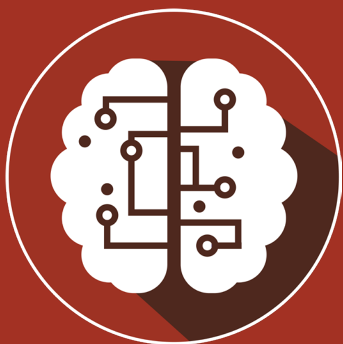
Effectiveness of TMC AI Applications in Case Studies

PUBLICATION NO. FHWA-HRT-21-081 JUNE 2022



U.S. Department of Transportation
Federal Highway Administration

Research, Development, and Technology
Turner-Fairbank Highway Research Center
6300 Georgetown Pike
McLean, VA 22101-2296

FOREWORD

The Turner-Fairbank Highway Research Center performs advanced research on several areas of transportation technology for the Federal Highway Administration (FHWA). The Office of Operations Research and Development (HRDO) focuses on improving operations-related technology through research, development, and testing.

This report presents research about the potential for using artificial intelligence (AI) algorithms for highway traffic incident detection, specifically an AI-based incident-detection framework that can leverage large-scale sensor data along with advanced learning algorithms to improve the performance of incident detection. This research project was sponsored by the FHWA's HRDO. This report may be a useful reference for incident management teams, traffic operators, and those interested in traffic incident detection.

Brian P. Cronin, P.E.
Director, Office of Operations Research and Development

Notice

This document is disseminated under the sponsorship of the U.S. Department of Transportation (USDOT) in the interest of information exchange. The U.S. Government assumes no liability for the use of the information contained in this document.

The U.S. Government does not endorse products or manufacturers. Trademarks or manufacturers' names appear in this report only because they are considered essential to the objective of the document.

Quality Assurance Statement

The Federal Highway Administration (FHWA) provides high-quality information to serve Government, industry, and the public in a manner that promotes public understanding. Standards and policies are used to ensure and maximize the quality, objectivity, utility, and integrity of its information. FHWA periodically reviews quality issues and adjusts its programs and processes to ensure continuous quality improvement.

Recommended citation: Federal Highway Administration, *Effectiveness of TMC AI Applications in Case Studies* (Washington, DC: 2021) https://doi.org/10.21949/1521668

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. FHWA-HRT-21-081	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Effectiveness of TMC AI Applications in Case Studies		5. Report Date February 2022 6. Performing Organization Code
7. Author(s) H. Yang (ORCID: 0000-0003-2808-8852), M. Cetin (ORCID: 0000-0003-2003-9343), Z. Wang, Z. Huang, S. Nallamothu (ORCID: 0000-0002-7457-3704), P. Huang		8. Performing Organization Report No.
9. Performing Organization Name and Address Leidos, Inc. 11251 Roger Bacon Drive Reston, VA 20190 Old Dominion University 4700 Elkhorn Avenue Norfolk, VA 23529		10. Work Unit No. 11. Contract or Grant No. DTFH61-16-D00030
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Highway Administration 1200 New Jersey Avenue SE Washington, DC 20590		13. Type of Report and Period Covered Final Report; January 2020–December 2020 14. Sponsoring Agency Code HRDO-10
15. Supplementary Notes The contracting officer's technical representative for this report was Peter Huang (HRDO-10).		

16. Abstract

Traffic incident detection is a crucial task in traffic management centers (TMCs) that typically manage large highway networks with limited staff. An effective automatic incident-detection approach could benefit TMCs by helping to report abnormal events in a timely and accurate manner and optimize operating resources. During the past decades, researchers have made significant progress in developing such automatic approaches. Nevertheless, the majority of the developed approaches have shown limited success in the field, largely because of concerns about their often-costly false alarms (e.g., misdispatching response teams to a nonexistent incident). Fortunately, recent advances in artificial intelligence (AI) are expected to provide opportunities for improving conventional TMC operations. This project aimed to propose an AI-based incident-detection framework that can leverage large-scale sensor data along with advanced learning algorithms to improve the performance of incident detection. Researchers investigated the generic algorithmic problems in designing a detection approach and emphasized the architecture of the AI-based detection framework by including learning and evolving capabilities. The proposed framework was assessed with a fully controlled experiment in simulation that consisted of numerous traffic and incident scenarios. The results indicated that the proposed AI-based framework achieved higher detection rates, lower false alarm rates, and shorter time to detect the incidents in the studied scenarios than conventional approaches. Some extensions of the proposed framework are also discussed.

17. Key Words Artificial intelligence, traffic incident detection, sensor data, traffic management centers, neural networks	18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161. http://www.ntis.gov		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 48	22. Price N/A

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized.

SI* (MODERN METRIC) CONVERSION FACTORS

APPROXIMATE CONVERSIONS TO SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
in	inches	25.4	millimeters	mm
ft	feet	0.305	meters	m
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
		AREA		
in^2	square inches	645.2	square millimeters	mm^2
ft^2	square feet	0.093	square meters	m^2
yd^2	square yard	0.836	square meters	m^2
ac	acres	0.405	hectares	ha
mi^2	square miles	2.59	square kilometers	km^2
		VOLUME		
fl oz	fluid ounces	29.57	milliliters	mL
gal	gallons	3.785	liters	L
ft^3	cubic feet	0.028	cubic meters	m^3
yd^3	cubic yards	0.765	cubic meters	m^3
		NOTE: volumes greater than 1,000 L shall be shown in m^3		
		MASS		
oz	ounces	28.35	grams	g
lb	pounds	0.454	kilograms	kg
T	short tons (2,000 lb)	0.907	megagrams (or “metric ton”)	Mg (or “t”)
		TEMPERATURE (exact degrees)		
°F	Fahrenheit	5 (F-32)/9 or (F-32)/1.8	Celsius	°C
		ILLUMINATION		
fc	foot-candles	10.76	lux	lx
fl	foot-Lamberts	3.426	candela/m^2	cd/m^2
		FORCE and PRESSURE or STRESS		
lbf	poundforce	4.45	newtons	N
lbf/in^2	poundforce per square inch	6.89	kilopascals	kPa

APPROXIMATE CONVERSIONS FROM SI UNITS

Symbol	When You Know	Multiply By	To Find	Symbol
		LENGTH		
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
m	meters	1.09	yards	yd
km	kilometers	0.621	miles	mi
		AREA		
mm^2	square millimeters	0.0016	square inches	in^2
m^2	square meters	10.764	square feet	ft^2
m^2	square meters	1.195	square yards	yd^2
ha	hectares	2.47	acres	ac
km^2	square kilometers	0.386	square miles	mi^2
		VOLUME		
mL	milliliters	0.034	fluid ounces	fl oz
L	liters	0.264	gallons	gal
m^3	cubic meters	35.314	cubic feet	ft^3
m^3	cubic meters	1.307	cubic yards	yd^3
		MASS		
g	grams	0.035	ounces	oz
kg	kilograms	2.202	pounds	lb
Mg (or “t”)	megagrams (or “metric ton”)	1.103	short tons (2,000 lb)	T
		TEMPERATURE (exact degrees)		
°C	Celsius	1.8C+32	Fahrenheit	°F
		ILLUMINATION		
lx	lux	0.0929	foot-candles	fc
cd/m^2	candela/m2	0.2919	foot-Lamberts	fl
		FORCE and PRESSURE or STRESS		
N	newtons	2.225	poundforce	lbf
kPa	kilopascals	0.145	poundforce per square inch	lbf/in^2

*SI is the symbol for International System of Units. Appropriate rounding should be made to comply with Section 4 of ASTM E380. (Revised March 2003)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

AI	artificial intelligence
ANN	artificial neural network
APID	all-purpose incident detection
ARIMA	autoregressive integrated moving average
CA No. 7	California algorithm No. 7
CNN	convolutional neural network
DBN	deep belief network
DL	deep learning
DR	detection rate
FAR	false alarm rate
FN	false negative
FP	false positive
GPS	global positioning system
HIOCC	high occupancy
KNN	K-nearest neighbors
LSTM	long short-term memory
ML	machine learning
TID	traffic incident detection
TMC	traffic management center
TP	true positive
TTD	time to detect
SND	standard normal deviate
SVM	support vector machine
vph	vehicles per hour

CHAPTER 1. INTRODUCTION

Traffic incidents are one of the critical factors that frequently disrupt daily operations of highway systems. It has been estimated that they account for nearly a quarter of all delays on the highway system in the United States (USDOT 2006, 2019). In addition, they can significantly impact the safety of both motorists and incident responders by exposing them to the risk of secondary incidents (Yang et al. 2018). Thus, incident-management agencies are actively working on various strategies with the common goal to detect, respond to, and remove incidents and restore traffic operations as safely and quickly as possible. Many of the implemented strategies, such as quick clearance laws, have shown great benefits in improving safety, mobility, and motorist satisfaction (Fries et al. 2012; Rensel et al. 2018). Among all the strategies, traffic incident detection (TID) is one of the most important tasks. Timely and accurate detection of incident occurrence is critical to time-sensitive incident-management plans. Delays in detecting incidents may increase the severity of victim injuries and cause heavier congestion. Meanwhile, falsely detecting incidents increases the cost of operating incident-management programs.

TID is challenging because of the randomness of incident occurrence time and location. Many traffic management centers (TMCs) still heavily rely on human-based detection approaches, such as visual checking by TMC operators reviewing highway surveillance cameras (Williams and Guin 2007). Human detection requires a large share of the workforce in TMCs to provide 24/7 examination over a broad road network. This demand often leads to a shortage of operators (e.g., during holidays and weekends) and high costs for recruiting and overtime pay for staff. These issues have led to a growing concern among transportation agencies that human detection is not capable of continuously providing the needed detection functionality given the limited resources. This concern has generated increased interest in automating the incident-detection process. Over the past few decades, several automatic detection algorithms have been developed as part of the solutions for intelligent transportation systems. These algorithms, ranging from simple comparative approaches to advanced machine learning (ML) methods, have emerged as viable alternatives for detecting highway incidents with the access of various sensor data (Dudek et al. 1974; Masters et al. 1991; Dia and Rose 1997; Samant and Adeli 2000; Jin et al. 2001; Yao et al. 2014; Wang et al. 2016). Despite promising features, early studies showed that these algorithms were often disabled or ignored by many TMCs because of concerns about the unacceptable false alarm rates and weak transferability among highways (Williams and Guin 2007). For example, a nationwide survey found that TMCs consider no more than 10 false alarms per day to be acceptable, but many existing algorithms cannot meet this expectation (Williams and Guin 2017). The failure of these algorithms is attributable to many factors, such as restricted modeling assumptions and time-consuming calibration efforts.

With fast-growing data and computational capabilities, the advancements of artificial intelligence (AI) have drawn significant attention in various fields, including the transportation sector. For example, many of the core functions of emerging autonomous vehicles (such as object detection and motion planning) have been built on the success of AI solutions. It is expected that TMCs can also benefit from AI in addressing safety and congestion challenges across the large highway networks they manage. The use of AI is largely facilitated by the massive amount of firsthand data that TMCs can access through their numerous sensor nodes in road networks. Although promising, AI techniques have not been fully tested in the context of

TMC operations such as incident detection and flow prediction. As such, the primary objective of this project was to examine the potential of using AI algorithms for highway TID. It intended to leverage the power of traffic-simulation models to quantitatively assess the performance of an AI-based detection approach in a fully controlled experimental condition. The simulation model enabled the creation of numerous incident cases and traffic conditions that cannot be easily collected in the field and helped to overcome many other real-world issues, such as the lack of precise incident data and sparse traffic measurements due to larger detector spacing. The algorithmic problem of incident detection is discussed to raise awareness of improving existing methods. This project demonstrates the architecture of the detection approach based on a simple AI-based framework. It contributes to the literature by demonstrating the advantages of including learning and evolving capabilities in the developed AI-based framework.

CHAPTER 2. LITERATURE REVIEW

Automatic TID has been discussed for many years. There has been extensive research on developing detection methods using different data and techniques. In general, the representative studies can be grouped into two main categories: methods comparing static or dynamic thresholds and methods based on statistical or ML approaches.

METHODS COMPARING STATIC OR DYNAMIC THRESHOLDS

Comparative methods assume that traffic metrics such as volume, occupancy, and speed will change after an incident. They compare observed traffic metrics with predefined threshold values. Once the metrics exceed such thresholds, an incident is reported. Typical algorithms include decision tree algorithms, pattern recognition algorithms, and all-purpose incident detection (APID) algorithms.

Decision tree algorithms organize predefined thresholds in a hierarchal structure and provide different output states on leaf nodes. One of the classic algorithms is California algorithm No. 7 (CA No. 7). CA No. 7 derives traffic metrics parameters via the observed occupancy data from two adjacent sensors and compares such parameters with predefined values (Payne and Tignor 1978). Later, an APID algorithm was developed to improve the transferability of CA No. 7 and expand the major elements of the California algorithms into a comprehensive structure (Masters et al. 1991; Li et al. 2016; Saifuzzaman et al. 2018). Collins et al. (1979) introduced the high occupancy (HIOCC) algorithm, which monitors the high-frequency detector data for changes over time. They conducted the field test with data from two highways in London and proved that HIOCC can work well under congested conditions. Other traffic metrics, such as travel time and vehicle speed, have also been used to detect incidents. For example, the Transport and Road Research Laboratory developed a pattern recognition algorithm (Collins et al. 1979) in which vehicle speeds were estimated by observing travel times between loop detectors. Once vehicle speeds exceeded the pre-established threshold values for a preset number of consecutive time steps, an incident alarm was triggered.

Similarly, researchers have developed methods based on statistical metrics to detect potential incidents, including the standard normal deviate (SND) algorithm and Bayesian algorithms (Dudek et al. 1974; Tsai and Case 1979; Li et al. 2017). The SND algorithm computes the SND of the traffic-control measure (such as occupancy), and if the SND exceeds a predefined threshold, an incident is reported. Bayesian algorithms introduce the relative difference of occupancies used in CA No. 7 approaches and derive the conditional probability using Bayesian statistics. These algorithms are often static and cannot be easily transferred to different scenarios. Besides simply using occupancy data, some models compare other traffic variables, such as speed and travel time, with predicted outputs via algorithms such as autoregressive integrated moving average (ARIMA) to prompt a potential incident report (Ahmed and Cook 1979, 1980, 1982). The ARIMA model takes advantage of the temporal correlation between traffic variables measured in current time step t and previous time step $t-k$ and learns the normal pattern of such a relationship under incident-free conditions. Nevertheless, the reliability of existing models remains an issue in scenarios like recurrent congestion. Some researchers also smoothed the observed traffic variables and used weighted average values as the model input to reduce potential false alarms (Wang et al. 2016).

Researchers have also sought data from sources other than traffic sensors, such as the global positioning system (GPS) and images, for incident detection. For instance, the Autoscope incident detection algorithm was improved by including ancillary information provided by video detection (Michalopoulos 1991; Michalopoulos et al. 1993). By using travel time and other spatial traffic measures collected by probes, information about traffic conditions can be archived with higher accuracy and precision (Asakura et al. 2017). For example, Hellinga and Knapp (2000) collected travel time data based on transponder readers to detect incidents. If the travel time exceeded the predefined thresholds, an incident was detected. Recently, Asakura et al. (2017) archived travel times and the number of probe vehicles passing through a bottleneck to detect potential incidents. These studies suggest that no single data source or algorithm will always work for different traffic and incident scenarios.

METHODS BASED ON STATISTICAL OR ML APPROACHES

Many learning algorithms consider incident detection as a task to construct a classification model according to traffic metrics. Approaches such as ML, deep learning (DL), and clustering have been applied to classify traffic states with and without incidents (Fries et al. 2012).

Artificial neural networks (ANNs) have been widely studied to detect freeway incidents, and their performance has been promising (Cheu and Ritchie 1995; Dia and Rose 1997; Lee et al. 2004). Many models have been developed, such as multilayer feed-forward neural network, constructive probabilistic neural network, and probabilistic neural network (Jin et al. 2001; Wen et al. 2001). Other ML algorithms, such as support vector machine (SVM), particle swarm optimization, and random forest, have also been applied to detect traffic incidents. For example, Yao et al. (2014) used the tabu search algorithm to optimize the parameters of SVM to detect incidents. Local optimums can be avoided. In addition, ML algorithms have been coupled with ANNs to improve the detection performance (Zhao et al. 2018). Kinoshita et al. (2015) introduced a traffic state model based on a probabilistic topic model to describe traffic states for a variety of roads.

Some researchers have examined the use of DL approaches for incident detection. DL uses multiple-layer architectures or deep architecture of neural networks to extract inherent features in data of different complexities and can represent them without prior knowledge, which offers promising functions for TID. For example, El Hatri and Boumhidi (2018) proposed a novel fuzzy DL-based detection method that considers the spatial and temporal correlations of traffic flow. The fuzzy logic is introduced to avoid the slow convergence rate and trapping by local optimums during tuning learning parameters. Zhang et al. (2018) used deep belief network (DBN) and long short-term memory (LSTM) to detect traffic accidents from social media data. DBN outperforms SVM and ANNs when processing Twitter data and matching tweets to nearby abnormal traffic data. LSTM does not perform well because it depends on sequential information, and words (token) in tweets are not sequentially organized well. When using DL approaches, the detection performance may improve but the relative time and space complexity will increase, so more computing resources will be needed for real-time implementation. Such complex models are also prone to overfitting issues.

With the growth of social media, crowdsourcing data, such as Twitter posts, have also been examined. For example, researchers extracted information from social media posts using natural language processing algorithms, mapped the data into the high-dimensional vectors in the feature space, and classified incident patterns on temporal and spatial dimensions (Nguyen et al. 2015; Gu et al. 2016; Salas et al. 2017). Gu et al. (2016) proposed a methodology to crawl, process, and filter tweets that are accessible by the public for free. Tweets were acquired from Twitter using the REST API in real time. However, relevant social media data are often limited by time and locations and can serve only as a supplement for incident detection because many incidents were not described in a timely or precise manner by social media sources.

Table 1 provides details of representative studies, including conventional approaches and ML approaches. The table lists information regarding data, method, variables, detection rate (DR), time to detect (TTD), false alarm rate (FAR), and complexity.

SUMMARY

The studies described examined the feasibility of detecting incidents with different types of data sourced from loop detectors, vehicle trajectories, images, and social media data. However, several issues have not been well addressed. One concern is the overfitting issue. Complex models, such as DL approaches, can outperform comparative approaches with training scenarios but are prone to inferior performance when transferred to other scenarios. In addition, incident detection under light volume remains a challenging issue because no comparable traffic pattern changes can be observed and used to prompt incident reporting. In addition, the unbalanced dataset applied in the tuning process of models such as SVM is often problematic. The lack of accurate incident information limits the training performance of these models. Another major concern is the models' inability to learn from historical information. Humans can avoid continuously making the same mistakes, but many models with fixed mathematical structures or decision rules cannot do so if the same input data are given. Updating and learning ability is needed to improve model performance. Thus, this project aimed to extend existing efforts by introducing an AI-based framework for enhancing predictive performance in incident detection.

Table 1. Representative work on incident detection.

Reference	Data (training/test size, time interval, location)	Method	Variables	DR (%)	TTD (s)	FAR (%)	Complexity
Stephenedes and Chassiakos 1993	I–35W, Minneapolis, MN	Filtering, CA No. 7	Occupancy	93.00	244.00	0.500*	Simple
Cheu and Ritchie 1995	Simulated data with 100 incidents, 30-s interval, SR 91 Riverside Freeway, California	ANN	Volume, occupancy at both upstream and downstream detectors	21.00	60.00	0.127*	Moderate
Adeli and Samant 2000	1 incident over a period of 150 min, 30-s interval, simulated freeway	ANN	Feature extraction of occupancy and volume	100.00	47.80	1.200*	Moderate
Jin et al. 2001	45 incidents, 30-s interval, I–880, California	ANN	Volume, occupancy, speed	86.96	228.00	0.200*	Moderate
Yao et al. 2014	304 incidents, 30-s interval, 500–700-m interval, April 16–20, 2012, Liaoning, China	SVM	Weather, time, occupancy, volume	95.70	72.60	4.820*	Moderate
Gu et al. 2016	322 incidents, 5-min interval, Twitter data, January–July 2013, Japan	DSAE	GPS, three layers	79.80	NA	0.040ǂ	Moderate
Asakura et al. 2017	Probe data, 1-min interval, April–July 2016, Iowa	SND, outlier detection	Speed, probe data	54.10	887.00	0.043ǂ	Simple
El Hatri and Boumhidi 2018	30 incidents, 100-s interval, simulated data via SUMO	DSAE	Traffic flow count	98.23	192.44	0.240ǂ	Complex
Zhao et al. 2018	138 incidents, 1,518 incident-free cases, Chongqing, China	LVQ, fuzzy logic	Volume, occupancy, speed, meteorological parameters	96.50	152.40	0.210*	Moderate
Zhang et al. 2018	584,000 geotagged tweets in northern Virginia, 2,420,000 tweets in New York, NY from January to December 2014	DBN, LSTM	Token features extracted from tweets	95.00	NA	30.000ǂ	Complex

*FAR was calculated based on the number of false alarm cases divided by the total number of instances; ǂFAR was calculated based on the number of false alarm cases divided by the total number of nonincident instances.
DSAE = deep sparse autoencoder; LVQ = learning vector quantization; NA = not available; SUMO = simulation of urban mobility.

CHAPTER 3. INCIDENT-DETECTION PROBLEM

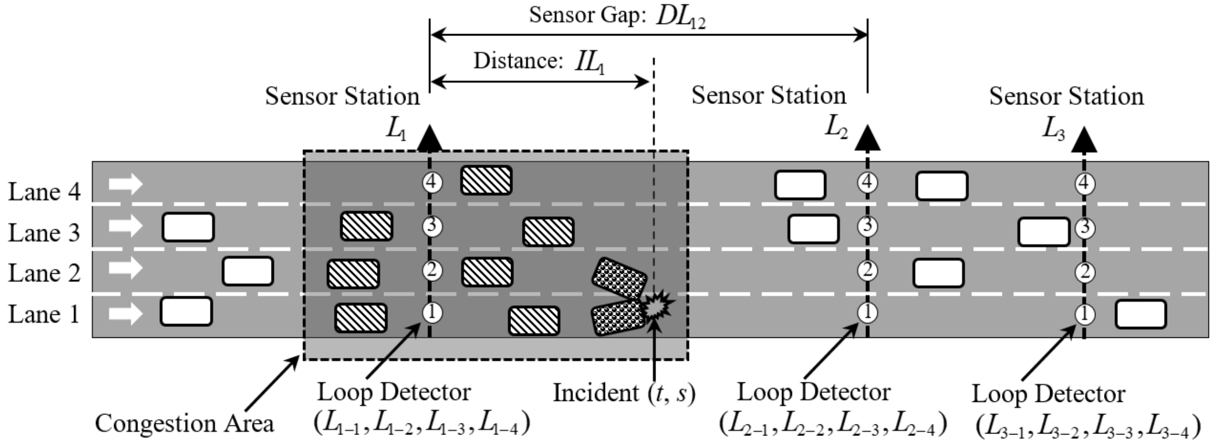
As illustrated in figure 1, suppose an incident occurred at location s and time step t. This incident may lead to congestion due to lane closure. From the perspective of TMCs, one key task is the timely detection of the incident. Data sources such as surveillance cameras and sensor systems are often used to detect the incident. Among these data sources, data from loop detectors are frequently used. The feasibility of using loop detector data for incident detection largely depends on the hypothesis that incident occurrence will be indirectly reflected by the fluctuation of traffic conditions. Thus, the generic problem of incident detection becomes the analytics of the changes in detector measurements. Equation 1 provides a high-level generalization of the detection problem.

$$Y \leftarrow f(\boldsymbol{X}) \tag{1}$$

Where:

Y = prevailing traffic condition with (y = 1) or without (y = 0) incident occurrence.
$\boldsymbol{X}$ = the detector measurements.
$f()$ = a specific modeling approach that associates the detector measurements with the prevailing traffic condition.

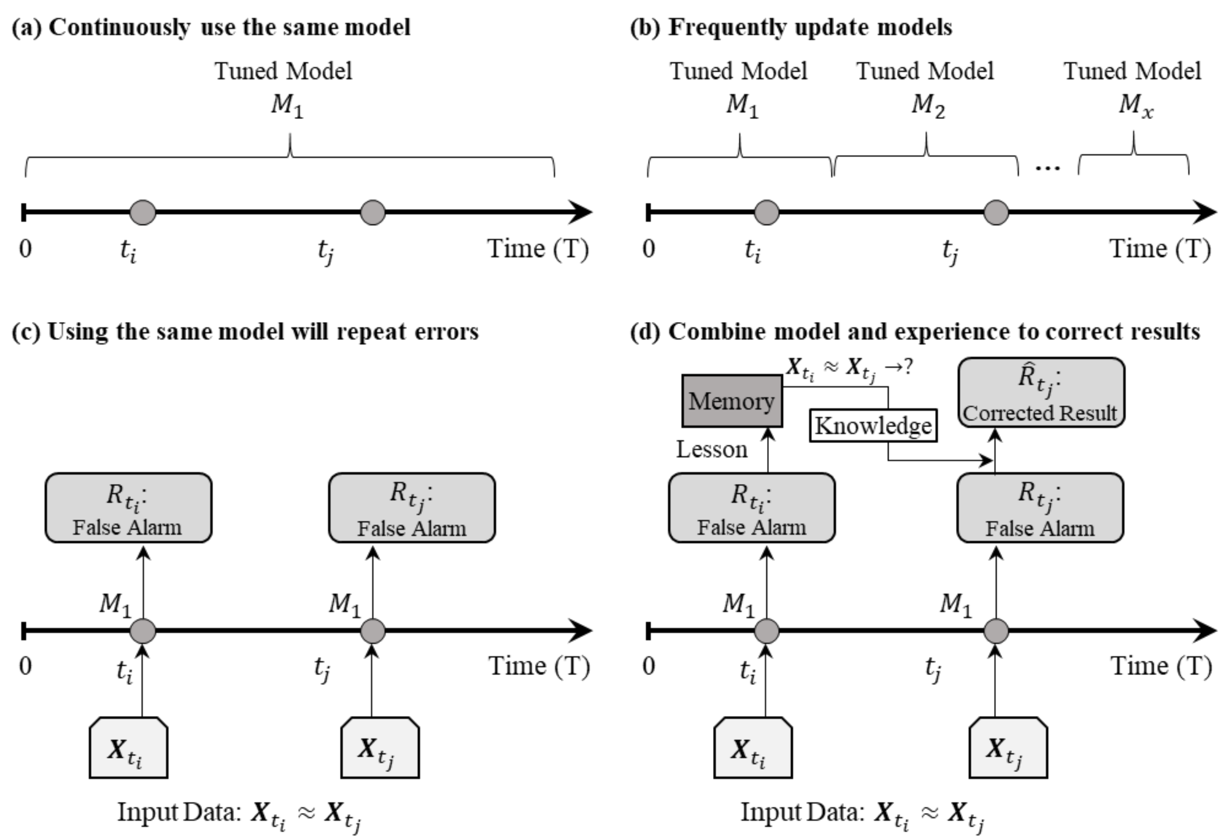
By specifying appropriate model structure and conducting model calibration and validation, a final well-tuned model M_x:$\hat{f}(\boldsymbol{X} \mid \alpha)$ can be established, where $\hat{f}()$ represents the finalized model with a calibrated parameter set α. Depending on the structure of $\hat{f}()$ and the number of elements in α, the model complexity and required computational resources will be different. The tuned model M_x is expected to be as efficient and accurate as possible.



Source: FHWA.

Figure 1. Illustration. Conceptual illustration of incident occurrence on highways.

Other than the inherent restrictions of each model M_x, a model's predictive performance will also be affected by its implementation strategy. Two typical strategies can be considered: continuously implementing a tuned model and periodically updating a model during implementation. The first strategy is illustrated in figure 2-A. Once model M_1 has been tuned, it is deployed continuously. Such a strategy does not consider any resultant error that the model may have. Suppose it made a false prediction at time step t_i. It will continuously repeat the same false prediction (figure 2-C) at time step t_j if the input data X_{tj} is similar to X_{ti}. If the prediction errors keep occurring, an update of the model may be necessary. The second implementation strategy is illustrated in figure 2-B. The deployed model M_1 is reviewed periodically (e.g., weekly, monthly). If the performance of the deployed model raises concerns, it is updated with additional calibration efforts. Within an updating time window, model M_1 (figure 2-D) will still repeat the same false prediction at time step t_i with the same input data X_{ti}. If an updated model M_2 is implemented in the second time window, it will produce new prediction result $\hat{R}_{tj}$ at time step t_j. If the updating efforts have successfully addressed the error issues observed in the previous time window, $\hat{R}_{tj}$ is likely to be improved. Otherwise, it may still be subject to similar errors because of the limited improvements of the modeling performance. Implementing updated models periodically has potential benefits in terms of error reduction. However, this strategy may not be computationally practical. One major concern is the updating frequency. Also, adding one historical record (e.g., X_{ti}) to a large training dataset may not update a model much in terms of predictive performance because the influence of a single record among a large amount of training data will be limited when calibrating model parameters (unless it is an extreme case). Thus, the value of adding X_{ti} to the training dataset is not substantial. In real-time application at a TMC, a stable approach that does not require frequent updates will be more likely to facilitate operators' use and management.



Source: FHWA.

Figure 2. Illustrations. Different modeling strategies and outcomes.

To augment the value of any historical false predictions (e.g., R_{ti}), it is helpful to carefully examine and leverage the use of the relevant traffic condition measurements X_{ti}. An experienced TMC operator that falsely flagged a traffic condition (e.g., in a case that never happened before) is highly likely to learn from the recent failure to improve subsequent judgments in similar traffic conditions. Thus, a data-driven approach with similar human intelligence is expected to iterate and improve during the implementation process.

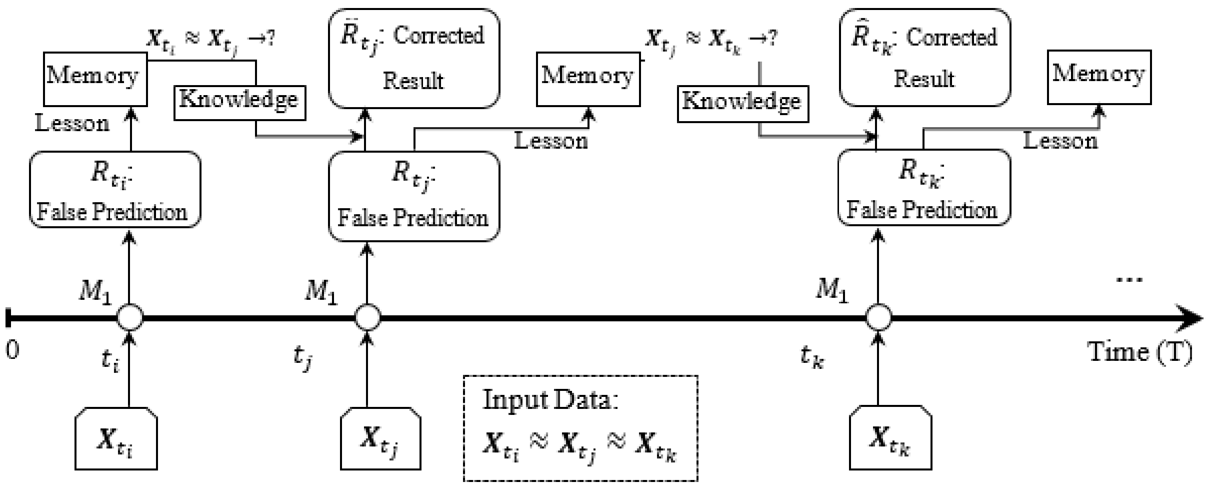
CHAPTER 4. METHODOLOGY

CONCEPTUAL FRAMEWORK OF AN AI-BASED INCIDENT-DETECTION APPROACH

The previous chapter discussed the problems associated with developing and implementing an incident-detection model. A fine-tuned model with an acceptable implementation strategy will facilitate the detection of incidents. However, any specific model will have certain capabilities and deficiencies. Modelers often hope to maximize a model's capabilities while reducing its deficiencies. Nonetheless, because of a number of factors, such as outliers, incomplete information, model assumptions, and exclusion of some factors, it is expected that a model's performance will be capped at a certain level. The deficiency will remain the same if no further effort is made. For example, ordinary least squares can be applied in linear regression to minimize the sum of the squares of the differences between the observed dependent variable in the given dataset and those predicted by the linear function. The calibrated coefficients can help account for only a certain amount of variation in the dependent variables (e.g., in terms of R^2). Likewise, in the context of incident detection, existing models often cannot achieve a perfect detection result. The inherent limitations of a deployed model may not be able to be addressed simply through recalibration.

Instead of tweaking the model again, other approaches that involve revisiting the modeling results and gathering related feedback or ensembling other processes to learn the failures may be more valuable. The needs of these approaches motivated the research team to expand the detection capability through the use of AI that does not limit itself to the fixed-model framework. In other words, AI is leveraged to imitate human behavior that may not be perfectly generalized by a mathematical model. In practice, operators learn from historical operations that they misclassified. The gradually accumulated lessons are likely to enable the operator to avoid repeating similar mistakes. Instead of purely relying on a model, operators can refer to lessons learned or knowledge acquired to double-check the model result. More importantly, operators can keep updating their knowledge while new cases present. Similar evolvement ability to digest and learn things has been frequently used in the literature to build various expert systems, which make machines behave intelligently. A well-known example is the IBM Watson question–answer computer system that successfully won the quiz show *Jeopardy!* against human champions in 2011 (Ferrucci et al. 2013). Watson maintains information from millions of documents such as dictionaries and encyclopedias to build its knowledge (Brynjolfsson and McAfee 2012; Ferrucci et al. 2013).

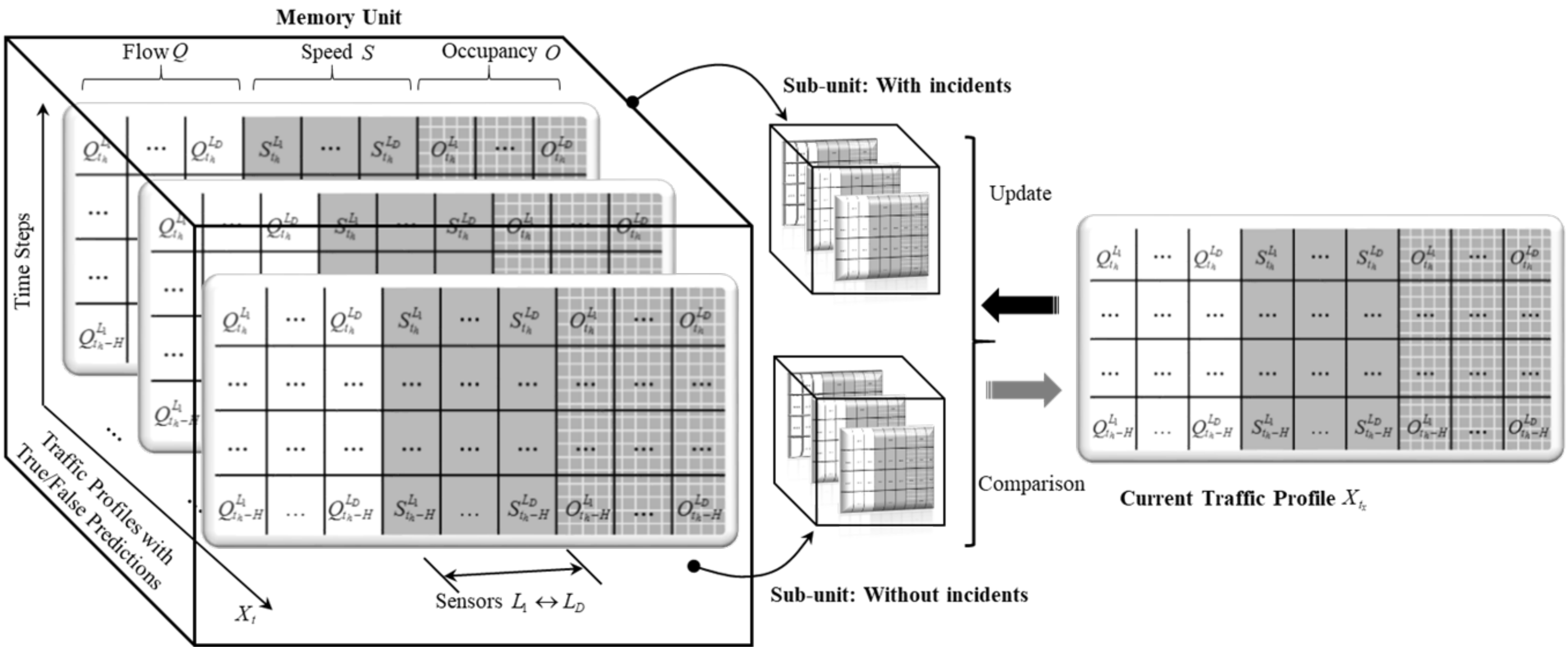
Inspired by the learning and reasoning ability of expert systems like Watson, this project expands current incident-detection modeling practices by framing an AI approach that combines a memory unit with a tuned model. Figure 3 presents the conceptual architecture of the AI approach. In a nutshell, whenever the deployed model (e.g., M_1) predicts the occurrence of an incident, the input data are stored to the memory unit and linked with a label as either a correct or false prediction after verification of the event. For example, the first time the model makes a false alarm R_{ti}, the input variables X_{ti} will be included in the memory (i.e., knowledge database) for future comparisons. At later steps, when false alarms R_{tj} and R_{tk} are prompted, their corresponding traffic conditions X_{tj} and X_{tk} will be further assessed with reference to the memory. Because traffic conditions X_{tj} and X_{tk} are similar to X_{ti} (which was associated with a false alarm), the initial predictions of R_{tj} and R_{tk} will be corrected to $\hat{R}_{tj}$ and $\hat{R}_{tk}$. Thus, despite the false prediction of detection model M_1, these two later cases will be labeled as nonincident scenarios, and their relevant information will be further updated in the memory. Referencing the memory at each step can be considered retrieving knowledge. Updating the memory dynamically is comparable to a TMC operator evolutionarily accumulating the knowledge.



Source: FHWA.

Figure 3. Illustration. An AI modeling framework with memory units and learning ability.

The memory unit described in figure 3 consists of a historical knowledge database that archives different types of traffic information. This is similar to a dictionary that depicts different traffic profiles with relevant indexes to incident occurrence (e.g., true/false alarms in incident detection). The construction of the traffic profiles in the memory unit may differ depending on the data source. In terms of the typical data from loop detectors, this project gathered traffic flow, speed, and occupancy, either by station-level averages or lane-by-lane measurements from simulation models. Other derived metrics, such as speed/flow variances and correlations, can also be included. These multidimensional measurements together represent a snapshot of the traffic condition at a given time period. Figure 4 illustrates an example of the memory unit that stores the traffic profiles. When model M_1 predicts the occurrence of an incident $R_{tx} = 1$, the prevailing traffic condition X_{tx} is compared with the memorized information. If similar traffic profiles are detected, their indexed incident facts (i.e., absence or presence of an incident) will be used to correct the initial prediction from M_1. For example, if K traffic profiles in the memory unit are very similar to X_{tx}, the corresponding incident facts $I_1, I_2, \ldots I_K \in \{0 = \text{no incident}; 1 = \text{incident}\}$ will be evaluated. If the incident facts show more "0's", the early prediction by M_1 is highly likely to be wrong. Thus, the prediction results $R_{tx} = 1$ will be updated as $\hat{R}_{tx} = 0$. If the incident facts show more "1's", the prediction by M_1 is likely to be correct, and the prediction results $R_{tx} = 1$ will be further confirmed as $\hat{R}_{tx} = 1$. This process ensures that the model results are not the sole determinant in the decision and helps reduce the risk of false predictions. Upon verification of $\hat{R}_{tx}$, the corresponding X_{tx} will also be included in the memory unit for future reference.



Source: FHWA.

Figure 4. Illustration. Memory unit with the archive of historical information for reference.

As discussed, a key function of the memory unit is to facilitate the assessment of the current traffic profile. It is used to determine similar historical traffic profiles in each subunit. The similarity between the current traffic profile and the historical ones can be determined using a number of approaches, such as random forest, SVM, and K-nearest neighbors (KNN). This project used the simplest KNN approach to illustrate how the similarity between traffic profiles can be quantified. For example, if a traffic profile is defined as $X_t(Q,S,O)$ (where Q represents traffic flow, S denotes sensor speed, and O is the occupancy), the similarity can be calculated based on the Euclidean distance D_{ix} between two profiles, as shown in equation 2:

$$D_{ix} = \sqrt{\sum_{h=0}^{H}[X_{t_x-h}(Q,S,O) - X_{t_i-h}(Q,S,O)]^2} \tag{2}$$

Where:

$X_{tx-h}(Q,S,O)$ = the current traffic profile $X_{tx}(Q,S,O)$'s element obtained at time $t_x - h$.
$X_{ti-h}(Q,S,O)$ = the element of the ith historical traffic profile $X_{ti}(Q,S,O)$ measured at $t_i - h$.
$i = 1, 2, \ldots n$.
t_x = the period that needs incident occurrence prediction.
t_i = the period of ith historical traffic profile $X_{ti}(Q,S,O)$.
h = the number of time steps and $h = 0, 1, \ldots H$.

In equation 2, t_x and t_i can be but might not be the same period of the day. With all the calculated D_{ix}, the top K profiles $X_{tg}(Q,S,O)$ with minimum distance D_{gx} ($g = 1, 2, \ldots K$; g is the index of selected K historical profiles) are selected. The majority rule can be used to update current prediction $R_{tx} = 1$ by model M_1, as shown in equation 3 through equation 5:

$$N_F = N(f(X_{t_g}(Q,S,O)) \rightarrow \text{False Predictions}) \tag{3}$$

$$N_T = N(f(X_{t_g}(Q,S,O)) \rightarrow \text{True Predictions}) \tag{4}$$

$$\hat{R}_{t_x} = \begin{cases} 1, & if\ N_T \geq N_F \\ 0, & if\ N_T < N_F \end{cases} \tag{5}$$

Where:

N_F = the number of profiles that produced false predictions.
N_T = the number of profiles that produced true predictions.

In these equations, $N_F + N_T = K$. In practice, an odd number K (e.g., $K = 5$) is suggested so that equation 5 will not involve the case of $N_F = N_T$. The described procedure will expand model M_1's capability to keep learning from the historical profiles. Any new profile can be rolled over to the memory unit, which keeps the memory unit as updated as possible. The number of historical profiles in the memory unit can be limited (e.g., 2,000 records) to reflect the memory capability of an operator. If the number of historical profiles exceeds the limit, the latest profiles will be kept and the older ones will be phased out so that the memory unit maintains the latest information.

SPECIFICATION OF MODEL M_1 WITH NEURAL NETWORKS

The AI modeling framework in figure 3 requires the implementation of a detection model M_1. Many candidate models can be used. Instead of simple regression approaches and rule-based methods, models that capture the nonlinear relationship between traffic measurements and incident occurrence are preferred. Among the different AI approaches, researchers often choose ANN. This project used a simple ANN to demonstrate a specification of M_1 to support the AI-based detection framework. Other methods, such as SVM, can also be considered. This project's neural network included only three layers: an input layer, a hidden layer, and the output layer. Mathematically, the ANN model can be written as in equation 6:

$$X_a^{(c)} = f_{NN}\left(\sum_{b=1}^{b=B}\left(\omega_a^{(b,c)} \cdot X_{a-1}^{(b)}\right) + \beta_a^{(c)}\right) \tag{6}$$

Where:

$X_a^{(c)}$ = the cth element in the ath layer.
$\omega_a^{(b,c)}$ = the weight parameter that links $X_{a-1}^{(b)}$ and $X_a^{(c)}$.
$X_{a-1}^{(b)}$ = the bth element in the output of the $(a-1)$th layer.
$\beta_a^{(c)}$ = the bias.
$a = 1, 2, \ldots A$ = the index of a layer in the neural network.
$b = 1, 2, \ldots B$ = the element index in the preceding $(a-1)$th layer.
$c = 1, 2, \ldots C$ = the element index in the following ath layer.
$f_{NN}()$ = the activation function.

In equation 6, in the simplest scenario $A = 3$. The activation function can be *tanh*(), *maxout*(), or *sigmoid*() for the calculation of a specific layer's output. For incident detection, *softmax*() was used to calculate the probability of incident occurrence as the final output. The input layer will have the elements in the current traffic profile $X_{tx}(Q,S,O)$. Like the example shown in figure 4, this profile can include current and multistep historical measurements for flow, speed, and occupancy by multiple sensors (e.g., L_1 and L_2 in figure 1) along the target highway section.

To start, input X can be written as $(Q_t^1,S_t^1,O_t^1,Q_t^2,S_t^2,O_t^2)$ because traffic metrics such as flow, speed, and occupancy have frequently been used in previous literature. Because congestion may lead to speed variations, an additionally derived metric, CUSUM, was introduced to depict the extent of speed variation (Sattayhatewa 1999; Parkany and Xie 2005). CUSUM is a cumulative sum control chart used to depict the cumulative sums of the deviations of the observed variables and has been successfully applied in other research areas, such as automatic control (Teng and Qi 2003; Liu et al. 2008). Thus, this project introduced the derived CUSUM of speed given equation 7 and equation 8:

$$HCS_t^i = \max[0, x_t - (\mu_0 + k) + HCS_{t-1}^i] \tag{7}$$

$$LCS_t^i = \max[0, (\mu_0 - k) - x_t + LCS_{t-1}^i] \tag{8}$$

Where:

μ_0 = the mean value of speed in given time steps.

k = slack value.

HCS^i_t = high CUSUM of speed at t of ith sensor.

LCS^i_t = low CUSUM of speed at t of ith sensor.

In equation 7 and equation 8, μ_0 is five time steps in this project, k is set at half of the calculated standard deviation, $i = 1$ stands for the upstream sensor, and $i = 2$ stands for the downstream sensor in terms of variables in input X. Whenever the speed value was outside the upper or lower limit, the variance was calculated and summed. For example, the typical change pattern when an incident occurs is a quick reduction in speed followed by the speed remaining stable. Such change patterns can be reflected by the calculated high CUSUM of speed and its peak values and thus are expected to provide useful information.

BENCHMARK ALGORITHMS

Two classic algorithms were used as the benchmarks: the ANN model and CA No. 7. The ANN model is equivalent to the proposed model without the learning component. As a comparative algorithm, CA No. 7 computes three traffic metrics for comparisons based on the occupancy measurements from a pair of loop detectors. These metrics are spatial difference in occupancy ($OccDF_t$), relative spatial difference of occupancies ($OccRDF_t$), and occupancy values obtained from downstream detector ($DOcc_t$), as shown in equation 9 through equation 11:

$$OccDF_t = Occ_t^{L_1} - Occ_t^{L_2} \tag{9}$$

$$OccRDF_t = \frac{Occ_t^{L_1} - Occ_t^{L_2}}{Occ_t^{L_1}} \tag{10}$$

$$DOcc_t = Occ_t^{L_2} \tag{11}$$

Where:

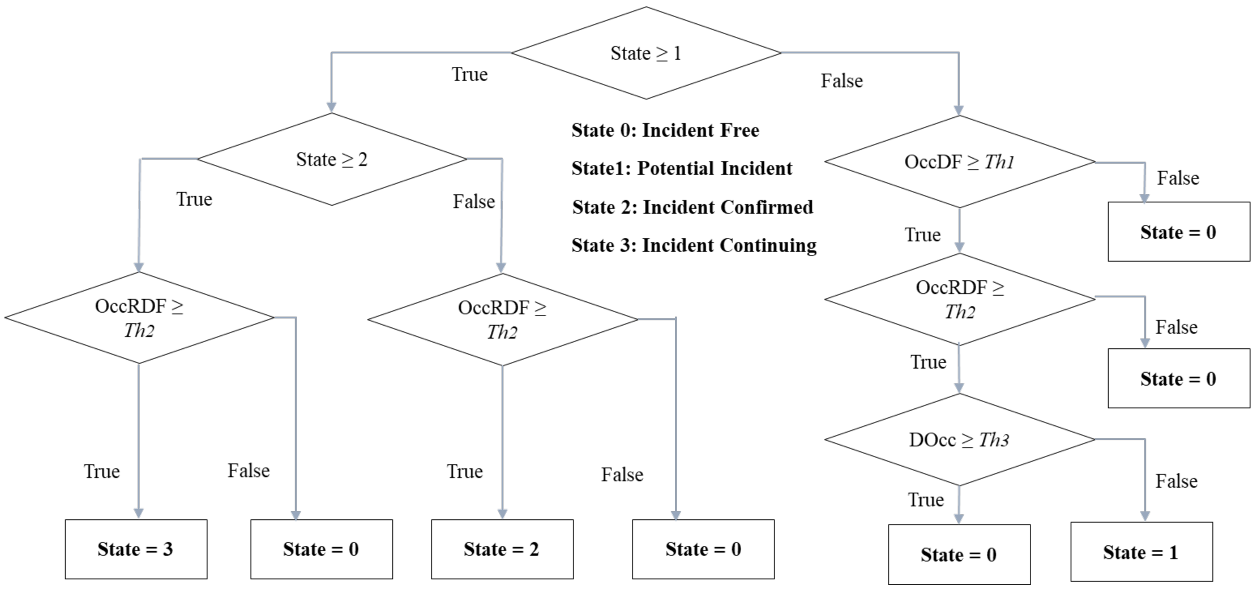
L_1 = the upstream sensor.

L_2 = the downstream sensor.

Occ_t = the measured occupancy at time t.

The three metrics obtained in equation 9 through equation 11 were compared with predefined thresholds Th_1, Th_2, and Th_3, respectively. If $OccDF_t > Th_1$, $OccRDF_t > Th_2$, and $DOcc_t < Th_3$, a potential incident was detected. Furthermore, if $OccRDF_t > Th_2$ for two consecutive steps, an incident occurrence was reported. The optimum thresholds can be calibrated.

As shown in figure 5, the four states of decisions are as follows: state 0 means that no incident exists, state 1 means that a potential incident is detected, state 2 means that the previous potential incident has been confirmed, and state 3 means that the detected incident is continuing. The decision tree is used to judge states based on the comparison between observed values and thresholds. An incident is considered detected for states 2 or 3.



Source: FHWA.

Figure 5. Illustration. Decision tree of CA No. 7.

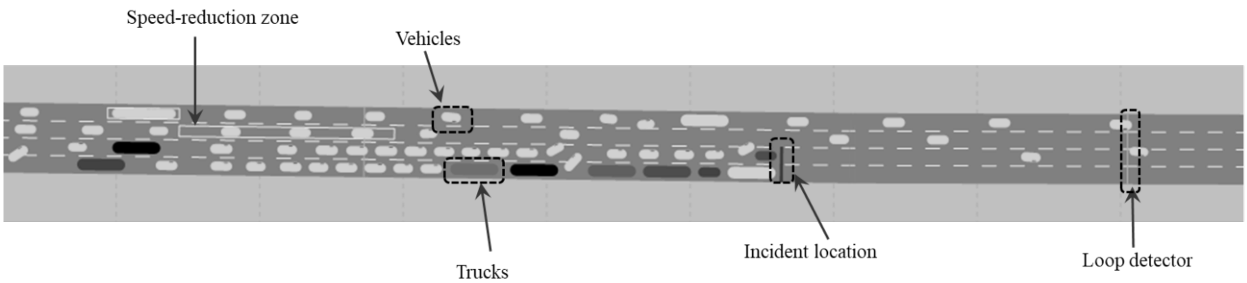
CHAPTER 5. EXPERIMENTS TO TEST THE PROPOSED AI APPROACH

EXPERIMENTAL DESIGN

Ideally, real-world incident and traffic data would be used to train and test the proposed model. A large volume of data is necessary to guarantee the effectiveness of an incident-detection approach based on ML or AI. Although it is not challenging to obtain real-time traffic data from sensor stations on highways (e.g., from Performance Measurement System of the California Department of Transportation), publicly available incident data with accurate incident information are scarce. In particular, the precise incident occurrence time is often not well archived in the database. In addition, the incident frequency of a single segment of road is relatively low because of the randomness of the events, causing data imbalance issues when preparing the training dataset.

Because of the limitations described in the previous paragraph, to test the performance of the proposed AI approach for this project, an experiment was designed using microsimulation in order to have full control over the data collection procedure. The simulation scenarios were fine-tuned to reflect the traffic conditions of a typical highway section. Figure 1 illustrates the designed test scenarios. A four-lane 4-mi straight highway section with a 65-mph speed limit was coded in PTV VISSIM™. Two sensor stations, L_1 and L_2, with a distance of DL_{12} were placed in the middle portion of the highway section. This project considered three levels for DL_{12}: 0.3, 0.5, and 1.0 mi. Each sensor station had four detectors to collect data on traffic flow, speed, and occupancy of each lane at 30-s time intervals. An incident was simulated to occur at time t between the two sensor stations, with a distance of IL_1 to the upstream sensor station L_1. Three values were considered for IL_1: $IL_1 \approx 0$, representing the incident occurring immediately after passing sensor station L_1; $IL_1 = ½DL_{12}$, representing the incident occurring in the middle of the segment; and $IL_1 \approx DL_{12}$, representing the incident occurring immediately before reaching sensor station L_2. The incident was simulated to last 20 min with either one or two lanes on the curbside blocked. To simulate incidents occurring during different traffic conditions, 30 levels of traffic demand were considered: traffic demand varied from 5,100 to 8,000 vehicles per hour (vph) in increments of 100. Altogether, 540 experimental scenarios were created: 3 levels (DL_{12}) × 2 levels (lane closure) × 3 levels (IL_1) × 30 levels (demand) = 540. Each simulation scenario ran for 30 min, with the first 5 min as the warmup period. The incident occurred at t = 10 min. Data were collected between t = 5 min and t = 15 min, which allowed data to be gathered for 5 min before and after the incident. The incident-detection approach was continuously implemented during these 10 min. Each scenario was replicated with 10 different random seeds in simulations; half were used for training the model and half were used for testing its detection performance. The final data included the averaged measurements at the sensor station and the lane-by-lane measurements of each detector.

The simulated scenarios are shown in figure 6. Speed-reduction zones and signal lights were deployed to simulate incident scenarios. For example, the speed limit under normal traffic conditions was set at 65 mph. When the incident occurs, one or two lanes are blocked, and the other lanes will also reduce speed through the use of speed-reduction zones. Vehicles will pass through the speed-reduction zone with a speed limit of 35 mph. Thus, vehicles in the blocked lanes have a chance to change lanes to avoid being stopped near the incident location. The ratio of truck traffic to passenger vehicles was set at 1:9. Before incident occurrence, trucks were assumed to not be allowed to use the leftmost lane. After incident occurrence, trucks could go through each available lane.



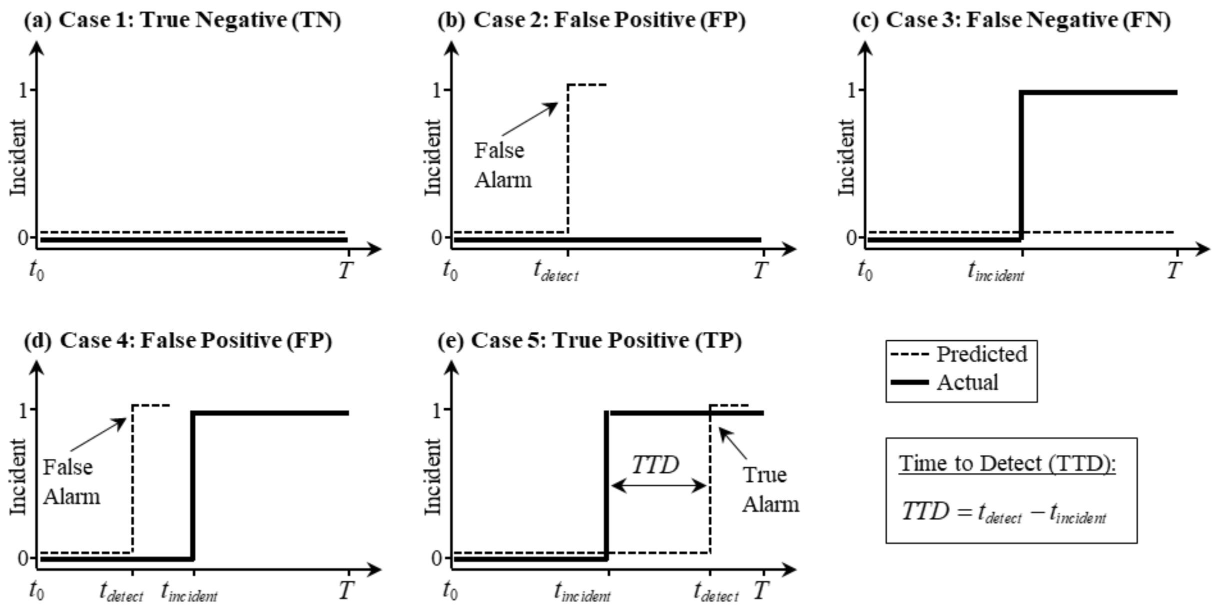
Source: Screenshot created by FHWA using VISSIM software by PTV Group®. VISSIM is the intellectual property of PTV Group and is used herein under the license purchased by Old Dominion University.

Figure 6. Illustration. Simulation in VISSIM.

PERFORMANCE CRITERIA

To evaluate the performance of incident-detection approaches, four measurements that construct the confusion matrix were considered: true negative: when there is no incident, an algorithm predicts no incident (figure 7-A); false positive (FP): when there is no incident, an algorithm predicts incident occurrence (figure 7-B and figure 7-D); false negative (FN): when there is an incident, an algorithm predicts no incident (figure 7-C); and true positive (TP): when there is an incident, an algorithm predicts its occurrence (figure 7-E). The dotted lines denote predicted incident conditions, and the solid lines denote actual incident conditions. In practice, the two cases involving FP predictions are critical because these false alarms will incorrectly report the situations that need incident clean efforts. Dispatching resources (e.g., responders) to these false alarms will greatly increase the incident-management cost. Thus, the incident-detection algorithm that produces fewer FPs is preferred. In addition, the number of FNs should be as small as possible so that the algorithm will not miss many actual incidents. Also, a good algorithm should detect an actual incident as early as possible. Thus, the time to detect (*TTD*) the actual incident was also considered. *TTD* is defined as the time difference between the actual incident occurrence time ($t_{incident}$) and the reported occurrence time by the algorithm (t_{detect}), as shown in equation 12:

$$TTD = t_{detect} - t_{incident} \tag{12}$$



Source: FHWA.

Figure 7. Graphs. Possible prediction results.

For comparisons, incident-detection algorithms were implemented in the study period (i.e., 10 min in each simulated scenario). If an algorithm reported the occurrence of an incident (either TP or FP), it was terminated for evaluation in the remaining period. Otherwise, it was run until the end of the study period. This allowed all algorithms under evaluation to have the same time horizon and fair testing scenarios. Besides the *TTD* for each tested scenario, the project team calculated two frequently used indicators to quantify the overall performance of compared incident-detection algorithms: the detection rate (*DR*) and false alarm rate (*FAR*), as shown in equation 13 and equation 14:

$$DR = \frac{\#TP}{\#\,Actual\ Incidents} \times 100\% \quad (13)$$

$$FAR = \frac{\#FP}{\#Tested\ Cases} \times 100\% \quad (14)$$

DR is the ratio of the total number of correctly detected incidents (TPs) to the total number of actual incidents. This indicator reflects the algorithm's accuracy at detecting actual incidents. A larger value of *DR* suggests that the algorithm is capable of reporting more incidents after their occurrence than algorithms with a smaller value of *DR*. *FAR* is the ratio of the total number of false alarm cases (FPs) to the total number of tested cases. Algorithms with smaller *FAR*s are preferred because of their lower number of false alarms. These performance indicators are reported based on each level of sensor spacing (DL_{12}). In this project, the total number of tested cases at a given level of DL_{12} is as shown in equation 15:

$$\#Tested\ Cases = 2\,levels\,(Lane\,closure) \times 3\,levels\,(IL_1) \times 30\,levels\,(Demand) \times 5\,(Random\ seeds) = 900 \quad (15)$$

Because the experiment did not simulate incident-free scenarios, the total number of actual incidents is the same as the total number of tested cases at a given level of DL_{12}. If some simulation scenarios without incidents were added, these two numbers would be different. The average *TTD* for each scenario running with five random seeds was calculated, and the overall average *TTD* under different flow conditions at a given level of DL_{12} was also computed.

CHAPTER 6. RESULTS

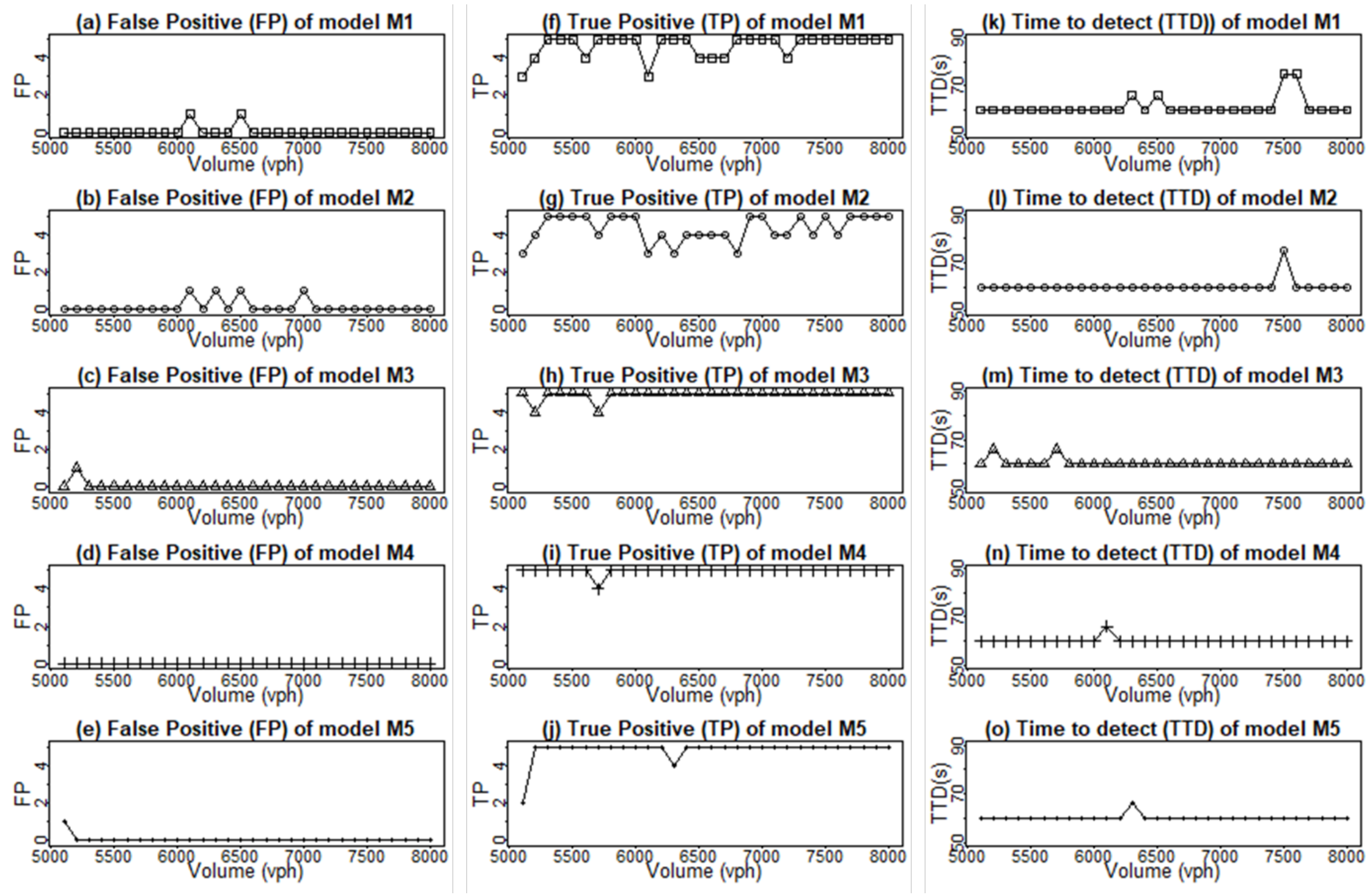
The designed experiments were implemented in VISSIM to collect the raw data. The output data from the loop detectors as well as the derived traffic metrics were organized to meet the input need of each incident-detection model. This project used the R program to develop the script for testing models with collected loop detector data. As mentioned in chapter 5, 50 percent of the data were used to train each model, and the other 50 percent were used to test the tuned model. For CA No. 7, the combination of three threshold parameters was numerically searched within the range of [0,1] with an interval of 0.05. However, because the optimal FAR and TTD cannot be guaranteed with one combination of threshold parameters at the same time, the research team selected two submodels of CA No. 7. Specifically, the model CA No. 7-1 (M1) was tuned with the aim of minimizing FAR and the model CA No. 7-2 (M2) was tuned with the aim of minimizing TTD. Additional selection criteria, such as the total number of false alarms must be less than 150 and TTD cannot exceed 180 s, were applied. For comparisons, this project also included the classic ANN model (M3) built using averaged sensor data and the AI-Avg. model (M4), which combines the aforementioned memory unit and tuned ANN model. Both M3 and M4 used one hidden layer with 20 neurons. Finally, the AI-Lane model (M5) with lane-based detector data used one hidden layer with 40 neurons. M3, M4, and M5 were also tuned by following the necessary calibration steps. All five models were trained and tested with data aggregated in 30-s time intervals.

The test results are shown in table 2. Overall, the proposed AI-based approaches that used either lane-based detector data (AI-Lane) or averaged sensor data (AI-Avg.) outperformed the CA No. 7-1 and CA No. 7-2 models in terms of DR, FAR, and TTD. Although the TTDs of the AI-based approaches were similar to those of the ANN model, the AI-based models had higher DRs and lower FARs compared with the ANN model. These results suggest that there are benefits to including the learning structure in the proposed detection framework because it can help correct some false alarms from the initial neural network algorithm.

Table 2. Average predictive performance without specifying incident locations and traffic demands.

Lane Closure	Method	DR (%)			FAR (%)			TTD (s)		
		***DL*₁₂ = 0.3 mi**	***DL*₁₂ = 0.5 mi**	***DL*₁₂ = 1.0 mi**	***DL*₁₂ = 0.3 mi**	***DL*₁₂ = 0.5 mi**	***DL*₁₂ = 1.0 mi**	***DL*₁₂ = 0.3 mi**	***DL*₁₂ = 0.5 mi**	***DL*₁₂ = 1.0 mi**
One lane	CA No. 7-1	84.0	61.6	46.0	0.7	4.7	0.2	97.5	91.5	127.2
	CA No. 7-2	76.2	56.7	42.9	2.9	10.0	0.7	90.8	84.5	130.1
	ANN	99.5	97.3	77.8	0.2	2.0	0.9	66.7	70.2	90.4
	AI-Avg.	99.7	98.4	78.0	0.0	0.9	0.7	66.7	70.2	90.4
	AI-Lane	99.7	98.2	82.4	0.2	0.9	0.0	65.2	83.8	86.3
Two lanes	CA No. 7-1	90.9	78.7	67.3	0.4	5.5	0.9	90.9	93.6	129.5
	CA No. 7-2	85.6	78.0	65.8	3.1	8.7	0.4	87.7	92.2	130.0
	ANN	99.3	97.1	93.3	0.4	2.2	0.9	69.6	70.7	95.7
	AI-Avg.	99.8	98.4	93.6	0.0	0.9	0.7	69.7	70.7	95.6
	AI-Lane	98.9	97.8	98.9	0.2	0.7	0.2	67.1	75.5	88.0

To understand the sensitivity of the proposed approach with respect to traffic volume changes, incident locations, and sensor spacing, the project team further investigated TPs, FPs, and TTD under each simulated scenario. Figure 8 shows the results from one such scenario, and the results of the other scenarios are provided in the appendix. These scenarios include different combinations of traffic volumes (from 5,100 to 8,000 vph in increments of 100), sensor gaps (0.3, 0.5, and 1 mi), and incident locations (near the upstream sensor, in the middle, and near the downstream sensor). Different marks denote the different models (M1 to M5). The performance metrics (TPs, FPs, and TTD) were affected by the prevailing traffic flows, sensor spacing, and incident locations. In particular, when an incident occurred in the middle of two loop detectors, incident detection was challenging for all models. Nevertheless, the proposed AI-Lane model was able to capture more incidents with relatively shorter TTD even under lower volume conditions compared with ANN, CA No. 7-1, and CA No. 7-2. When the sensor spacing increased, each model tended to have more FN predictions. However, among the compared algorithms, the proposed AI-Lane and AI-Avg. approaches still produced better results in terms of FNs and TTD.



Source: FHWA.

Figure 8. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (0.3 mi, near upstream).

CHAPTER 7. FUTURE WORK

The present study explored the feasibility of using AI approaches to detect incidents. Although the proposed approach is promising, the following limitations still exist, so future research is needed to improve the overall incident-detection performance.

First, the current study did not use real-world data to validate the AI-based TID approach. Thus, assessments with field data are recommended. Field data are prone to issues associated with accuracy and missing values, and it is unclear how these issues might affect the TID performance of the proposed approach. Thus, it will be meaningful to explore the performance of the proposed approach with real-world data. If more simulation studies are performed, calibrating simulation models based on field data can help generate more realistic output to support testing of the AI-based TID approach.

In addition, other types of input data can be used to improve TID performance. This project used only loop detector measurements such as flow, occupancy, and speed, whereas previous studies have introduced contributing factors such as time information and weather conditions to enhance TID approaches. For example, taking the traffic flow and time periods into consideration can better address the fact that the same incident can lead to different impacts during peak hours and off-peak hours. In addition, seasonal effects, events, and holidays also offer valuable information. Therefore, exploring suitable ways to incorporate heterogeneous input data into the AI-based approach is also important. It is reasonable to expect that more vehicle trajectory information in the context of connected vehicles will be available in the near future. Questions such as how to use information such as headways, acceleration speed, and lane-change behaviors to improve TID performance require more research.

Last but not least, different learning algorithms, especially DL approaches, should be examined to improve TID performance and better meet the needs of more scenarios. For example, previous studies proved the correlation between traffic states and incident occurrence. Thus, it is possible to stack historical traffic information using LSTM or convolutional neural network (CNN) approaches to capture such correlations. Nevertheless, TMC operators need to wisely make the tradeoff between model complexity and TID performance. For example, the AI-based framework in the current study is relatively simple and expected to be further improved if complex learning algorithms such as CNN and LSTM are incorporated into the framework. But the benefits of deep neural network models are accompanied by their high computational complexity and difficulties for real-time applications. Therefore, more efforts are needed to assess the suitability of various learning algorithms for incident detection.

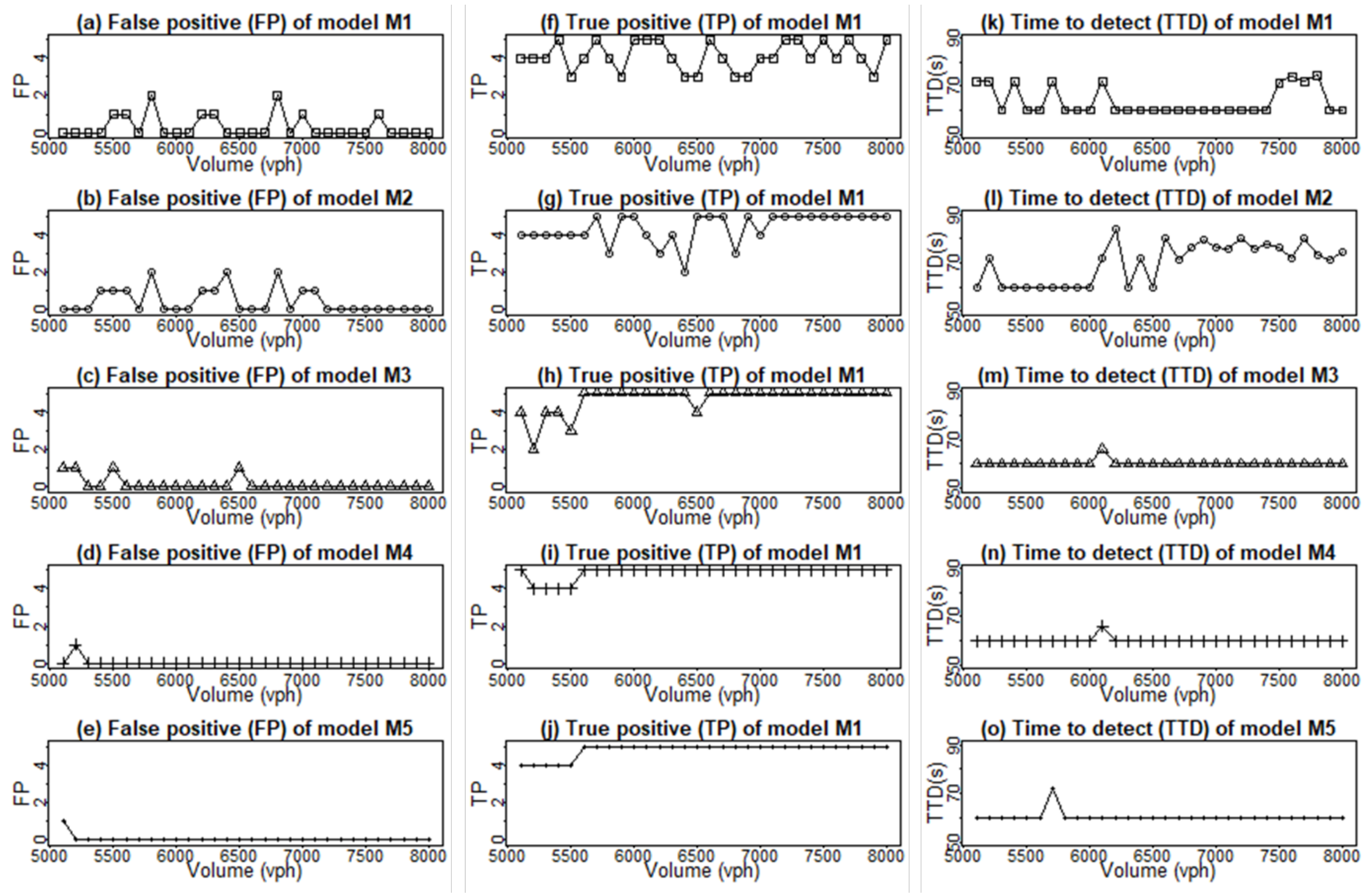
CHAPTER 8. CONCLUSION

TID is critical for highway operations and safety. A number of automatic TID approaches based on loop detector data have been developed in the past, but the performance of these approaches is not adequate, which prohibits their widespread implementation. Considering the complexity and the nonlinear relationship between traffic incidents and traffic metrics, this project proposed an AI-based incident-detection framework that leverages data from loop detectors and typical learning features of expert systems. The key components of the proposed framework include the ensemble of an ML approach and the memory unit. The use of the memory unit in terms of the knowledge database coupled with the similarity analysis between current and historical traffic profiles enables corrections to initial incident predictions. The use of the memory unit ultimately offers a detection framework with learning and evolving capabilities that benefits from similar historical errors. The developed AI-based framework was assessed through a fully controlled simulation experiment that consisted of numerous traffic and incident scenarios. The obtained results show that the proposed AI-based framework (using either lane-based data or station-level average data) performs better than the two classic approaches (CA No. 7 and the ANN). The better performance of the proposed approach was primarily demonstrated in terms of shorter TTD, lower FAR, and higher DR. The presented analysis results confirm the improved performance of the proposed AI-based framework regardless of the sensor spacing.

The current study emphasizes the architecture of the AI-based framework. This project did not focus on assessing whether the involved individual models (i.e., neural network and KNN in this study) are superior to other models such as SVM or DL approaches. These integrated algorithms can be replaced by other models without changing the proposed framework. For example, if an SVM model is adopted as the initial classification algorithm, it can replace the neural network component in the framework. Because of the inherent limitations of simulation models, some of the simulated data from this project might not perfectly reflect field situations. Given the availability of sufficient traffic data and accurate incident records, further testing the performance of the AI-based framework with field data in the future is recommended.

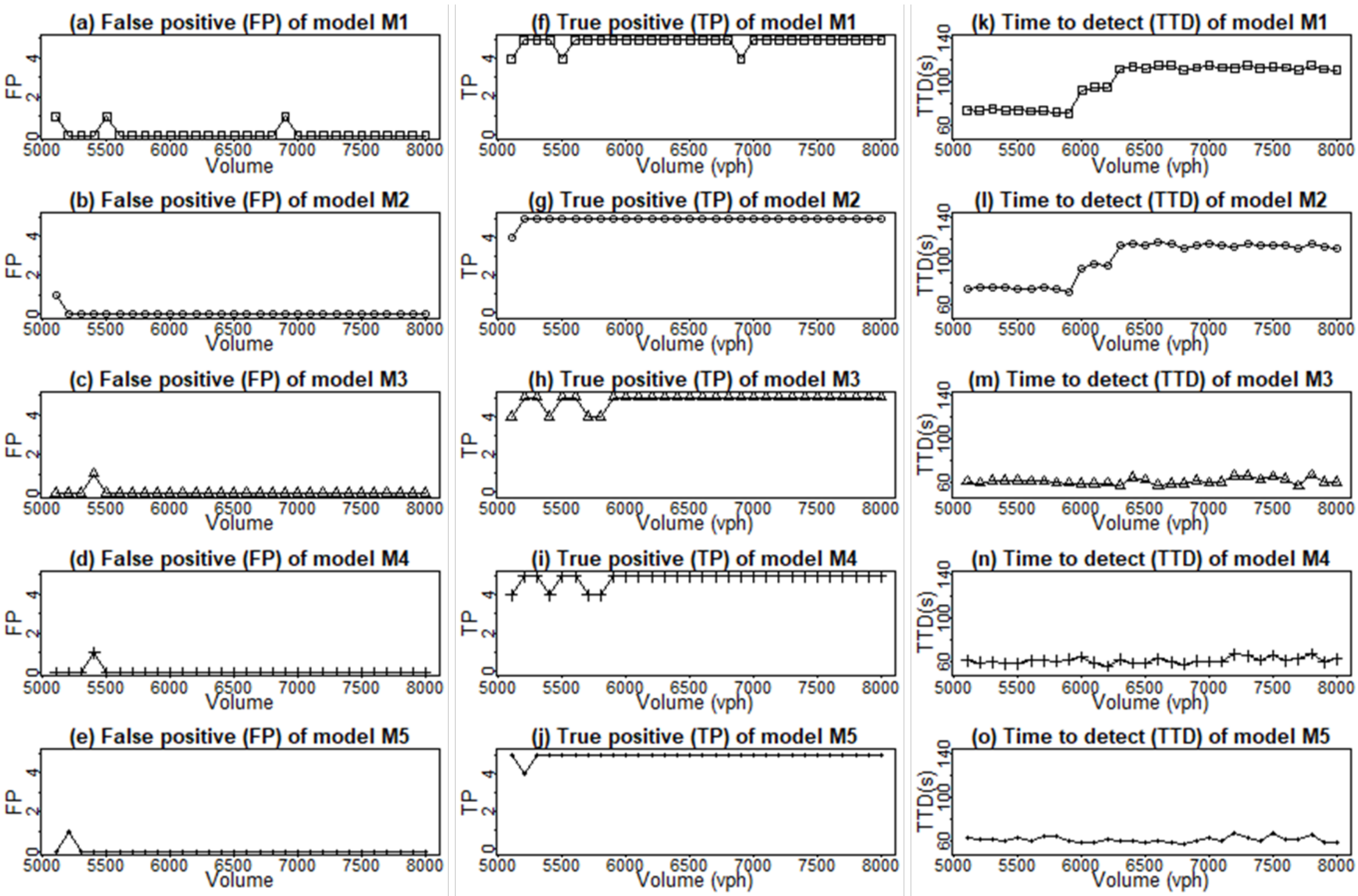
APPENDIX. PERFORMANCE COMPARISONS OF DIFFERENT MODELS UNDER DIFFERENT SCENARIOS

Figure 9 through figure 16 describe the performance comparison (FP, TP, and TTD) of five different models (CA No. 7-1, CA No. 7-2, ANN, AI-Avg., and AI-Lane) under different conditions (loop detector interval and incident location).



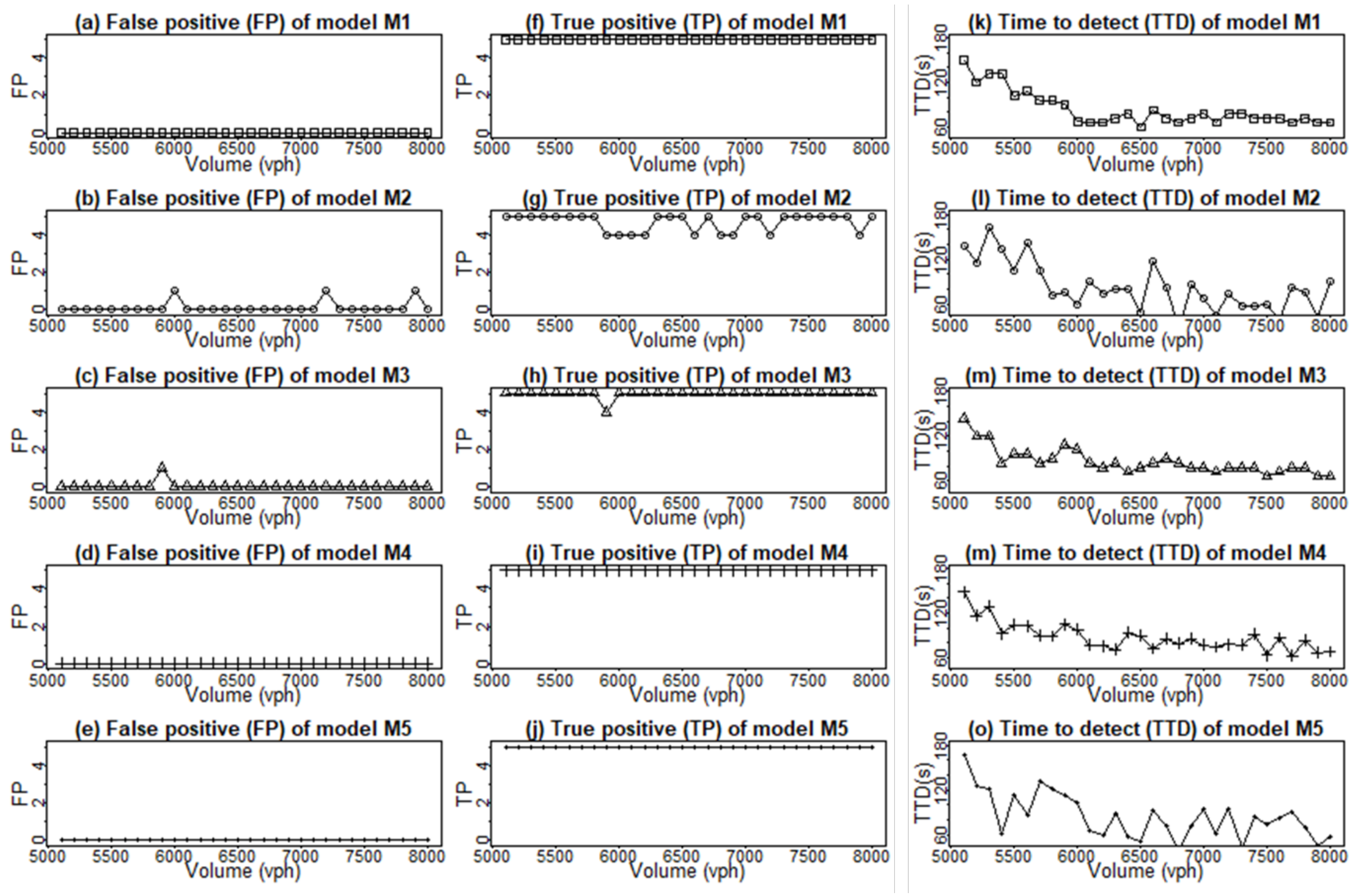
Source: FHWA.

Figure 9. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (0.5 mi, near upstream).



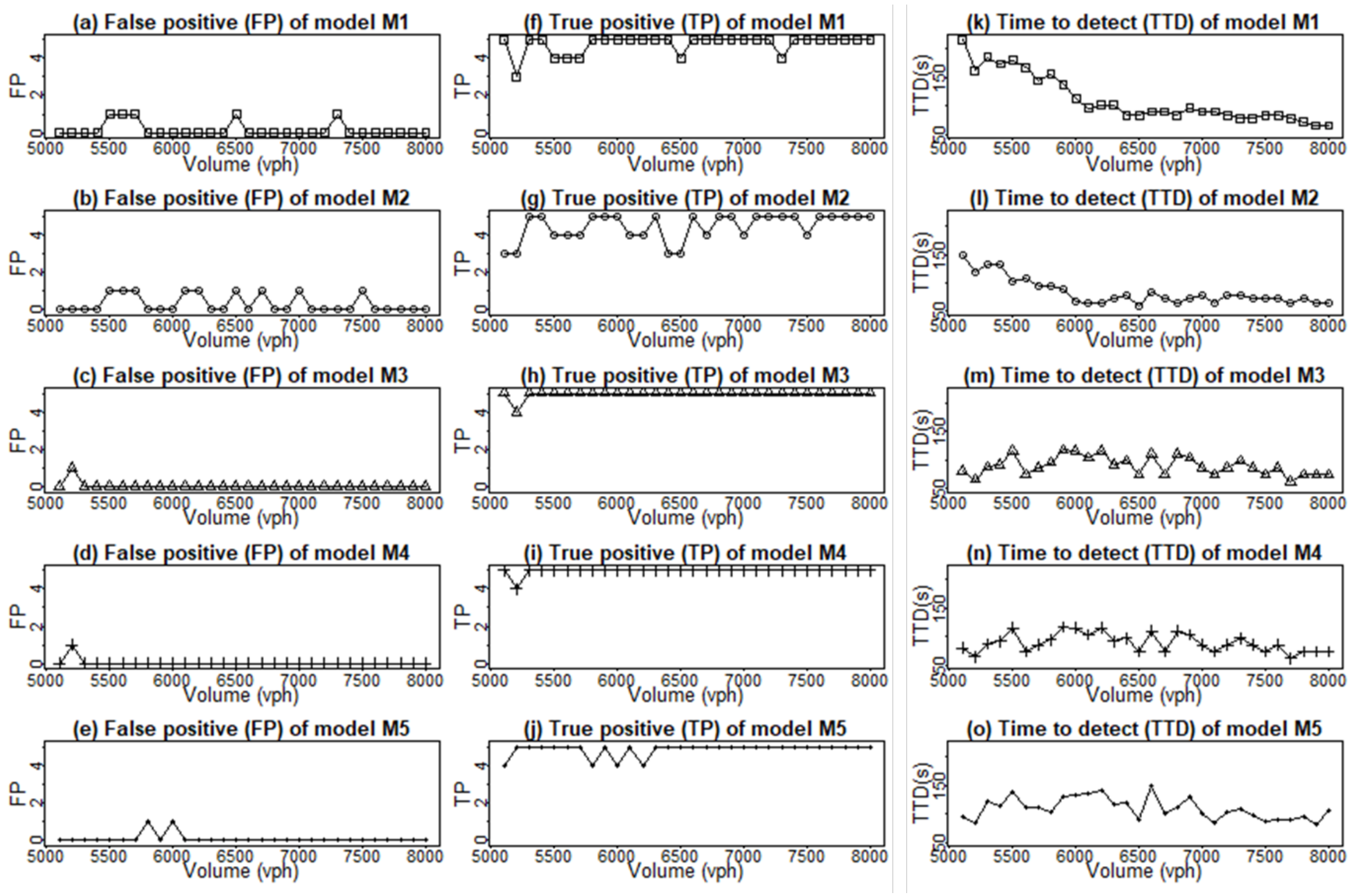
Source: FHWA.

Figure 10. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (1.0 mi, near upstream).



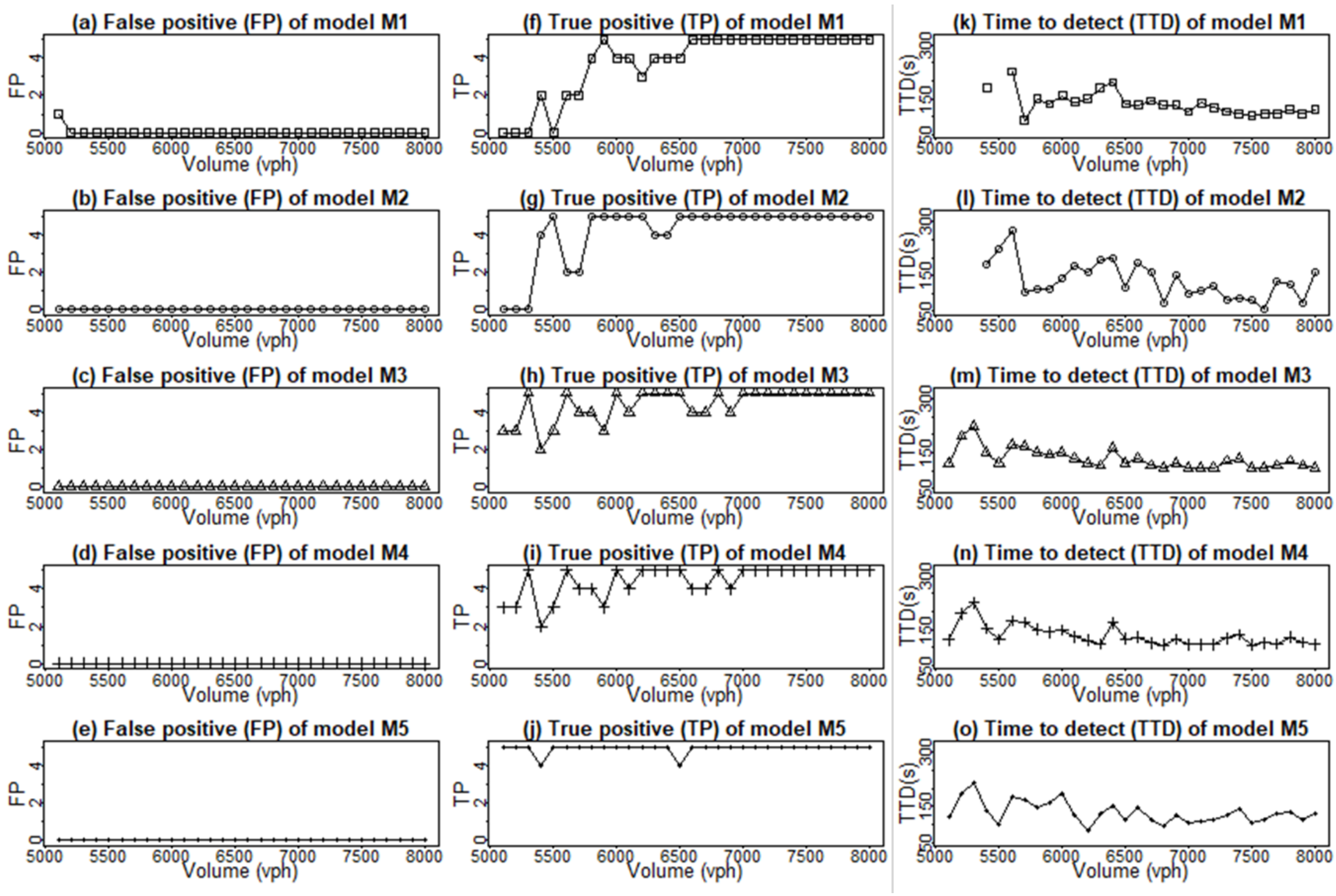
Source: FHWA.

Figure 11. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (0.3 mi, near middle).



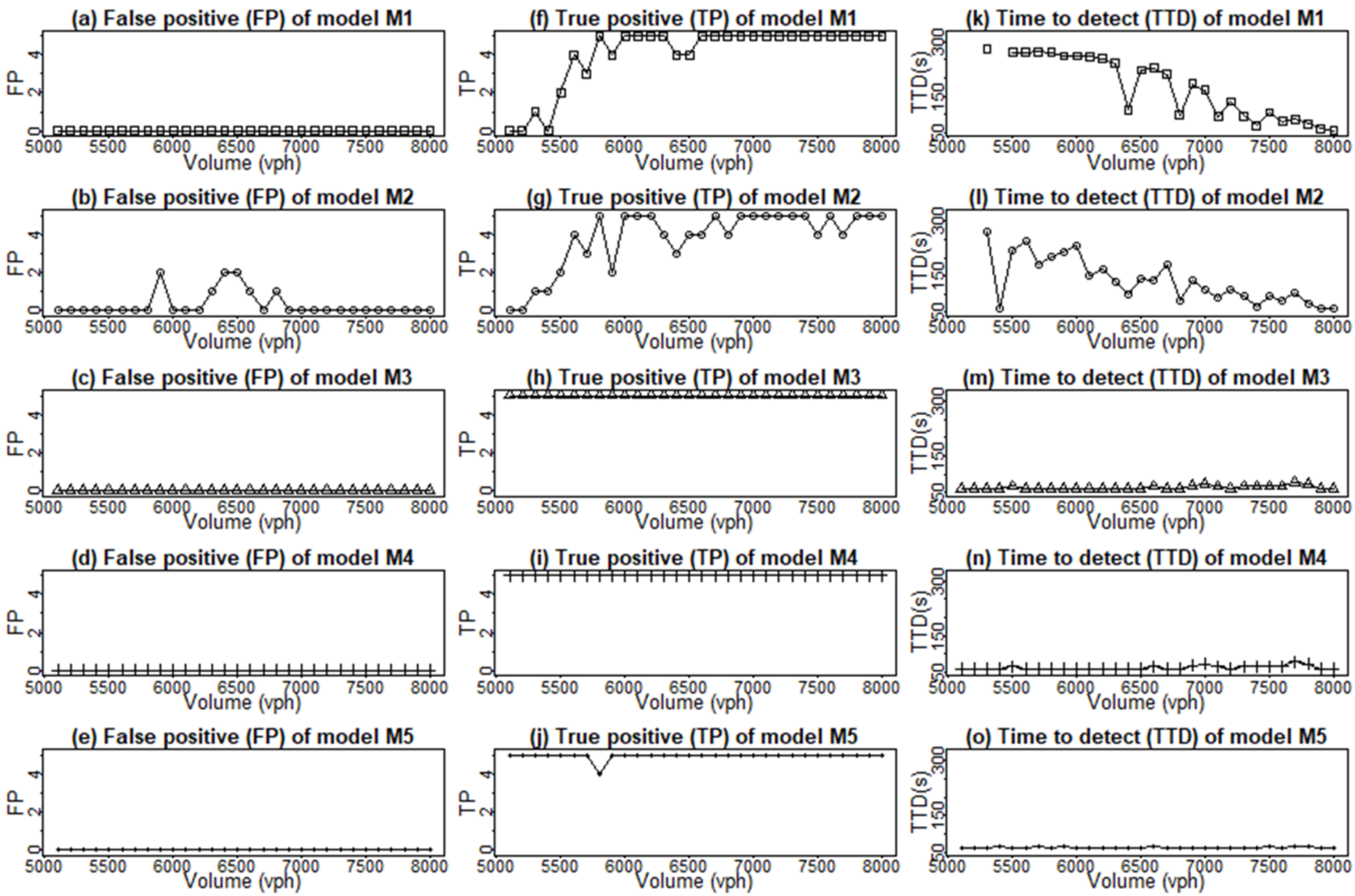
Source: FHWA.

Figure 12. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (0.5 mi, near middle).



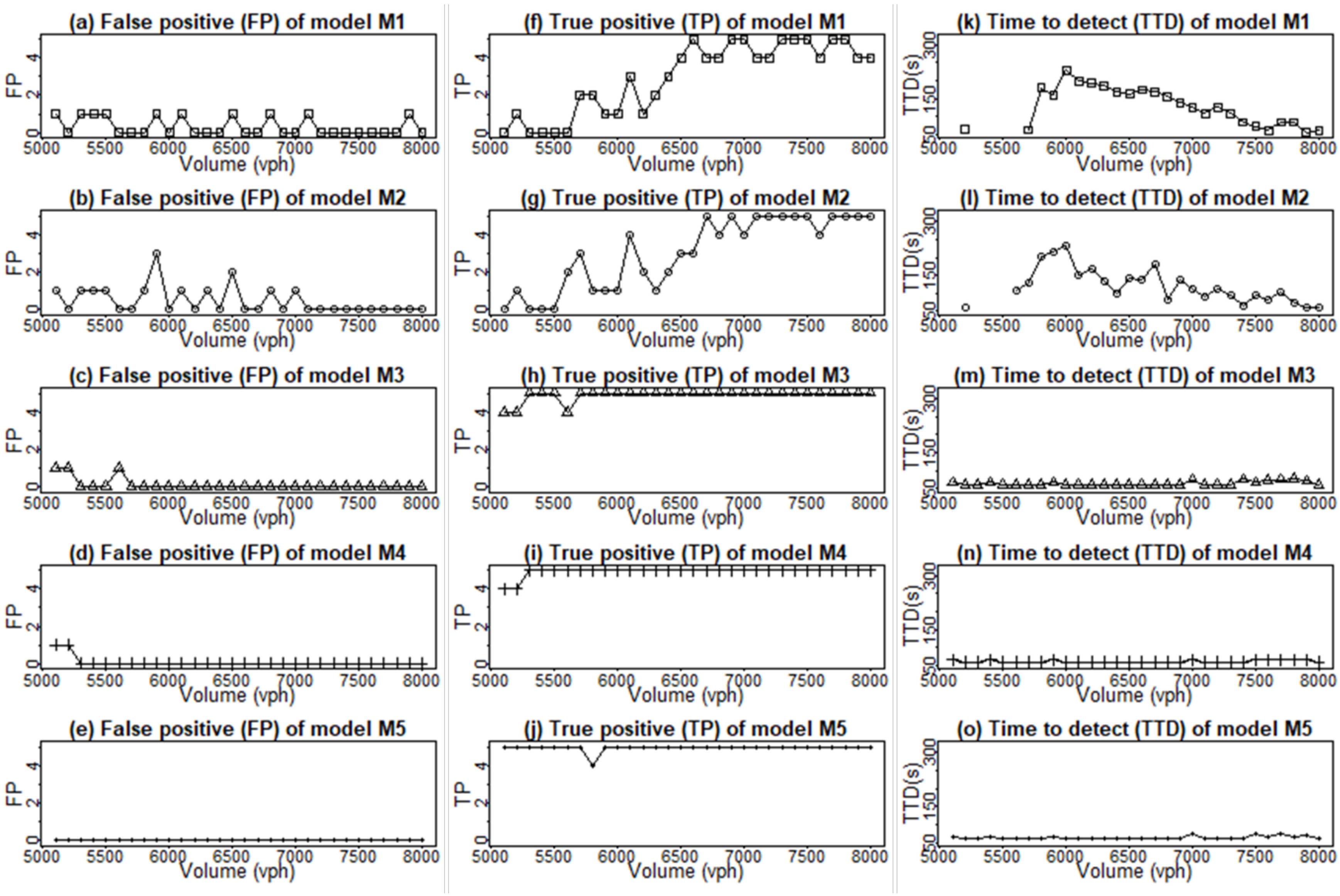
Source: FHWA.

Figure 13. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (1.0 mi, near middle).



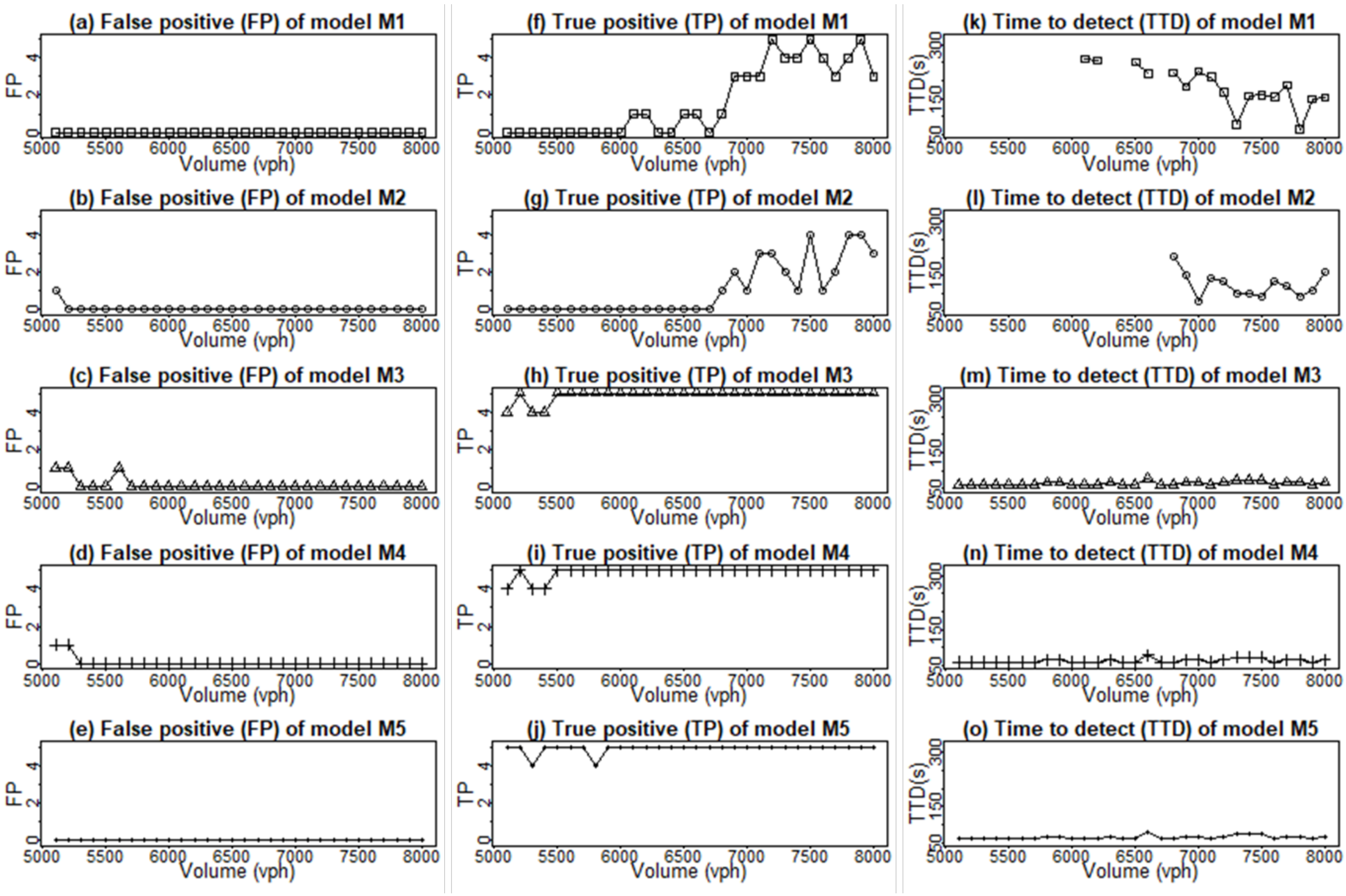
Source: FHWA.

Figure 14. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (0.3 mi, near downstream).



Source: FHWA.

Figure 15. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (0.5 mi, near downstream).



Source: FHWA.

Figure 16. Graphs. Performance comparison of different models in terms of FPs, TPs, and TTD (1.0 mi, near downstream).

REFERENCES

Adeli, H. and Samant, A. (2000). “An Adaptive Conjugate Gradient Neural Network–Wavelet Model for Traffic Incident Detection.” *Computer-Aided Civil and Infrastructure Engineering 15*(4), pp. 251–260, John Wiley & Sons, Inc., Hoboken, NJ.

Ahmed, M.S. and Cook, A. R. (1979). “Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques.” *Transportation Research Record 722*, pp. 1–9, Transportation Research Board, Washington, DC.

Ahmed, S.A. and Cook, A.R. (1980). “Time Series Models for Freeway Incident Detection.” *Journal of Transportation Engineering 106*(6), pp. 731–745, American Society of Civil Engineers, Reston, VA.

Ahmed, S.A. and Cook, A. R. (1982). “Application of Time-Series Analysis Techniques to Freeway Incident Detection.” *Transportation Research Record 841*, pp. 19–21, Transportation Review Board, Washington, DC.

Asakura, Y., Kusakabe, T., Nguyen, L.X., and Ushiki, T. (2017). “Incident Detection Methods Using Probe Vehicles with On-Board GPS Equipment.” *Transportation Research Part C: Emerging Technologies 81*, pp. 330–341, Elsevier, Amsterdam, Netherlands.

Brynjolfsson, E. and McAfee, A. (2012). “Winning the Race with Ever-Smarter Machines.” *MIT Sloan Management Review 53*(2), p. 53, Massachusetts Institute of Technology, Cambridge, MA.

Cheu, R.L. and Ritchie, S.G. (1995). “Automated Detection of Lane-Blocking Freeway Incidents Using Artificial Neural Networks.” *Transportation Research Part C: Emerging Technologies 3*(6), pp. 371–388, Elsevier, Amsterdam, Netherlands.

Collins, J., Hopkins, C., and Martin, J. (1979). *Automatic Incident Detection – TRRL Algorithms HIOCC and PATREG*, Supplementary Report 526, Transport and Road Research Laboratory, Crowthorne, Berkshire, UK.

Dia, H. and Rose, G. (1997). “Development and Evaluation of Neural Network Freeway Incident Detection Models Using Field Data.” *Transportation Research Part C: Emerging Technologies 5*(5), pp. 313–331, Elsevier, Amsterdam, Netherlands.

Dudek, C.L., Messer, C.J., and Nuckles, N.B. (1974). “Incident Detection on Urban Freeways.” *Transportation Research Record 495*, pp. 12–24, Transportation Research Board, Washington, DC.

El Hatri, C. and Boumhidi, J. (2018). “Fuzzy Deep Learning Based Urban Traffic Incident Detection.” *Cognitive Systems Research 50*, pp. 206–213, Elsevier, Amsterdam, Netherlands.

Ferrucci, D., Levas, A., Bagchi, S., Gondek, D., and Mueller, E.T. (2013). “Watson: Beyond Jeopardy!” *Artificial Intelligence 199*, pp. 93–105, Elsevier, Amsterdam, Netherlands.

Fries, R., Hamlin, C., Chowdhury, M., Ma, Y., and Ozbay, K. (2012). “Operational Impacts of Incident Quick Clearance Legislation: A Simulation Analysis.” *Journal of Advanced Transportation 46*(1), pp. 1–11, Hindawi, London, UK.

Gu, Y., Qian, Z.S., and Chen, F. (2016). “From Twitter to Detector: Real-Time Traffic Incident Detection Using Social Media Data.” *Transportation Research Part C: Emerging Technologies 67*, pp. 321–342, Elsevier, Amsterdam, Netherlands.

Hellinga, B. and Knapp, G. (2000). “Automatic Vehicle Identification Technology-Based Freeway Incident Detection.” *Transportation Research Record 1727*(1), pp. 142–153, Transportation Research Board, Washington, DC.

Jin, X., Srinivasan, D. and Cheu, R.L. (2001). “Classification of Freeway Traffic Patterns for Incident Detection Using Constructive Probabilistic Neural Networks.” *IEEE Transactions on Neural Networks 12*(5), pp. 1173–1187, Institute of Electrical and Electronics Engineers, New York, NY.

Kinoshita, A., Takasu, A., and Adachi, J. (2015). “Real-Time Traffic Incident Detection Using a Probabilistic Topic Model.” *Information Systems 54*, pp. 169–188, John Wiley & Sons, Inc., Hoboken, NJ.

Lee, D.-H., Wang, H., Cheu, R., and Teo, S. (2004). “Taxi Dispatch System Based on Current Demands and Real-Time Traffic Conditions.” *Transportation Research Record 1882*, pp. 193–200, Transportation Research Board, Washington, DC.

Li, D., Hu, X., Jin, C.-j., and Zhou, J. (2017). “Learning to Detect Traffic Incidents from Data Based on Tree Augmented Naive Bayesian Classifiers.” *Discrete Dynamics in Nature and Society 2017*, 8523495, Hindawi, London, UK.

Li, M.-h., Chen, S.-y., and Lao, Y.-c. (2016). “Automatic Incident Detection Algorithm Based on Under-Sampling for Imbalanced Traffic Data.” *Proceedings of the 2016 International Conference on Green Building, Materials and Civil Engineering*, April 26–27, 2016, CRC Press, Hong Kong, China.

Liu, Y., Lei, Y., Yi, Q., Jianquan, W., and Huimin, W. (2008). “Traffic Incident Detection Algorithm for Urban Expressways Based on Probe Vehicle Data.” *Journal of Transportation Systems Engineering and Information Technology 8*(4), pp. 36–41, Elsevier, Amsterdam, Netherlands.

Masters, P.H., Lam, J.K., and Wong, K. (1991). “Incident Detection Algorithms for COMPASS-An Advanced Traffic Management System,” *Vehicle Navigation and Information Systems Conference*, October 20–23, 1991, Troy, MI, Institute of Electrical and Electronics Engineers, New York, NY.

Michalopoulos, P.G. (1991). "Incident Detection Through Video Image Processing." *Applications of Advanced Technologies in Transportation Engineering*, Second International Conference, August 18–21, 1991, Minneapolis, MN, American Society of Civil Engineers, New York, NY.

Michalopoulos, P.G., Jacobson, R.D., Anderson, C.A., and DeBruycker, T.B. (1993). "Automatic Incident Detection Through Video Image Processing." *Traffic Engineering & Control 34*(2), pp. 66–75, Hemming Group, London, UK.

Nguyen, H., Liu, W., Rivera, P., Cai, C., Yee, D., and Chen, F. (2015). "TrafficWatch: Real-Time Traffic Incident Detection and Network Monitoring Using Social Media." Presented at 22nd Intelligent Transport System World Congress Proceedings, October 5–9, 2015, Bordeaux, France.

Parkany, E. and Xie, C. (2005). *A Complete Review of Incident Detection Algorithms & Their Deployment: What Works and What Doesn't*, Report NETCR 37, New England Transportation Consortium, Storrs, CT.

Payne, H.J. and Tignor, S.C. (1978). "Freeway Incident-Detection Algorithms Based on Decision Trees with States." *Transportation Research Record 682*, pp. 30–37, Transportation Research Board, Washington, DC.

Rensel, E., Yorks, C., James, R., Robinson, E., and Motamed, M. (2018). *Further Assessments of Safe, Quick Clearance Strategies, Phase II*, Report SPR-1655, Michigan Department of Transportation, Lansing, MI.

Saifuzzaman, M., Djukic, T., Hernandez-Potiomkin, Y., Mena-Yedra, R., Bert, E., and Casas, J. (2018). "Understanding Incident Impact on Traffic Variables to Reduce False Incident Detection." *Australasian Transport Research Forum Proceedings*, October 30–November 1, 2018, Darwin, Northern Territory, Australia.

Salas, A., Georgakis, P., and Petalas, Y. (2017). *Incident Detection Using Data from Social Media*. Presented at ITSC 2017: IEEE International Conference on Intelligent Transportation Systems, October 16–20, 2017, Yokohama, Japan.

Samant, A. and Adeli, H. (2000). "Feature Extraction for Traffic Incident Detection Using Wavelet Transform and Linear Discriminant Analysis." *Computer-Aided Civil and Infrastructure Engineering 15*(4), pp. 241–250, John Wiley & Sons, Inc., Hoboken, NJ.

Sattayhatewa, P.R. (1999). "Arterial Incident Detection: Applying CUSUM Chart Method." *Traffic Engineering & Control 40*(12), pp. 582–585, Hemming Group, London, UK.

Stephanedes, Y.J. and Chassiakos, A.P. (1993). "Freeway Incident Detection Through Filtering." *Transportation Research Part C: Emerging Technologies 1*(3), pp. 219–233, Elsevier, Amsterdam, Netherlands.

Teng, H. and Qi, Y. (2003). "Detection-Delay-Based Freeway Incident Detection Algorithms." *Transportation Research Part C: Emerging Technologies 11*(3-4), pp. 265–287, Elsevier, Amsterdam, Netherlands.

Tsai, J. and Case, E. (1979). "Development of Freeway Incident-Detection Algorithms by Using Pattern-Recognition Techniques." *Transportation Research Record 722*, pp. 113–116, Transportation Research Board, Washington, DC.

USDOT. (2006). *National Strategy to Reduce Congestion on America's Transportation Network*, U.S. Department of Transportation, Washington, DC, https://www.heartland.org/_template-assets/documents/publications/21283.pdf, last accessed Sep 10, 2020.

USDOT. (2019). "Operations Story." (website). Available online: https://ops.fhwa.dot.gov/aboutus/opstory.htm, last accessed June 15, 2019.

Wang, R., Work, D.B., and Sowers, R. (2016). "Multiple Model Particle Filter for Traffic Estimation and Incident Detection." *IEEE Transactions on Intelligent Transportation Systems 17*(12), pp. 3461–3470, Institute of Electrical and Electronics Engineers, New York, NY.

Wen, H., Yang, Z., Jiang, G., and Shao, C. (2001). "A New Algorithm of Incident Detection on Freeways." *Proceedings of the IEEE International Vehicle Electronics Conference*, September 25–28, 2011, Tottori, Japan, Institute of Electrical and Electronics Engineers, New York, NY.

Williams, B.M. and Guin, A. (2007). "Traffic Management Center Use of Incident Detection Algorithms: Findings of a Nationwide Survey." *IEEE Transactions on Intelligent Transportation Systems 8*(2), pp. 351–358, Institute of Electrical and Electronics Engineers, New York, NY.

Yang, H., Wang, Z., Xie, K., Ozbay, K., and Imprialou, M. (2018). "Methodological Evolution and Frontiers of Identifying, Modeling and Preventing Secondary Crashes on Highways." *Accident Analysis & Prevention 117*, pp. 40–54, Association for the Advancement of Automotive Medicine, Chicago, IL.

Yao, B., Hu, P., Zhang, M., and Jin, M. (2014). "A Support Vector Machine with the Tabu Search Algorithm for Freeway Incident Detection." *International Journal of Applied Mathematics and Computer Science 24*(2), pp. 397–404, Zielona Góra, Poland.

Zhang, Z., He, Q., Gao, J., and Ni, M. (2018). "A Deep Learning Approach for Detecting Traffic Accidents from Social Media Data." *Transportation Research Part C: Emerging Technologies 86*, pp. 580–596, Elsevier, Amsterdam, Netherlands.

Zhao, M., Chen, X., and Sun, D. (2018). "An Incident Detection Method Considering Meteorological Factor with Fuzzy Logic." *Engineering Review 38*(1), pp. 104–114, Rijeka, Croatia.

Recycled
Recyclable

HRDO-10/06-22(WEB)E