training data for each direction is from those three corresponding upstream detector locations. It is assumed here that the traffic patterns reflected in these three locations are similar because there are no ramps involved in these three freeway segments.

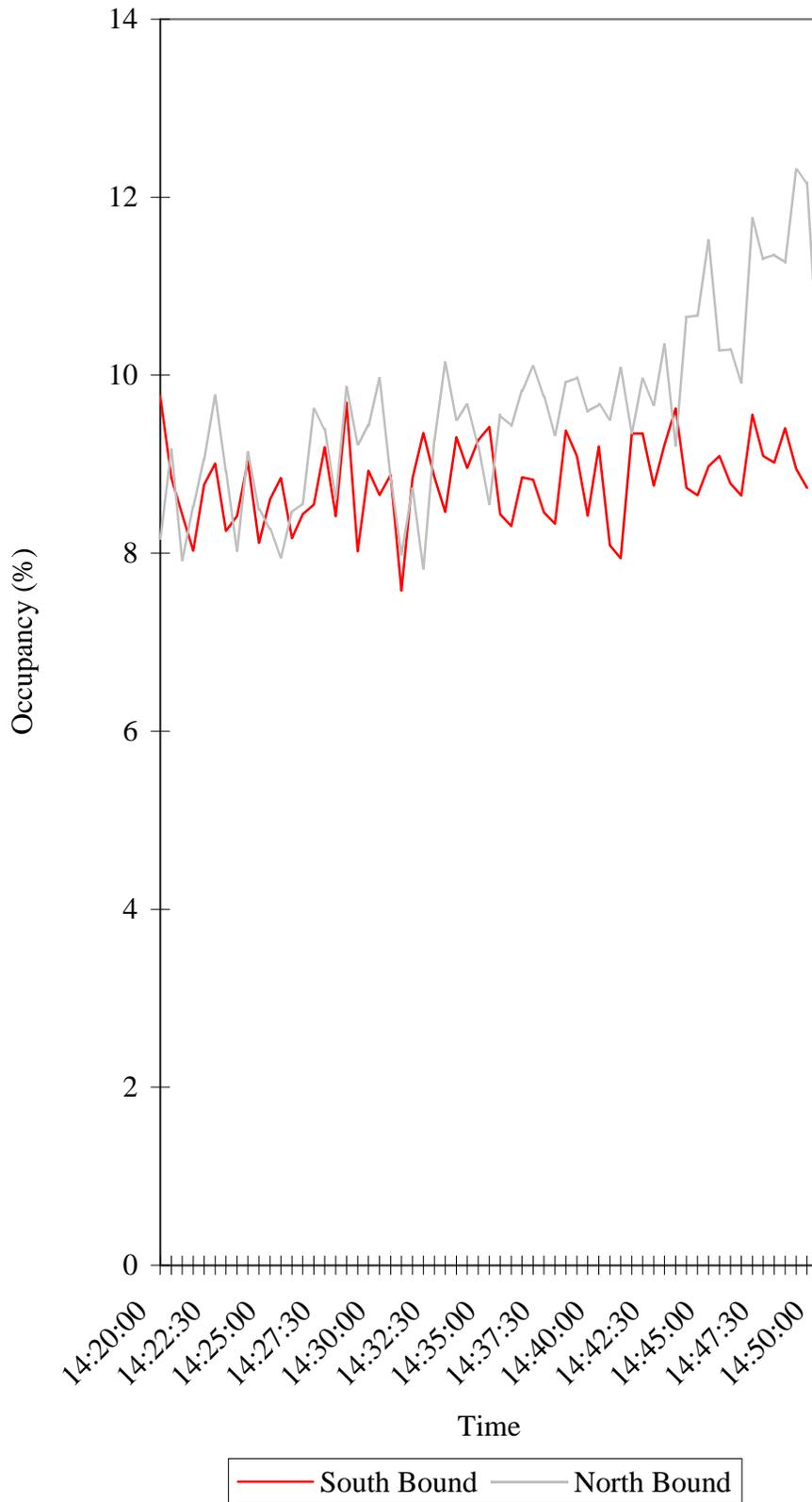


Figure 3.2 Mean of the Occupancy Data

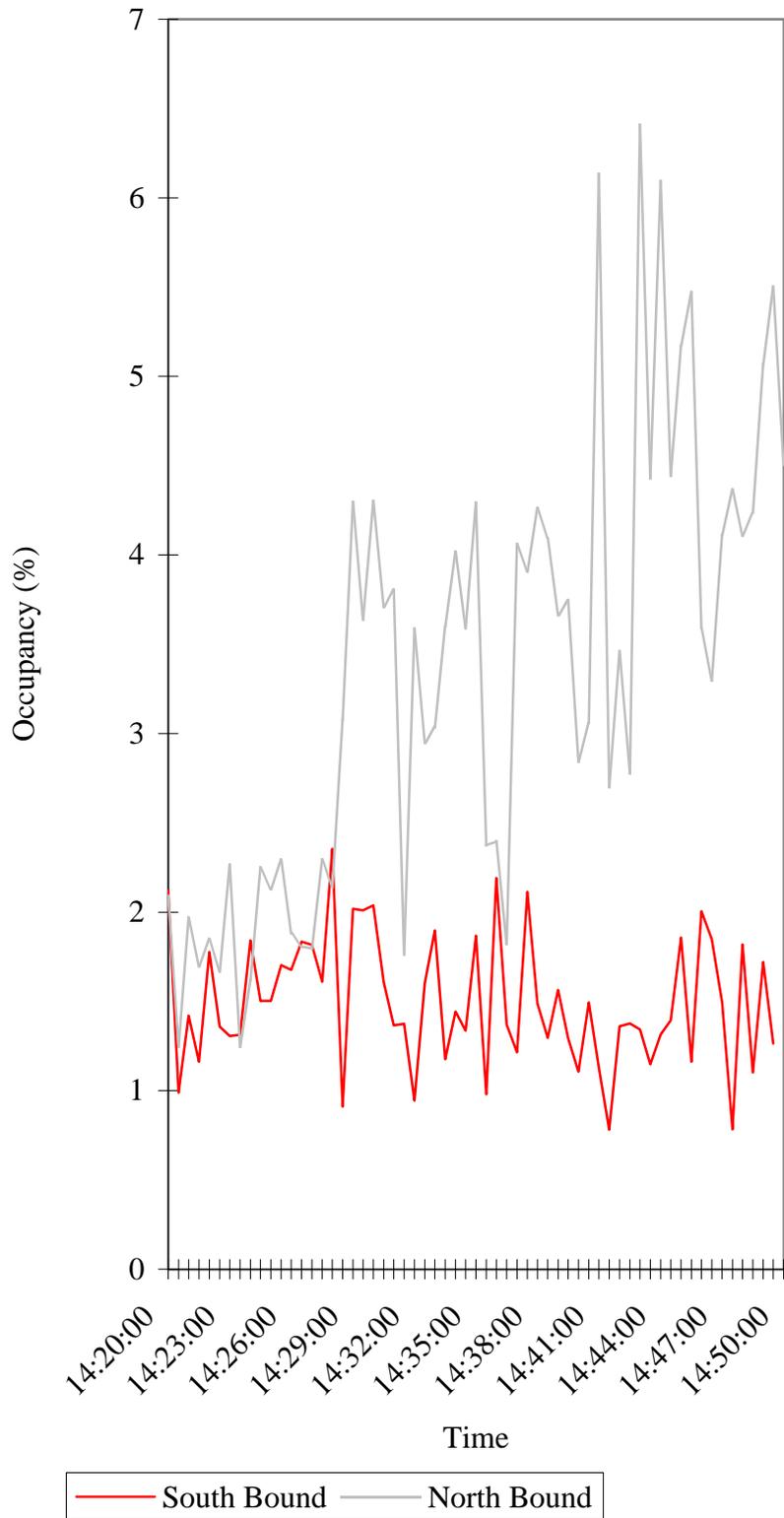


Figure 3.3 Standard Deviation of the Occupancy Data

BrainMaker software was used to create the neural networks. BrainMaker is one of the most popular neural network software packages and has been used in other traffic prediction research (25,30). The first step in training the neural networks was to import the training data from a spreadsheet into NetMaker, which is software that creates a BrainMaker data file that can eventually be trained into a neural network. While in NetMaker, the inputs and patterns are selected. The pattern provides the network with an example of what the output of the network should look like. NetMaker takes 90 percent of the file for training the network; the other ten percent is used for testing. The file is then ready to be placed into BrainMaker.

Neural networks are created by learning the patterns and associations of the training data. The network learns by processing a run of the training data and then propagates back through the network to make corrections to the network structure. A run consists of the network evaluating the training data once and comparing each output to the parameters. The network keeps a running tab of correct and incorrect output and then computes a RMSE. The network learns as it completes each run, and creates a weighted matrix, which gives the best performance for varying structures. As the runs progress they eventually converge to lower values of RMSE.

There are many different parameters that can affect how well the network is trained. The three parameters that were altered in this study were; network size and structure, learning rate, and amount of runs used to train the network.

The size and structure of a neural network are made up of inputs, hidden layers, neurons per hidden layer, and outputs. The numbers of inputs and outputs have been predetermined. However, the hidden layers, and neurons per hidden layer determine how

many connections will be made. This is important because it determines how the network will learn. If there are too few connections, the network may memorize the data. If there are too many connections, the network will have difficulty identifying patterns and associations and be unable to learn properly. Although as many as three hidden layers were tried, it was found that one hidden layer provided the lowest RSMR. The number of neurons per hidden layer varied between 2 and 20. Two neurons per layer provided the lowest RSME with the time only method, while 20 neurons had the lowest RSME for the method with time and occupancy. A diagram of the occupancy only NNTP model can be seen in Figure 3.4.

The learning rate determines the speed at which the network is trained. A high rate of learning will train the network in a shorter time, but the network may converge prematurely, leaving a network that generates a greater number of errors. If the learning rate is too slow, the training time is elongated, and the network may never converge. Different learning rates were tried at the beginning of the study, however the learning rate of one, provided the best results and was used for all final training.

Eventually the networks converge to a point where no further progress in learning is made. At the beginning of this study, the networks were trained to 1500 runs. It was found that the networks would converge at about 100 runs. To build in a factor of safety and a standard process, all of the networks were trained to 150 runs.

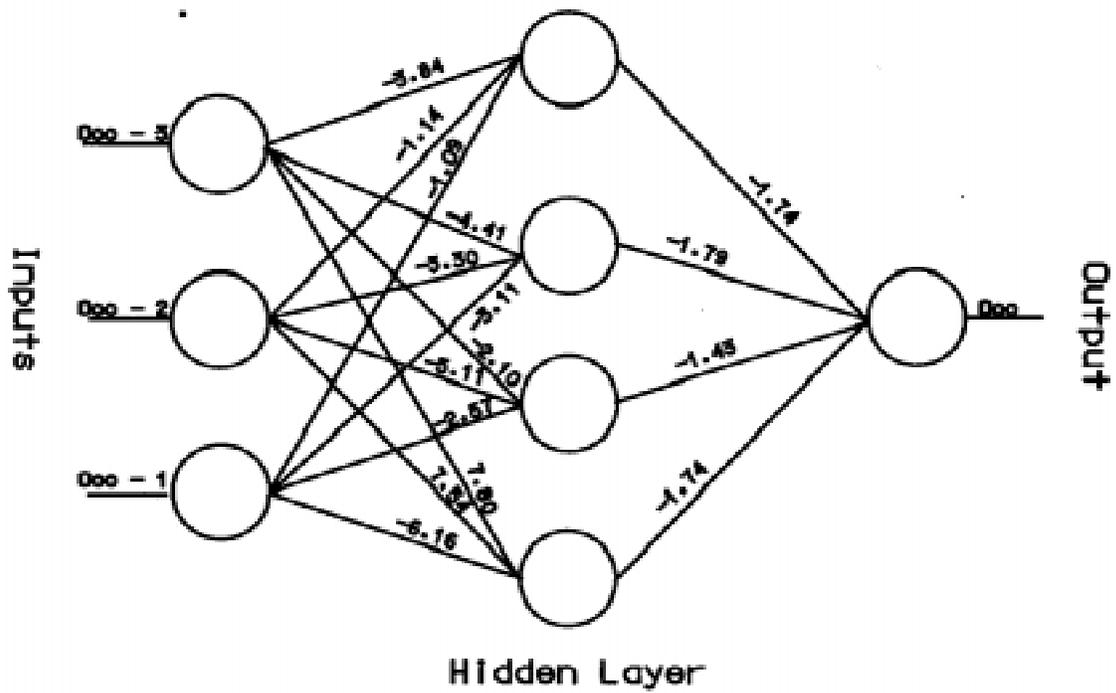


Figure 3.4 Diagram of the Occupancy Only NNTP Model

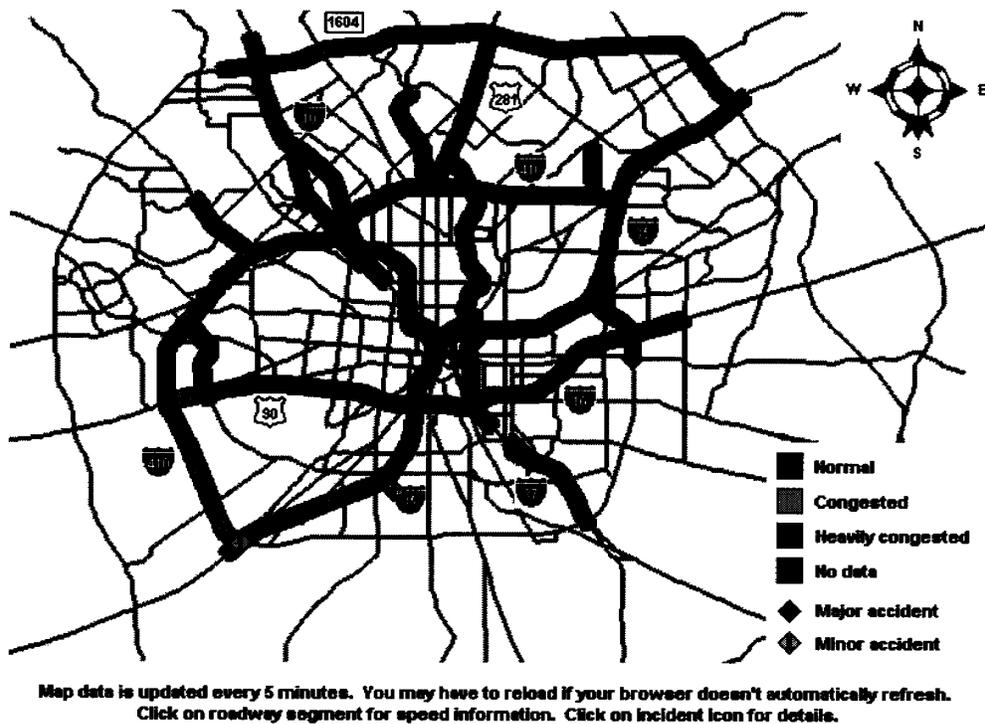
### **3.3 Incident Detection Algorithm**

After the NNTP models are built, they will then be ready to be integrated into the incident detection algorithm. The first step in this process will be to select an appropriate algorithm for testing. The ideal algorithm is one that has been adopted by a traffic management center, in addition to being used in current research. These criteria are desirable to show that this research could possibly have future real-world applications, as well as staying consistent and up-to-date with the research community. Although these criteria are desired, the primary criteria for the algorithm is to be “compatible” with the NNTP model. Specifically, the output from the NNTP should be used as part of the decision making function of the algorithm. Once the detection algorithm has been selected, each of the three NNTP models will be integrated. This will be done by replacing the variable in the decision function that currently is based on a source such as a historical database with a value from the output of the NNTP. The chosen detection algorithm will be run with each of the three NNTP models as well as in the original form. The original model of the algorithm will serve as the control for this study. The results will be evaluated by looking at the detection and false alarm rates.

Incident detection algorithms usually have a traffic prediction model incorporated into the system. The purpose of the traffic prediction model, such as time series, historical data based, or real time, is to estimate the current traffic conditions and provide a reference for detection. The Standard Normal Deviate (SND) algorithm (16) is a time series based detection algorithm that has been frequently employed (24) for actual traffic management centers and other research. The most noted application of the SND algorithm is the TransGuide Traffic Management Center in San Antonio (31), Texas. The SND algorithm is

versatile and can be configured to predict speed, energy, volume, and occupancy. Since the SND algorithm has been widely used and can detect incidents based on occupancy, it was chosen as the model to be retrofitted with the neural network traffic prediction models.

The TransGuide system in San Antonio is a state-of-the-art traffic management center that started in 1993 to manage 26 miles of freeway around San Antonio. It is currently being expanded to manage 191 miles of primary and secondary roads. The system generates a traffic map that can be viewed on the Internet at <http://www.transguide.dot.state.tx.us/map/>. An example of this map can be seen in Figure 3.5. The core of the incident detection part of the TransGuide system is the SND algorithm. The TransGuide system uses speed and occupancy to detect incidents. The system sets major and minor alarm values for each variable. The threshold values for occupancy are major and minor alarm values of 35 percent and 25 percent, respectively. The thresholds can be updated every 15 minutes based on values derived from historical data. This system boasts a 15-second detection time. Unfortunately, the false alarm and detection rates are unavailable (32).



**Figure 3.5 The TransGuide Map Illustrates a Visual Representation of the Data Collected by the TransGuide System**

The SND algorithm is a time series model that was developed by Dudek and Messer. The model is based on the traffic variable  $y_t$ , which can be expressed as follows:

$$y_t = \bar{y}_t + \delta_t, \quad (3.1)$$

this indicates that a traffic variable can be predicted by its theoretical mean  $\bar{y}_t$ . This prediction has theoretical error  $\delta_t$ , which is assumed to have a mean zero and variance  $\sigma$ .

The decision rule derived from this equation can be written as:

$$\frac{y_t - \bar{y}_t}{\hat{\sigma}} \geq h \quad (3.2)$$

where  $h$  is the threshold for decision making and  $y_t$  is the current occupancy value from the detector.

The decision function of the SND algorithm takes the same form as Equation (3.2).

The original formula for calculating  $\hat{y}_t$  in the SND algorithm is:

$$\hat{y}_t = \frac{\sum_{i=t-m}^{t-1} y_i}{m}. \quad (3.3)$$

Integrating the NNTP into the detection was done using a similar decision function as Equation (3.2), which can be written as:

$$\frac{y_t - \hat{y}_t}{\hat{s}_t} \geq h \quad (3.4)$$

where  $\hat{y}_t$  and  $\hat{s}_t$  are estimated parameters.  $\hat{y}_t$  is obtained from the trained neural network models given occupancies in previous time intervals. The formula for calculating  $\hat{s}_t$  is:

$$\hat{s}_t = \sqrt{\frac{\sum_{i=t-m}^{t-1} (y_i - \hat{y}_i)^2}{m-1}}, \quad (3.5)$$

where  $m$  denotes the number of previous traffic occupancies used for prediction.

The decision function in Equation (3.4) is derived based on the assumption that the mean of the errors is zero. Incident conditions alter the flow of traffic such that the mean of the errors may not be zero anymore. This may also change the variance of errors. It can be expected that the average value of errors following an incident would become positive because the occupancies in incident conditions are usually greater than those in normal conditions. Therefore, to detect an incident it is necessary to detect whether there is an increase in the mean of errors.

## CHAPTER 4

### RESULTS

The results of this study are divided into two parts: performance of the NNTP models, and performance of the SND algorithm with and without the NNTP models. Section one will explore the outcome and performance of the three different NNTP models for traffic prediction. These results will then be incorporated into the incident detection algorithm. The results from the detection algorithms will be discussed in section two. The results of the incident detection algorithm will discuss the evaluation criteria, as well as the performance of the modified versions of the SND algorithm compared to the standard SND algorithm.

#### **4.1 Neural Network Traffic Prediction Model Results**

The NNTP models were trained using the formatted data from the FSP data using the neural network software BrainMaker. As the networks are trained, BrainMaker plots a learning curve, which consists of the Root Mean Squared Error (RMSE) versus the run number. The RMSE is calculated by the following equation:

$$RMSE = \sqrt{\frac{\sum(O-P)^2}{N}}$$

Where O = the output

P = the pattern

N = the number of facts

A network with a high RMSE value, i.e. close to 1.0, represents a poorly trained network. The goal is to obtain a NNTP with a RMSE that converges as close to zero as possible.

Figures 4.1 and 4.2 show the respective training curves for the north and southbound lanes. The time-and-occupancy model clearly outperforms the time-only and occupancy-only models in the southbound lane. The time-and-occupancy model had the lowest RMSE value of 0.0479. The occupancy-only and the time only, respectively, had RMSE values of 0.0502 and 0.0597. The results are not as clear in the northbound lane. In the northbound lane, the occupancy-only model had the lowest RMSE value of 0.0279, it was followed by the time-and-occupancy model with a RMSE of 0.0293, and the time-only model with an RMSE of 0.0882. These results show that the occupancy only and the time and occupancy models yield relatively equal performances, with the occupancy-only model performing best. The time-only model had the highest RMSE in both cases. This can be attributed to the dramatic variance in the occupancy in the northbound lane.

A sample of the accuracy of the different prediction models can be seen in Figure 4.3. This figure shows a sample of 30 minutes of predicted occupancy values from the three NNTP models compared to the actual occupancy data at 4p.m. in the southbound lane. The occupancy-only model accurately predicts the valleys in the graph, while the time and occupancy model seems to be better at predicting the peaks in the occupancy. Overall, the time and occupancy method seems to be the most accurate.

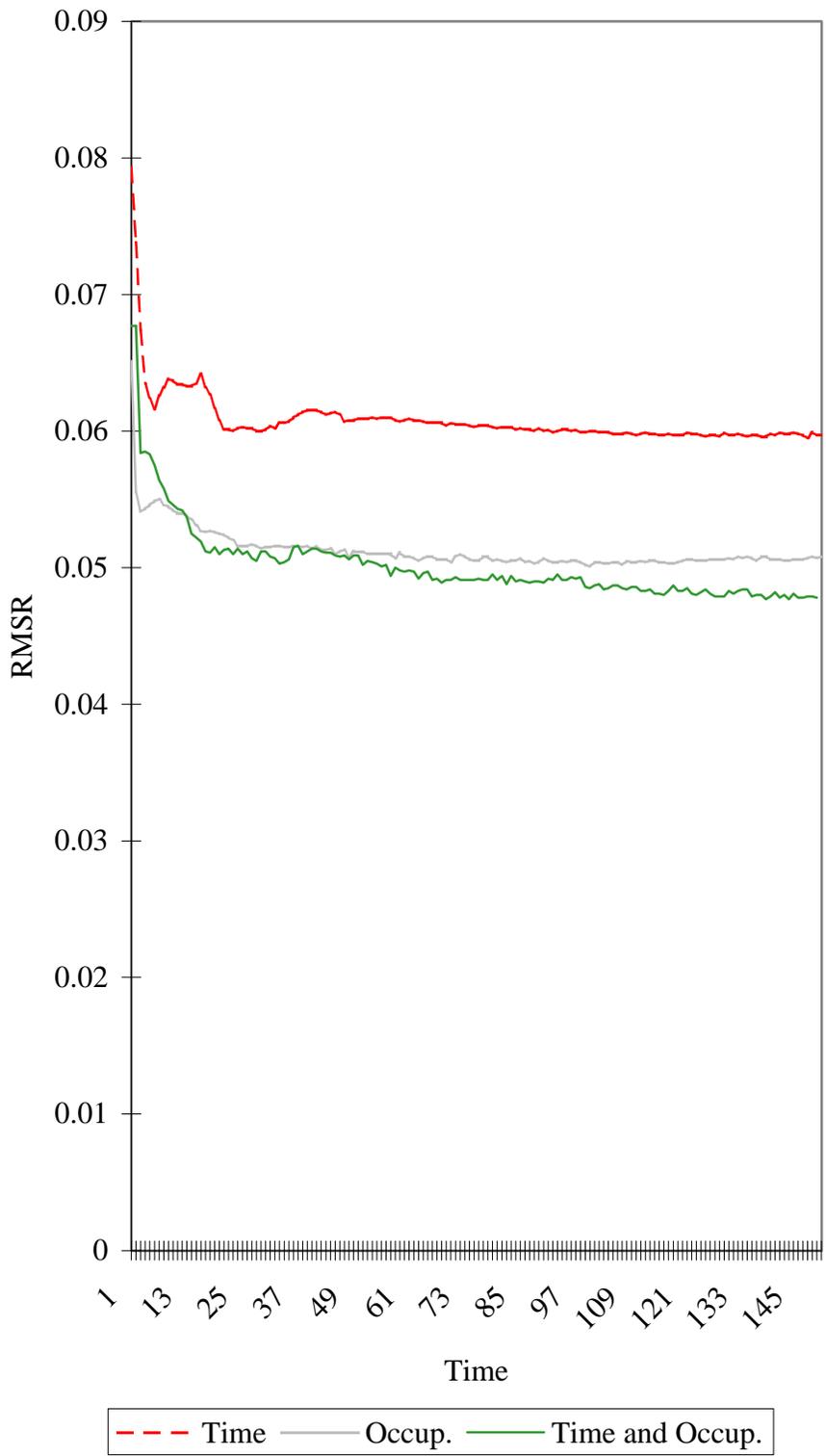


Figure 4.1 Training Curves for the Southbound NNTP Models

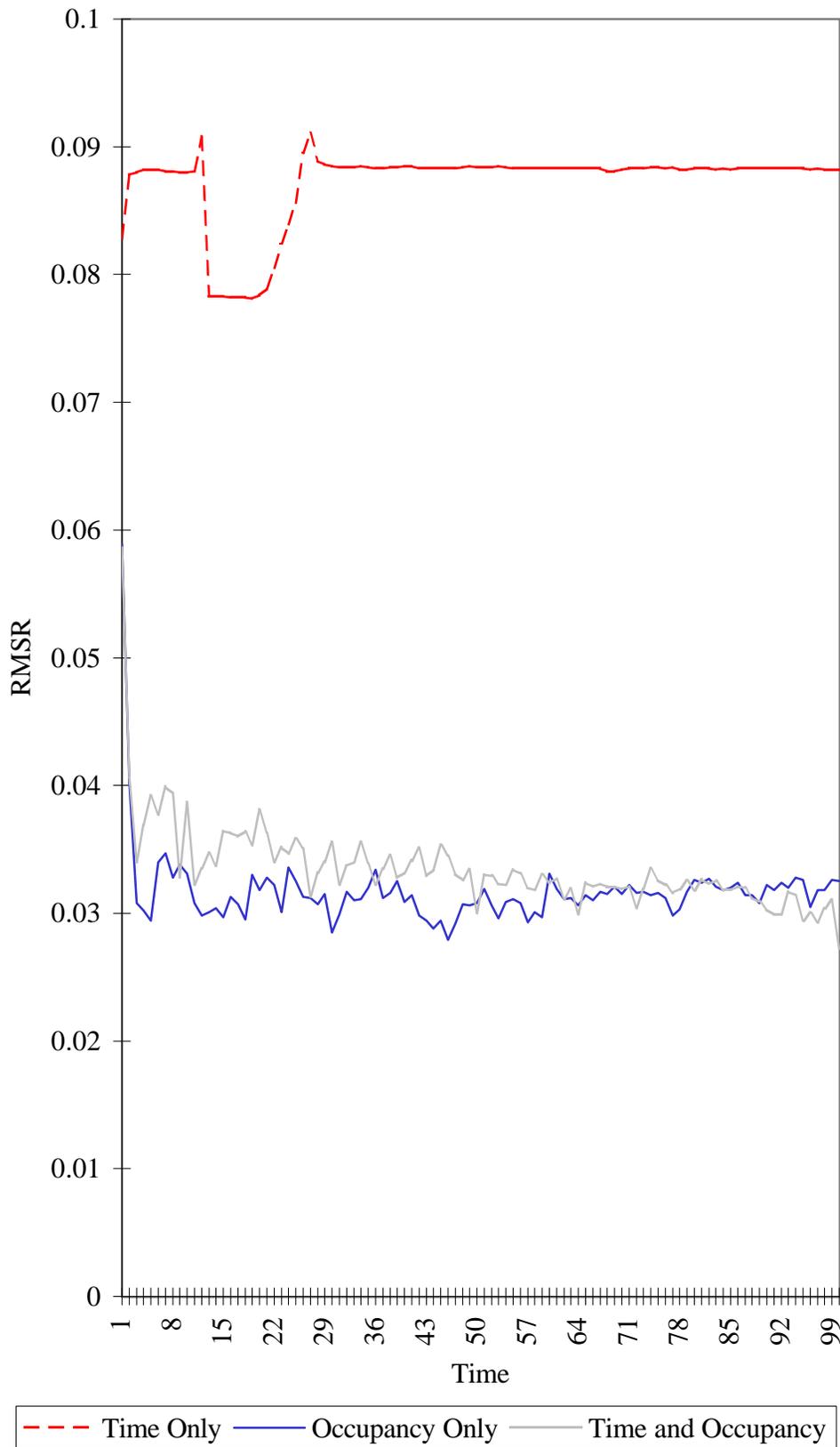


Figure 4.2 Training Curves for the Northbound NNTP Models

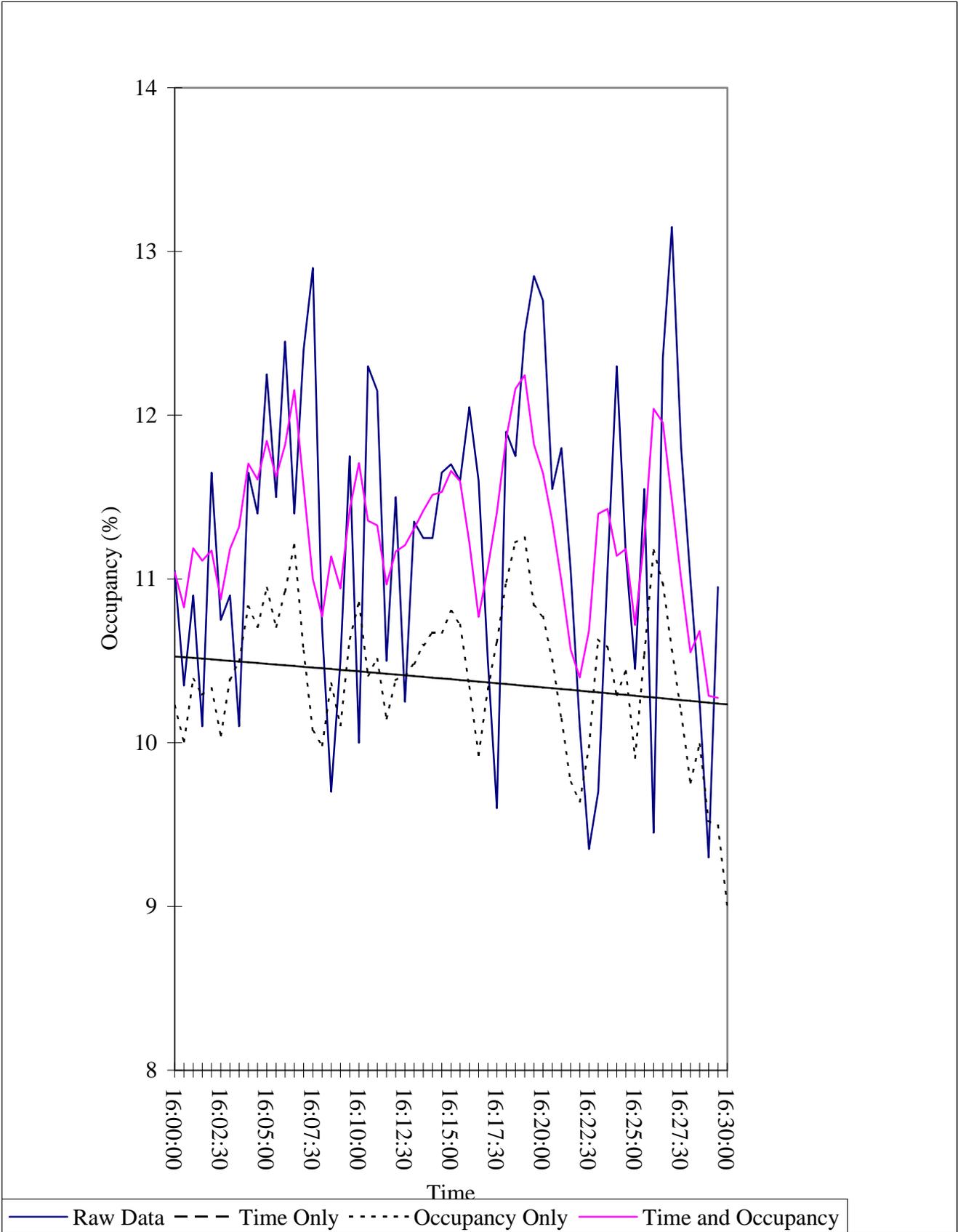


Figure 4.3 Comparison of Raw Occupancy Data Compared to the NNTP Model Output

The results from the northbound lanes provide evidence to support the contention that traffic prediction models which incorporate time components will not improve when traffic patterns vary substantially. A possible solution to adjust for varying traffic patterns may come from incorporating data from neighboring loop detectors. This essentially would be applying the principles of dual loop detection to make a dual loop predictor. The time component did make a significant improvement to the prediction model when the traffic patterns do not vary excessively.

#### **4.2 Incident Detection Algorithm Results**

All three of the NNTP models are then integrated into the SND algorithm. The logic behind this is that, although one NNTP model may outperform the others, as far as prediction is concerned, there is no guarantee that this performance will carry over into the detection algorithm. So by using all three NNTP models, a more thorough investigation can be conducted to see what type of prediction model applies best for incident detection.

The performance of incident detection algorithms is typically evaluated on two basic criteria, detection rate and false alarm rate. These values are then plotted and a graphical depiction of the performance and trade-off can be seen. Once these results have been graphed, conclusions about the system can be made as well as recommendations for future studies.

The detection rate is defined as the ratio of the number of incident detected over the actual number of incidents. The detection rate can be found by:

$$DR = \frac{\text{Number of Detected Incidents}}{\text{Total Number of Incidents}}. \quad (4.2)$$

There are two definitions of false alarm rate, on-line and off-line. The on-line definition of false alarm rate refers to the percentage of incident messages that were false alarms. This definition is not widely used throughout the research community because it does not accurately depict the amount of time consumed by the false alarms. The off line definition is the ratio between the number of time periods when an incident decision has been made under normal conditions versus the total number of time periods under normal conditions. The off-line definition can be written as:

$$FAR = \frac{\text{Number of Time Intervals in Normal Conditions Identified as Incidents}}{\text{Total Number of Time Intervals in Normal Conditions}}. \quad (4.3)$$

The off-line definition of false alarm rate will be used for this study to stay consistent with current research practices.

The detection and false alarm rates are very closely related. A high false alarm translates into lower sensitivity of the algorithm. This lowered sensitivity allows the detection algorithm to detect more incidents because the algorithm becomes less selective. Once the algorithm detects more incidents, the detection increases. The opposite occurs when the false alarm rate is low. If the false alarm rate is low, the algorithm will become more sensitive in distinguishing between free flow and incident traffic conditions. The detection algorithm will then in turn detect fewer incidents, and may even miss a few incidents. A conservative algorithm will have a low false alarm rate, but may not detect every incident. A more liberal algorithm will have a high false alarm rate, which will detect more incidents, but will require more time verifying whether there actually is an incident.

Once the performance criteria are selected, the testing of the incident detection algorithm began. The testing was conducted by sequentially running the occupancy data through the computer. The algorithm would stop and send out an alarm when the algorithm

value was equal to or exceeded the threshold value. The algorithm would restart after incident conditions were over. This occurred only on the incidents the algorithm actually detected.

The data that were used to evaluate the detection algorithm had incidents in them, and were not used for the training of the NNTP. When the testing of the detection algorithms was conducted, the threshold values were set between 2 and 6. This range was set so that a wide range of false alarm and detection rates could be generated. A threshold of two will have a high false alarm rate, a quick detection time, and a high detection rate. While a threshold of six will have a low false alarm rate, a slow detection time, and a low detection rate.

A master data sheet was compiled before running the tests. This master data sheet contained the number of incidents as well as the time and duration of each incident. Once the algorithm processed the data using the different configurations of the SND algorithm mentioned in Chapter 3, the output was evaluated and compared to the master data. The detection and false alarm rates were then calculated.

The performance results of the incident detection algorithms can be seen in Figure 4.4. This figure shows all three variations of the modified versions of the SND algorithm as well as the unmodified version of the SND algorithm. It can easily be seen that all of the modified versions of the SND algorithm outperform the original version. The next observation is that when the false alarm rate is less than 0.01, the time only NNTP version of the SND algorithm works the best. When the false alarm rate is greater than or equal to 0.01 the occupancy only NNTP version of the SND algorithm performs the best. Surprisingly, the time and occupancy NNTP version of the SND performed the worst of the three models. The time and occupancy NNTP version of the SND algorithm is actually

outperformed by the original version of the SND algorithm when the false alarm rate is greater than 0.03.

The observations from Figure 4.4 show that a good prediction model may not necessarily have a good incident detection capability. Figure 4.4 shows that the algorithm performs differently with the prediction approaches for different false alarm rate ranges. Although these three prediction approaches have different prediction accuracy, there are many reasons behind this phenomenon. One possible problem may result from the designed incident detection, which transform the prediction results for detection decision making. The quality of this transformation is sometimes difficult to be quantified as for the complexity of the detection decision function employed. The other possible problem is the temporal incident distribution, which may not be coincident with the time period when prediction has a good performance. In such a case, the advantage of a prediction model might not be realized.

Moreover, a tradeoff between improvement in false alarm rate and possible reduction in detection rate due to good prediction performance should be recognized. Again observing Figure 4.3, the difference between the real occupancy and the predicted occupancy may be small under incident free conditions for a good prediction model. This may result in a decrease in false alarm rate. Under incident conditions, however, this good prediction model may make the difference smaller than a bad prediction model. Thus, according to the decision function in the SND algorithm, a possible detection may be made by an algorithm incorporated with a bad prediction model. Then, detection rate may be increased for such an algorithm, rather than the algorithm with a good prediction model. As a whole, overall performance of an incident detection algorithm in terms of detection rate versus false alarm rate may not be improved for an algorithm integrated with a good prediction model. The

analytical evaluation of such a trade-off seems nearly impossible for the following reasons:  
The complexity in traffic processes under both normal and incident conditions; the prediction model and detection decision rules employed; the distribution of incidents. Thus, evaluation of the algorithm based on a rich set of real data becomes necessary.

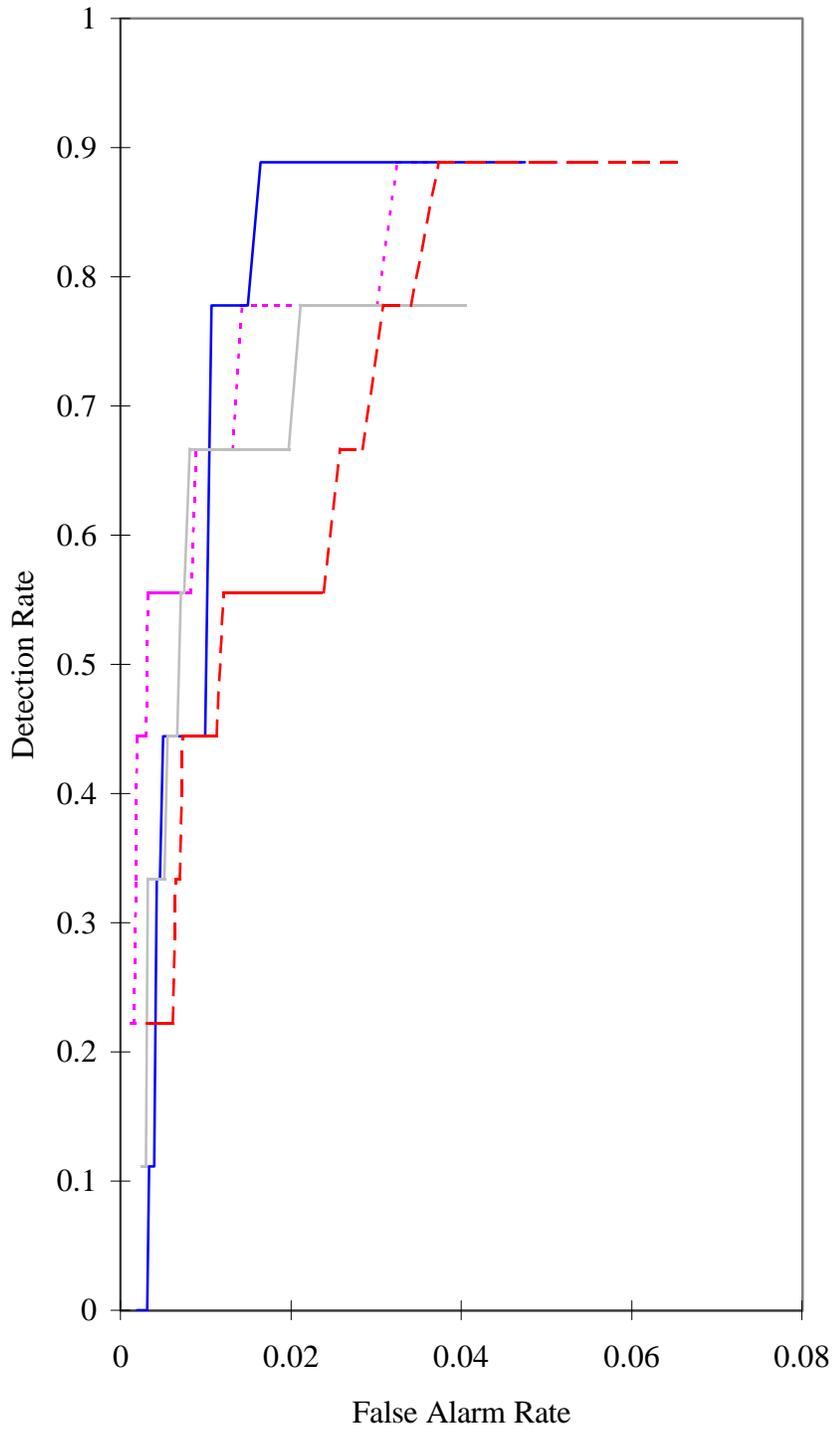


Figure 4.4 Incident Detection Performance Curves

## CHAPTER 5

### CONCLUSIONS AND RECOMMENDATIONS

The overall objective of this research is to evaluate the feasibility of integrating an NNTP model into an incident detection algorithm. This integration is to be accomplished by merging the output of the NNTP model into the decision function of the incident detection algorithm. The expected result is that the output of the NNTP model would be more accurate than the original variable in the decision function. By improving the accuracy of the decision function, the overall performance of the algorithm should improve through increased detection rates and lowered false alarm rates.

The secondary objective is to determine if a time factor could be introduced into the NNTP to increase prediction accuracy. The theory behind this is that prediction accuracy could be improved during the times when traffic conditions change between peak and normal conditions. Accurate prediction during the changing of conditions would be beneficial, because this is a place where false alarms are often generated. The goal is to create the most accurate prediction model possible, the improvements in prediction would then translate into improvements in the detection process.

It is expected that an incident detection model incorporated with advance prediction models would yield better performance in all cases. In a comparison between the SND algorithm, which employs a simple prediction model, and one of the modified algorithms that are combined with a neural network prediction, the modified SND performs better. On the other hand, there is also evidence showing that a good prediction model may not consistently result in a good detection algorithm. Due to the complexity of the elements influencing

algorithm performance, incident detection algorithms should be evaluated using sufficient real data.

The results of the NNTP testing have shown that a time component may only increase the prediction accuracy when traffic profiles show a stable trend. Considering a time component when the traffic profile is not stable may not improve prediction substantially. A possible solution to adjust for varying traffic patterns may come from incorporating data from neighboring loop detectors to make a dual loop predictor.

There are several studies that could be conducted to further this research. As mentioned earlier, considering traffic as variable in the nearby detector stations may improve prediction when traffic profiles are not clear. In addition to re-evaluating this research with more data, may also be helpful to evaluate different detection algorithms. Detection algorithms that could be investigated could include the McMaster and the DEMA algorithm to further research on single detector algorithms. If the incorporation of multiple loops to predict traffic is incorporated, it may be useful to examine dual detector algorithms such as the California algorithms.

Since the SND algorithm is currently in use, there is potential for the possible implementation of this research. However, before implementation could take place, a study to evaluate the nature of traffic, and to determine traffic patterns would need to be conducted. The NNTP model would then have to be calibrated to the specific site conditions. In addition to the improved detection algorithm, an expert system could be developed to handle the incident response operations.

## REFERENCES:

1. Freeway Incident Management Handbook, FHWA, July 1991, pp. 1-20.
2. Jeffery, D.J., K. Russam, and D.I. Robertson. Electronic Route Guidance by AUTOGUIDE: The Research Background. Traffic Engineering and Control, 1987, pp.525-529.
3. Stephanedes, Y.J., P.G. Michalopoulos, and R.A. Plum. Improved Estimation of Traffic Flow for Real-Time Control. In Transportation Research Record 795, TRB, National Research Council, Washington D.C., 1981 pp. 28-39.
4. Okutani, I., and Y.J. Stephanedes. Dynamic Prediction of Traffic Volume Through Kalman Filtering Theory. Transportation Research, Part B. Vol. 18., 1984, pp. 1-11.
5. Williams B.M., P.K. Durvasula, D.E. Brown; Urban Freeway Traffic Flow Prediction: Application of Seasonal ARIMA and Exponential Smoothing Models, Presented at the 77th Annual Meeting of Transportation Research Board, Washington, D.C., 1998.
6. Junchaya, T., G. Chang, and A. Santiago. Advanced Traffic Management System: Real-Time Network Traffic Simulation Methodology with a Massive Parallel Computing Architecture. In Transportation Research Record 1358, TRB, National Research Council, Washington D.C., 1992. pp. 13-21.
7. Lawrence J.; Introduction to Neural Networks: Design, Theory and Application; California Scientific Software, CA, 1994.

8. Dougherty M.S., R.K. Howard, and R.D. Boyle; The Use of Neural Networks to Recognise and Predict Traffic Congestion, *Traffic Engineering and Control*, Vol. 34, No.6, pp.311-314, 1993.
9. Zhang H.S., S.G. Ritchie, and Z.-P. Lo; Macroscopic Modeling of Freeway Traffic Using an Artificial Neural Network, Presented at 72nd Annual Meeting of the Transportation Research Board, Washington, D.C., January, 1993.
10. Smith B.L. and M.J. Demetsky; Short-Term Traffic Flow Prediction: Neural Network Approach, *Transportation Research Record* 1453, pp. 98-104.
11. Chang E.C.P., K. Huarng; Application of Neural Network for Freeway Traffic Prediction, Presented at the 73rd Annual Meeting of Transportation Research Board, Washington, D.C., 1994.
12. Kwon E. And Y.J. Stephanedes; Comparative Evaluation of Adaptive and Neural-Network Exist Demand Prediction for Freeway Control, *Transportation Research Record* 1446, pp.66-76
13. Park B., C.J. Messer, and T. Urbanik II; Short-Term Freeway Traffic Volume Forecasting Using Radial Basis Function Neural Network, Presented at the 77th Annual Meeting of Transportation Research Board, Washington, D.C., 1998.
14. Ahmed S.A. and A.R. Cook, Discrete Dynamic Models for Freeway Incident Detection Systems, *Transportation Planning and Technology*, Vol. 7, pp.231-242, 1982.
15. Cook, A.R. and D.E. Cleveland, The Detection of Freeway Capacity-Reducing Incidents by Traffic Stream Measurements, Transportation Research Board, *Transportation Research Record* 495, 1974, pp. 1-11.

16. Dudek, C. L., C.J. Messer, and N.B. Nuckles; Incident Detection on Urban Freeways, Transportation Research Record 495, pp. 12-24, 1974.
17. Gall, A.I., and F. Hall, Distinguishing Between Incident Congestion and Recurrent Congestion: a Proposed Logic, Transportation Research Record 1232, pp. 1-8, 1990.
18. Hsiao C.-H., C.-T. Lin, and M. Cassidy; Application of Fuzzy Logic and Neural Networks to Automatically Detect Freeway Traffic Incidents, Journal of Traffic Engineering, Vol. 120, No.5, 1994.
19. Teng, H.-L. Detecting Changes in Freeway Traffic States Using the CUSUM Algorithm, Ph.D. Dissertation, Department of Civil Engineering, Purdue University, 1997.
20. Stephanedes Y., and A. Chassiakos, Techniques for Detection of Incidents and Traffic Disturbances, University of Minnesota, Minneapolis, Center for Transportation Studies, Minneapolis, MN, 1994, pp. 110.
21. Dia H. and G. Rose; Development and evaluation of neural network freeway incident detection models using field data, Transportation Research, Part C, Vol. 5, No. 5, pp. 313-331, 1997.
22. Ritchie S.G. and R.L. Cheu, Simulation of Freeway Incident Detection Using Artificial Neural Networks, Transportation Research Part C, Vol. 1, No. 3, pp. 203-217, 1993.
23. Stephanedes Y. And X. Liu; Artificial Neural Networks for Freeway Incident Detection, Transportation Research Record 1494, pp. 91-97, 1993.
24. Black, Justin, Automatic Incident detection Algorithms, <http://path.berkeley.edu>. 1997.

25. Cheu R.L. and S.G. Ritchie; Automated Detection of Lane-Blocking Freeway Incidents Using Artificial Neural Networks, Transportation Research Part C, Vol. 3, No. 6, pp. 371-388, 1995.
26. Stephanedes Y. And Hourdakis, J., Transferability of Freeway Incident Detection, Transportation Research Record 1554, pp. 184-95, 1994.
27. Skarbodinis, A., K. Petty, and R. Bertini, I-880 Field Experiment: Analysis of Incident Data, Transportation Research Record 1603, pp. 72-80, 1997.
28. Petty, K., FSP 1.1: Analysis Software for the FSP Project, Department of Electrical Engineering, University of California, Berkley, 1994.
29. Deek, A., and S. Ishak, A. Khan, Impact of Freeway Geometric and Incident Characteristics on Incident Detection, Journal of Transportation Engineering, 122(6), pp. 440-46, 1996.
30. Chang E. C.-P.; A Neural Network Approach to Freeway Incident Detection, Proceedings of the 3rd International Conference on Vehicle Navigation & Information Systems, pp. 641-647, 1992.
31. McGowan, P., and P. Irwin, TransGuide Transportation Guidance System: Technology in Motion, <http://www.transguide.dot.state.tx.us>, 1994.
32. Description of the TransGuide ATMS Design, Section 5, <http://www.transguide.dot.state.tx.us>, 1994.