DETECTION OF TRAFFIC INCIDENTS USING

MACHINE LEARNING TECHNIQUES

by

Osama Mohamed ElSahly

A Dissertation presented to the Faculty of the
American University of Sharjah
College of Engineering
In Partial Fulfilment
of the Requirements
for the Degree of

Doctor of Philosophy in Engineering –
Engineering Systems Management

Sharjah, United Arab Emirates

April 2023

Declaration of Authorship

I declare that this dissertation is my own work and, to the best of my knowledge and belief, it does not contain material published or written by a third party, except where permission has been obtained and/or appropriately cited through full and accurate referencing.

Signed Osama Mohamed ElSahly

Date 17/04/2023

Approvals

We, the undersigned, approve the PhD Dissertation written by: Osama Mohamed ElSahly

Dissertation Title: DETECTION OF TRAFFIC INCIDENTS USING MACHINE LEARNING TECHNIQUES

Date of Defence: 17/04/2023

Name, Title and Affiliation	Signature
Dr. Akmal Abdelfatah Professor, Department of Civil Engineering Dissertation Advisor	
Dr. Ghassan Abu-Lebdeh Professor, Department of Civil Engineering DAC Committee Member	
Dr. Mahmoud Awad Associate Professor, Department of Industrial Engineering DAC Committee Member	
Dr. Tariq Ali Professor, Department of Civil Engineering Dissertation Examiner	
Dr. Khaled Hamad Professor, Department of Civil and Environmental Engineering, University of Sharjah Dissertation Examiner	

Accepted by:

Dr. Lotfi Ramadan

Associate Dean for Graduate Affairs and Research

College of Engineering

Dr. Mohamed El-Tarhuni
Vice Provost for Research and Graduate Studies Office of
Research and Graduate Studies

Acknowledgements

I would like to extend my heartfelt gratitude to my advisor, Dr. Akmal Abdelfatah, for his unwavering support and guidance throughout my graduate studies. His immense knowledge, motivation, patience and continuous encouragement were integral to my success. It was a privilege to work under his supervision and I am grateful for the opportunity to learn from him.

I am also thankful to the American University of Sharjah for providing me the graduate assistantship and for creating an environment that fostered my academic and professional growth.

My deepest appreciation goes to my parents for their love, prayers, care, and unwavering support throughout my life. Without their encouragement and support, I would not have been able to accomplish what I have today.

I am forever grateful to my wife for her love, care, and support. Her unwavering encouragement and understanding have been a source of strength and motivation for me.

Finally, I would like to express my sincere thanks to Dr. Hussam AlShraideh for his help in developing the AI models and his valuable time.

Thank you all for being a part of my journey.

Abstract

This dissertation proposes new models for detecting traffic incidents on freeways using machine learning algorithms to classify traffic data collected from the freeway system. These models are generic and consider multiple factors that affect incident detectability simultaneously. The models were trained and tested on simulated traffic data that represent normal and incident conditions using the well-known microsimulation software VISSIM. The proposed models, which include the Random Forest (RF) and Multilayer Feedforward Artificial Neural Network (MLF), consider four factors: the congestion level, the distance between the upstream and downstream detector stations, the location of the incident relative to the detector stations, and the severity of the incident. The results showed that the developed models achieved excellent performance, surpassing existing models in the literature. During training, the MLF model achieved a detection rate (DR) of 95.96%, a mean time to detect (MTTD) of 0.89 minutes, and a false alarm rate (FAR) of 1.01%. During testing, the MLF model achieved a DR of 100%, MTTD of 1.6 minutes, and FAR of 1.29%. Similarly, the RF model achieved a DR of 96.97%, MTTD of 1.05 minutes, and FAR of 0.62% during training, and a DR of 100%, MTTD of 1.17 minutes, and FAR of 0.862% during testing. The results revealed that incident detection systems may have difficulty detecting incidents with minor severity during low traffic volumes. The FAR decreased with the increase in the Demand to Capacity ratio (D/C), while the MTTD increased with the increase in D/C. Additionally, higher incident severity resulted in lower MTTD values, while the distance between the incident location and upstream detector had the opposite effect. The FAR decreased as the incident moved farther from the upstream detector but increased with the distance between detectors. Larger detector spacings were associated with longer detection times. The proposed models can significantly improve traffic performance on freeways, especially during incidents, and benefit both local and international transportation agencies. The study's contribution lies in developing efficient and reliable incident detection models that consider multiple variables simultaneously, improving traffic safety, reducing congestion, saving lives and properties, and reducing pollution.

Keywords: Automatic Incident detection, Machine Learning, Artificial intelligence (AI), VISSIM simulation software, Random Forest, Artificial Neural Network (ANN).

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AID	Automatic Incident Detection
ANN	Artificial Neural Network
APID	All Purpose Incident Detection
ARIMA	Autoregressive Integrated Moving-Average
ART2	Adaptive Resonance Theory
CCTV	Closed-Circuit Television system
CNN	Convolutional Neural Network
CU	Control Unit
D/C	Demand to Capacity
DES	Double exponential smoothing
DR	Detection Rate
FAR	False Alarm Rate
FCD	Floating Car Data
FIS	Fuzzy Inference System
FL	Fuzzy Logic
FLM	Fuzzy logic model
FN	False Negative
FP	False Positive
GMM	Gaussian Mixture Model
GNAs	Generative Adversarial Networks
HIOCC	High Occupancy
IOV	Internet of Vehicles
IQD	Inter Quartile Distance
LSTM	Long-and Short-Term Memory Network
ML	Machine Learning
MLF	Multilayer Feedforward Neural Network
MSE	Mean Squared Error
MTTD	Mean Time to Detect
NB	Naïve Bayes
NBTree	Naïve Bayes and decision tree
OBU	On-Board Unit

PATREG	Pattern Recognition
PeMS	Performance Measurement System
PNN	Probabilistic Neural Network
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Networks
RPN	Region Proposal Network
RSKNN	Random Subspace K Nearest Neighbour
RSU	Road-Side Unit
SASYNO	Self-Adaptive Synthetic Oversampling
SGD	Stochastic Gradient Descent
SND	Standard Normal Deviate
SOFM	Self-Organizing Feature Map
SVM	Support Vector Machine
TMC	Traffic Management Centre
TN	True Negative
TP	True Positive
TSC	Traffic Services Corporation
TTD	Time to Detect
TTI	Texas Transportation Institute
TUM	The Technical University of Munich
VANET	Vehicular Ad-Hoc Networks
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V/C	Volume-to-Capacity ratio
WHO	World Health Organization

Chapter 1. Introduction

1.1. Introduction

The continuous population and economic growth have increased the demand for the mobility of passengers and goods, which in turn has resulted in a growth of the number of vehicles on the roads. The growth in the number of vehicles is usually associated with an increase in traffic congestion. Traffic incidents are chronic issues that cause traffic congestion, increase air pollution, property damage, deaths, injuries and financial costs. Traffic incidents can be defined as non-recurring events that can be due to broken vehicles on the roads, temporary road maintenance, adverse weather conditions or vehicles' crashes that can disrupt normal traffic flow [1]–[4]. These incidents may block some of the lanes on the freeways, which reduce the capacity of the freeway section and result in more delays and increase the travel costs [4], [5]. In addition, due to the slowing down of the vehicles on the freeway and the formation of queues, upstream of the incident, the emissions of vehicles during congestions increase, which degrades ambient air quality. According to the World Health Organization (WHO), in 2021, traffic accidents cause more than one million deaths every year, and between 20 and 50 million people suffer non-fatal injuries and disabilities [6]. These traffic accidents cost 3% of the gross domestic product of most of the countries [6]. Thus, early incident detection and management is one of the crucial functions in any Traffic Management Centres (TMCs), to make timely and effective response such as dispatching of emergency services and incident management crews, control and re-routing the traffic around incident locations and provision of real-time traffic information for road users [7]–[9]. Over the past few decades, researchers have developed various incident detection methods, where they vary in their data collection technologies and data processing and detection algorithms.

1.2. Objectives

The aim of this research is to develop automated incident detection models for freeways that enhance the accuracy and speed of incident detection in order to improve traffic safety and efficiency. To achieve this aim, the following tasks were pursued:

- ✓ Develop machine learning models for incident detection that consider multiple factors simultaneously.

- ✓ Evaluate the performance of the developed incident detection models using simulated traffic data generated from VISSIM software.
- ✓ Investigate the impact of these factors on the performance of incident detection models to gain a deeper understanding of how these factors affect the behavior of AID systems.
- ✓ Compare the performance of the developed incident detection models to existing models in the literature in terms of their effectiveness and reliability.
- ✓ Identify the limitations and weaknesses of existing incident detection models in the literature and propose improvements based on the findings of this research.

1.3. Problem Statement

The negative impacts of traffic incidents are a matter of serious concern, affecting traffic operations, safety, and the economy. Consequently, there is a growing need for effective and reliable AID systems to address this problem. Despite consistent efforts to develop and enhance AID systems, persistent challenges remain in effectively and accurately detecting incidents. These challenges have prompted the need for innovative approaches to overcome them. The primary objective of this dissertation is to address these obstacles by utilizing machine learning techniques to create efficient and generic incident detection models. This dissertation stands out from previous research in that it considers four critical factors that significantly impact the performance of AID systems simultaneously. These factors include congestion levels, severity of incidents, location of occurrences, and the distance between detectors/sensors. By considering all these factors when creating detection models, this dissertation aims to produce more realistic solutions and gain a deeper understanding of how AID systems behave under varying circumstances. The developed incident detection models are designed not only for effectiveness and reliability but also to be generic and realistic by considering all possible factors known or believed to impact the performance of AID systems. The ultimate goal of this study is to contribute to the field of transportation management by improving the accuracy and speed of traffic incident detection. This, in turn, can lead to a safer and more efficient road network, mitigating the negative impacts of traffic incidents on the economy, safety, and traffic operations.

1.4. Dissertation Organization

This research is organized as follows:

- Chapter 1: provides an introduction about the impacts of traffic incidents and the importance of traffic incident detection and management. In addition, research objectives and problem statements are presented in this chapter.
- Chapter 2: discusses the previous research with a focus on the traffic interruption caused by traffic incidents and the concept of the incident timeline. In addition, different incident detection algorithms and systems are discussed and evaluated in this section.
- Chapter 3: presents the methodology that is followed to develop the proposed incident detection models.
- Chapter 4: discusses and evaluates the proposed models' results.
- Chapter 5: presents the conclusions and recommendations of this research.

Chapter 2. Background and Literature Review

The incident detection models have evolved from non-automatic systems that detect incidents through phone calls by eyewitnesses [10] to Automatic Incident Detection (AID) systems that gather parametric data from the roads and analyse it to extract traffic patterns that are used to detect normal and abnormal situations [11]. The development and testing of (AID) systems has begun since the early 1970s [12]–[17]. Several detection algorithms have been developed to automatically trigger incident alarms when the input traffic data meet certain pre-set conditions [18]. These algorithms depend on a major concept, that traffic incidents usually cause traffic anomalies in the traffic flow on freeways, thus traffic incident detection systems synchronously monitor these traffic anomalies to detect traffic incidents instantly [19]. When an incident occurs on a section of road and blocks one or more lanes, it can create a bottleneck that causes a queue of vehicles to form up from the location of the incident illustrated in Figure 1.

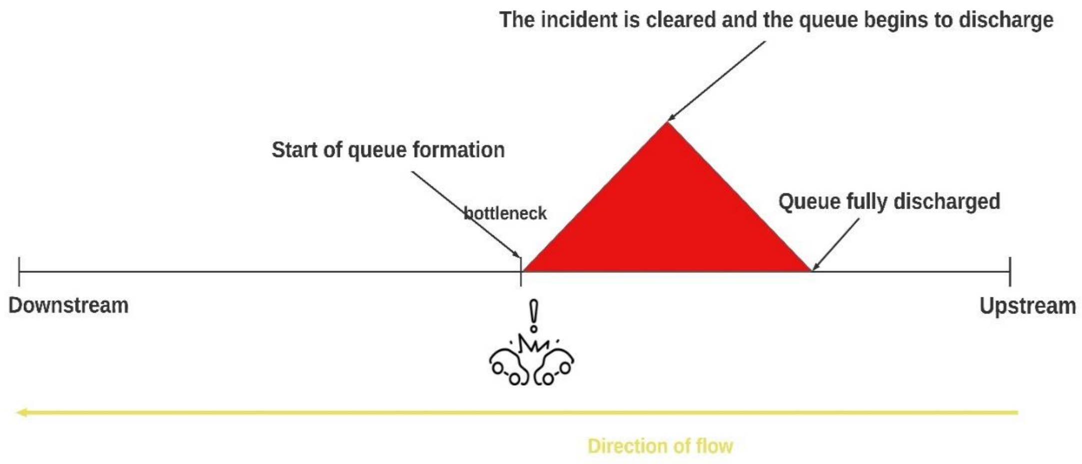

Figure 1 Roadway section under an incident [20].

This queue will propagate in the direction of the upstream section of the road, leading to different flow rates and speeds between the upstream and downstream sections [21], [22]. At the incident location, congestion will cause a decrease in traffic speed and volume, as illustrated in the red portion of the curve in Figure 2 [20], [21]. This can lead to an increase in the detector's occupancy in the upstream section [23], [24]. In contrast, the speed at the downstream section will improve since the number of vehicles passing through the incident location is decreased, leading to a decrease in traffic volume and detector's occupancy [25]. These effects will continue until the incident is cleared, and the queued vehicles start to discharge, allowing the volume and speed to return to their normal levels. Once the incident is cleared, normal flow will be restored, and traffic will return to its normal state.

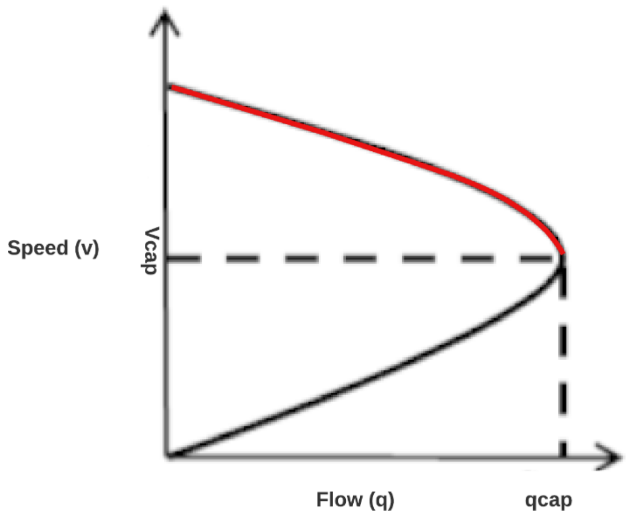

Figure 2 Flow-speed relationship: the Greenshields model [26], [27].

Figure 3, Figure 4 and Figure 5 show the changes in traffic flow rate, speed, and occupancy for both upstream and downstream stations during the simulation of an incident that blocked one lane of the road. Overall, these figures provide a comprehensive visualization of the impact of the incident on traffic flow.

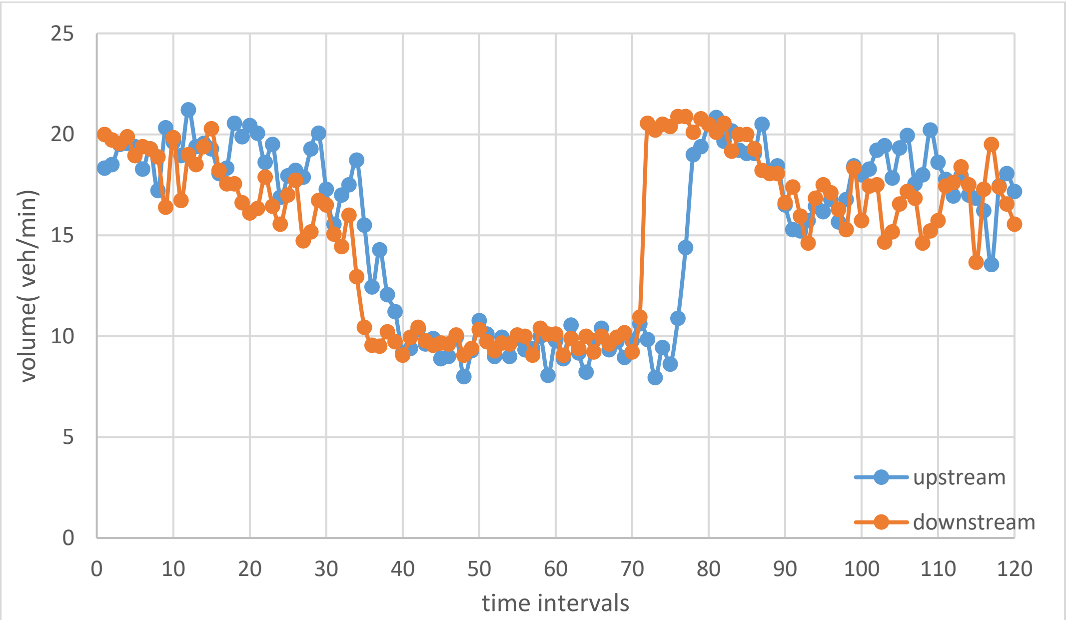

Figure 3 Changes in flow rate of upstream and downstream stations during traffic incident.

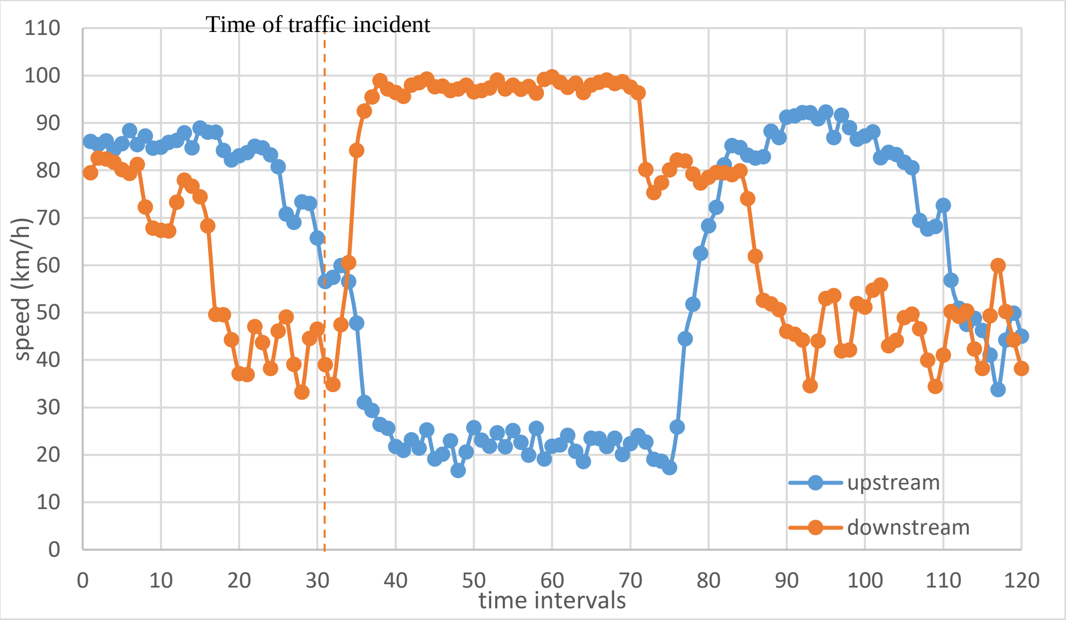

Figure 4 Changes in speed of upstream and downstream stations during traffic incident.

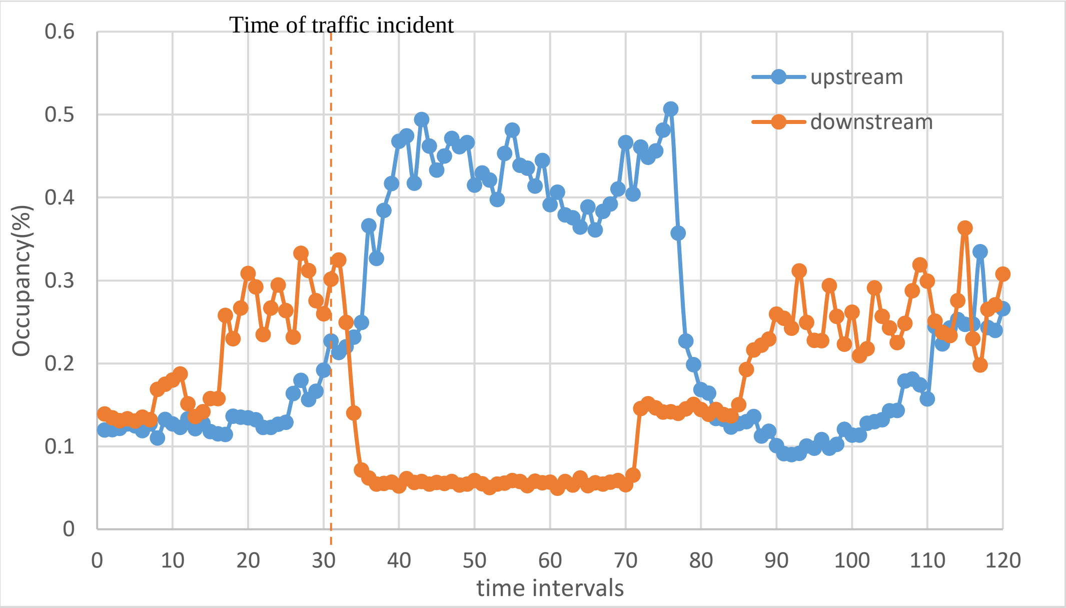

Figure 5 Changes in occupancy of upstream and downstream stations during traffic incident.

As the duration of the traffic incident increases the traffic flow deteriorates further and its impacts expand further [28]. The duration of incident can be defined as time between the occurrence of the incident and the clearance of the incident during which traffic is disrupted and returns to normal once the incident has been resolved [29]–[32]. Incident duration consists of four intervals: detection, response, clearance, and recovery [33], [34]. These intervals are depicted in Figure6.

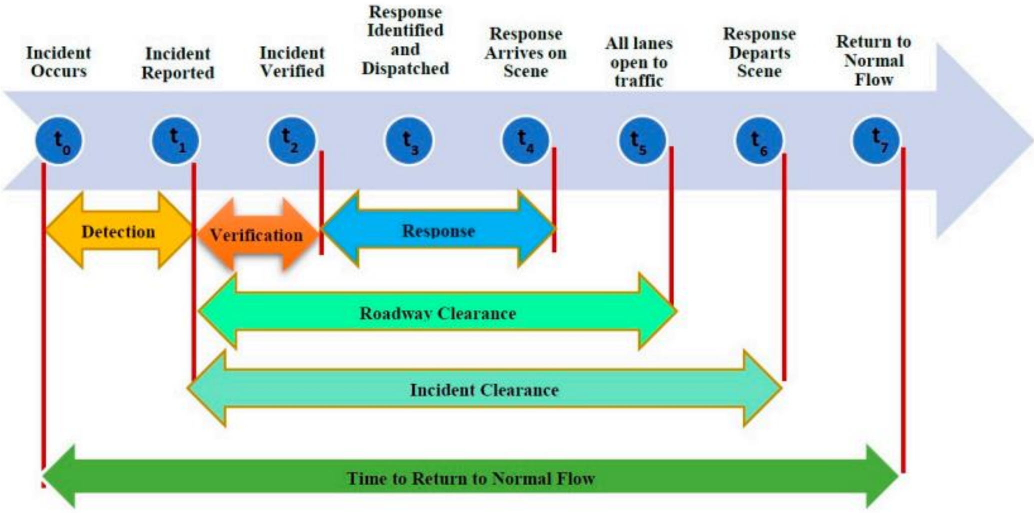

Figure 6 Timeline of incidents [35].

The actual incident occurrence time (T0) is difficult to detect and it depends on several factors such as the impact of this incident on the traffic parameters, the location of this incident and the ways of detection and reporting the incidents to TMC [36]. Usually, incident reporting time is used instead of the actual occurrence time to estimate the duration of the incidents. The timely estimate of incident duration plays a key role in incident management processes. Hence, the aim of an effective incident management system is to detect emerged traffic incidents as soon as possible to allow the TMC to take a quick response, provide necessary emergency services, alleviate traffic congestion and restore normal traffic flow [37]. AID algorithms can be classified according to the techniques used to collect traffic data either by using fixed sensors such as inductive loop detectors, radars, infrared sensors or surveillance cameras [38]–[40] or by using moving sensors such as GPS trackers, drivers' smartphones, probe vehicles, acceleration sensors and crash sensors [1], [18]. On the other hand, they can be categorized based on their data processing and detection algorithms as indicated in several references [41], [42]. However, there are some arguments in the literature about the number of these categories. Nevertheless, most of the previous research agrees that the incident detection algorithms can be classified into four categories: comparative, statistical, artificial intelligence based and video image processing algorithms [27].

2.1. Performance Measures

Incident detection is a binary classification problem where there are two classes, class 1 (no incident) and class 2 (incident). As mentioned before, incidents are non-recurring events therefore the first binary class is the majority class, and the second class is the minority class thus this is imbalanced binary classification problem. In the imbalanced classification problems, the majority class is referred to as the negative outcome while the minority class is referred as the positive outcome [43], [44]. The confusion matrix, shown in Table1, is used to provide insight into the performance of the classification model by showing which class was classified correctly and which was classified incorrectly.

Table 1 Confusion Matrix for Imbalanced Binary Classification Problems.

	Actual no incident	Actual incident
Predicted no incident	True Negative (TN)	False Negative (FN)
Predicted incident	False Positive (FP)	True Positive (TP)

Model's Accuracy is measured to determine how often correct predictions can be expected from the model out of the total number of predictions made [43], [44]. Accuracy is calculated using the following equation:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + True\ Negatives + False\ Negatives} \quad (1)$$

It is argued that accuracy is not a suitable measure in imbalanced datasets as it does not distinguish between the numbers of correctly classified examples of different classes, leading to erroneous conclusions. The impact of the minority class is reduced when compared to the majority class, and models that predict the majority class for all examples can achieve high accuracy but perform no better than an unskilled majority class classifier. Therefore, alternative metrics are commonly used to summarize model performance for imbalanced classification problems.

The Precision of the model is a metric that measures how many positive instances were classified correctly among the total number of positive instances classified by the model [43], [44]. The Precision of the model is calculated by the following equation.

$$Precison = \frac{True\ Positives}{True\ Positives+False\ Positives} \quad (2)$$

Although Precision is a useful performance metric, it doesn't indicate how many real positive instances were misclassified as false negative instances. Another metric that is often used to evaluate the performance of the model is the Recall , which quantifies the number of positive instances that were classified correctly among all positive instances in the dataset [43], [44]. Thus, the Recall is calculated by the following equation:

$$Recall = \frac{True\ Positives}{True\ Positives+False\ Negatives} \quad (3)$$

So, it provides an indication of misclassified positive instances. However, neither Precision nor Recall tells the whole story. A model can have excellent Precision with low Recall or vice versa. The problem of using these two metrics is that they conflict with each other. Increasing the Precision will decrease the number of false positives. In contrast, increasing the Recall will increase the number of false positives. So, it is challenging to obtain a model with high Precision and high Recall as increasing the Recall comes at the expense of reduction in Precision. Nonetheless, the F-score is a new metric that is widely used to capture the properties of both the Precision and Accuracy of the model [43], [44]. F-score represents the harmonic mean of the two metrics and can be calculated as follows:

$$F - score = \frac{2 * Precision * Recall}{Precision+Recall} \quad (4)$$

The F-score equally weighs the Precision and Recall thus it is the most common metric used for imbalanced binary classification model [43], [44].

The performance of an AID algorithm is usually evaluated by three measures; Detection Rate (DR), False Alarm Rate (FAR) and Mean Time to Detect (MTTD) [45]. The three measures can be defined as:

Detection Rate (DR): is a measure of the effectiveness of an alarm or incident detection system. It is defined as the proportion of true incidents that are correctly detected by the system. In other words, it is the ratio of the number of incidents detected by the system to the total number of incidents that actually occurred [46]–[48]. DR can be calculated using equation 5.

$$DR = \frac{total\ number\ of\ correct\ incident\ detections}{total\ number\ of\ incidents\ in\ data\ set} * 100 \quad (5)$$

False Alarm Rate (FAR): is a measure of the number of false alarms generated by the system. In other words, it is the proportion of alarms generated by the system that are not actually incidents. Two methods exist in the literature for calculating False Alarm Rate (FAR) [49]. The first method calculates FAR as the ratio of the number of false alarms generated by the algorithm to the total number of alarms generated by the algorithm [48], [50]. The equation for this method can be represented as:

$$FAR = \frac{\text{Number of False Alarms}}{\text{Total Number of Alarms}} * 100 \quad (6)$$

The second method calculates FAR as the ratio of the number of false alarms generated by the algorithm to the total number of times the algorithm was applied [49], [51]. The equation for this method can be represented as:

$$FAR = \frac{\text{Number of False Alarms}}{\text{total no.of applications of the algorithms}} * 100 \quad (7)$$

For instance, considering an example where the traffic state is monitored once every minute for an hour period. If the algorithm reports three false alarms during this hour, the FAR can be calculated, using Equation (7), as (3 false alarms) / (60 applications) = 5%. This percentage represents the proportion of times the algorithm reported a false incident during the hour-long monitoring period. In this dissertation, the second method will be used to calculate FAR.

Mean Time to Detect (MTTD): is a measure of the time it takes for an incident detection system to correctly detect an incident after it has occurred. It is defined as the average time elapsed between the occurrence of an incident and its detection by the system [50]. MTTD is a measure of the responsiveness and accuracy of the system in detecting incidents.

$$MTTD = \frac{\sum_{i=1}^{N_{det}} (t_d - t_{inc})}{N_{det}} \quad (8)$$

where:

N_{det} is the number of incidents detected,

t_d is the time at which the incident is detected, and t_{inc} is the time at which the incident occurred,

When relating the incident detection measures to the binary classification measure, the DR represents the Recall of the incident detection algorithm that was discussed previously, while the FAR can be calculated as:

$$FAR = \frac{False\ Positives}{True\ Positives + False\ Positives} \tag{9}$$

Or

$$FAR = 1 - precision \tag{10}$$

These three measures reflect the effectiveness, Reliability and the efficiency of AID algorithms, and they are not independent. When selecting an algorithm, a trade-off between these measures must be made. The higher the DR the higher the FAR and vice versa. an increase in DR can be associated with an increase in FAR because the incident detection system is designed to balance between detecting incidents accurately while minimizing false alarms [46]. As the system becomes more sensitive, it is more likely to detect true incidents, but it may also create false alarms. It is essential to find the optimal balance between DR and FAR when designing an incident detection system. While a low FAR is desirable to ensure the reliability of the system and reduce false warnings, it can be accomplished by lowering the DR of the algorithm, which can affect the effectiveness of the algorithm and contradict the ultimate goal of an AID algorithm [52], [53]. Therefore, selecting an algorithm requires balancing the need for accurate detection of incidents with the need to minimize false alarms. On the other hand, to shorten MTTD, it may increase the FAR. However, a persistence test can be used to reduce the false alarms, where an incident alarm triggers only after an incident pattern is observed and lasts for a number of consecutive intervals [54].

The following subsections provide an overview of different incident detection algorithms' categories (comparative, statistical, artificial intelligence based and video image processing algorithms), where the detection logic, the strength and the limitations of these algorithms are discussed.

2.2. Comparative (Pattern-based) Algorithms

These algorithms collect the traffic flow parameters from the field (traffic volume, traffic speed and occupancy). These traffic parameters are compared with preset thresholds that define normal traffic flow to assess the traffic conditions and identify unusual patterns. If they are outside the normal flow, an alarm of possible incident is set off [18], [46]. However, establishing and setting these thresholds are difficult and time consuming which is one of the main weaknesses of the algorithms in this category [46]. Comparative category includes several algorithms. However, the most common algorithms in this category are: California algorithm [16], which is the first and the most famous algorithm in this category, the Pattern Recognition (PATREG) algorithm [55] and All Purpose Incident Detection (APID) algorithm [56]. The details of these algorithms are discussed in the following subsections.

2.2.1. Comparative algorithms using occupancy.

The most widely known algorithm in this category is the California algorithm series developed by the Traffic Services Corporation (TSC). The California algorithm, also known as Traffic Services Corporation (TSC) algorithm 2, was among the first incident detection algorithms to be developed and is still widely used as a baseline for comparison. It is based on the principle that an incident is likely to increase the occupancy significantly in the upstream section while reducing occupancy in the downstream section. In TSC algorithm 2, the occupancies from two adjacent loop detectors are measured to test the occurrence of incidents in three steps as illustrated in Figure 7 [57].

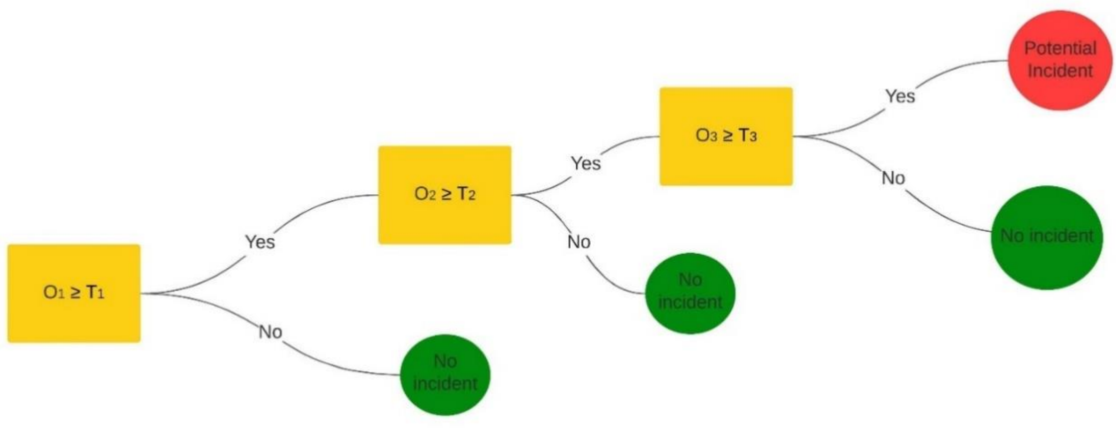

Figure 7 Schematic of TSC algorithm [57].

Where O_1 represents the difference in occupancy between the upstream and downstream detectors, O_2 is the proportional difference between the upstream and downstream occupancy levels, and O_3 is the change in occupancy at the downstream detector over a specific time. In addition, T_1, T_2, and T_3 are predefined thresholds [57]. A potential incident is detected if the values in three tests exceeds the preset thresholds [46]. These thresholds are calibrated from historical incident data and vary for each location where the detectors are installed. Also, they depend on road geometries, which require extensive calculation and computational time. Moreover, the performance of this algorithm is impacted by these thresholds. If the thresholds are too high, the algorithm may fail to detect certain types of incidents, while if the thresholds are too low, the algorithm may generate a large number of false alarms. These are the main drawbacks of this algorithm [58]. Additionally, this algorithm has some others flaws and limitations. This algorithm checks occupancies between two adjacent detector stations against predefined thresholds to determine the occurrence of an incident. However, if there is any malfunction in any detectors at any stations, the algorithm will not detect incidents between the upstream and downstream of the malfunctioning stations [59]. Moreover, some factors such as the grade changes, lane drops or ramps between detector stations can cause natural changes in occupancy and traffic operations similar to the pattern recognized by the algorithm as potential incidents [59]. Also, frequent bottlenecks during peak periods or failures of the detectors disrupt the continuity of the traffic operations and mimics the incident patterns and resulting in false alarms [20].

Payne and Tignor [15], [16] developed 10 new modified algorithms of the original California algorithm. Some of the best performing algorithms are TSC 7 and TSC 8 [16], [60]. TSC 7 replaced the test that uses relative temporal differences in downstream occupancy to detect incident in TSC2, with a simpler one that uses only downstream occupancy measurements [42]. By doing so, TSC 7 was able to avoid false alarms caused by compression waves which are common in heavy traffic. Thus, the TSC 7 algorithm is able to distinguish between true incidents and recurring patterns of congestion. In addition to test modification, a persistence check was incorporated into the basic algorithm, which mandates that a traffic disruption must persist for a certain period of time before an incident alarm is triggered. On the other hand, TSC algorithm 8 is the most complex and the best performing of the modified versions. According to Cohen and Ketselidou [60], TSC algorithm 8 utilizes a repetitive test for compression waves that can cause traffic slowdowns and momentary stoppages in heavy flows, which can result in false alarms. By analyzing traffic data, compressions can be detected, and alarms can be suppressed for up to five minutes at upstream locations, thereby reducing the occurrence of false alarms from normal traffic congestion. This algorithm categorizes traffic data into nine different states and it requires the calibration of five different threshold values. Its advanced features have made it the most reliable and efficient algorithm in detecting incidents in traffic systems [42], [60]. A study by Abdulhai and Abdelwahab [61] compared the performance of the three algorithms using data from a section of Highway 401 in Toronto, Canada. The study found that TSC7 had the highest DR of 92%, followed by TSC8 with a DR of 88%, and TSC2 with a DR of 84%. The FAR was highest for TSC2 at 12%, followed by TSC8 at 8%, and TSC7 at 5%. The MTTD was shortest for TSC7 at 2.5 minutes, followed by TSC8 at 3.5 minutes, and TSC2 at 4.5 minutes [61]. Another study by Stephanedes and Hourdakis (1996) compared the performance of TSC2 and TSC7 algorithms using data from a section of I-880 in Oakland, California [62]. The study found that TSC7 had a higher DR of 96% compared to TSC2 with a DR of 89%. The FAR was 18% for TSC2 and 6% for TSC7. The MTTD was also shorter for TSC7 at 1.1 minutes compared to TSC2 at 2.2 minutes [62]. Overall, these studies suggest that TSC7 and TSC8 algorithms have better performance in terms of DR, FAR, and MTTD compared to TSC2. TSC7 has been shown to have the highest DR and shortest MTTD among the three algorithms, while TSC8 has been shown to have the lowest FAR [63], [64]. However, it is important to note that the performance of these algorithms may vary depending on the specific road conditions and traffic patterns. In general, the California algorithm has a good DR and a tolerable FAR. However, it comes at the expense of delayed detection of incidents. As a result, the MTTD of this algorithm can reach up to four minutes, which is not acceptable in some cases [59].

All Purpose Incident Detection algorithm (APID) was developed by Masters et al. [65] to be used in COMPASS advanced traffic management system in Toronto. It is a combination of various California algorithms along with the compression wave test program and persistence for incident conditions test program to provide outstanding performance under all traffic conditions. That is where “all purpose” acronym came from. The APID uses five incident variables which are: the spatial difference in downstream and upstream occupancies, relative spatial difference in occupancies, relative temporal difference in downstream occupancy, downstream occupancy and relative temporal difference in speed for upstream. The first three variables are the same for the original California algorithm. The output of this algorithm can be one of three incident states; incident free, tentative incident and confirmed incident as shown in Figure8 [65]. In an offline evaluation, the APID algorithm demonstrated a high level of effectiveness in detecting incidents during periods of high traffic volume. However, its performance was observed to be suboptimal during periods of low traffic volume. The algorithm was able to detect 66% of the 29 incidents that caused traffic flow abnormalities (which corresponds to an 86% DR for incidents that can be detected). Additionally, it exhibited a MTTD of 2.55 minutes and a FAR of 0.05% per station, indicating its high accuracy in identifying incidents during high traffic conditions [65].

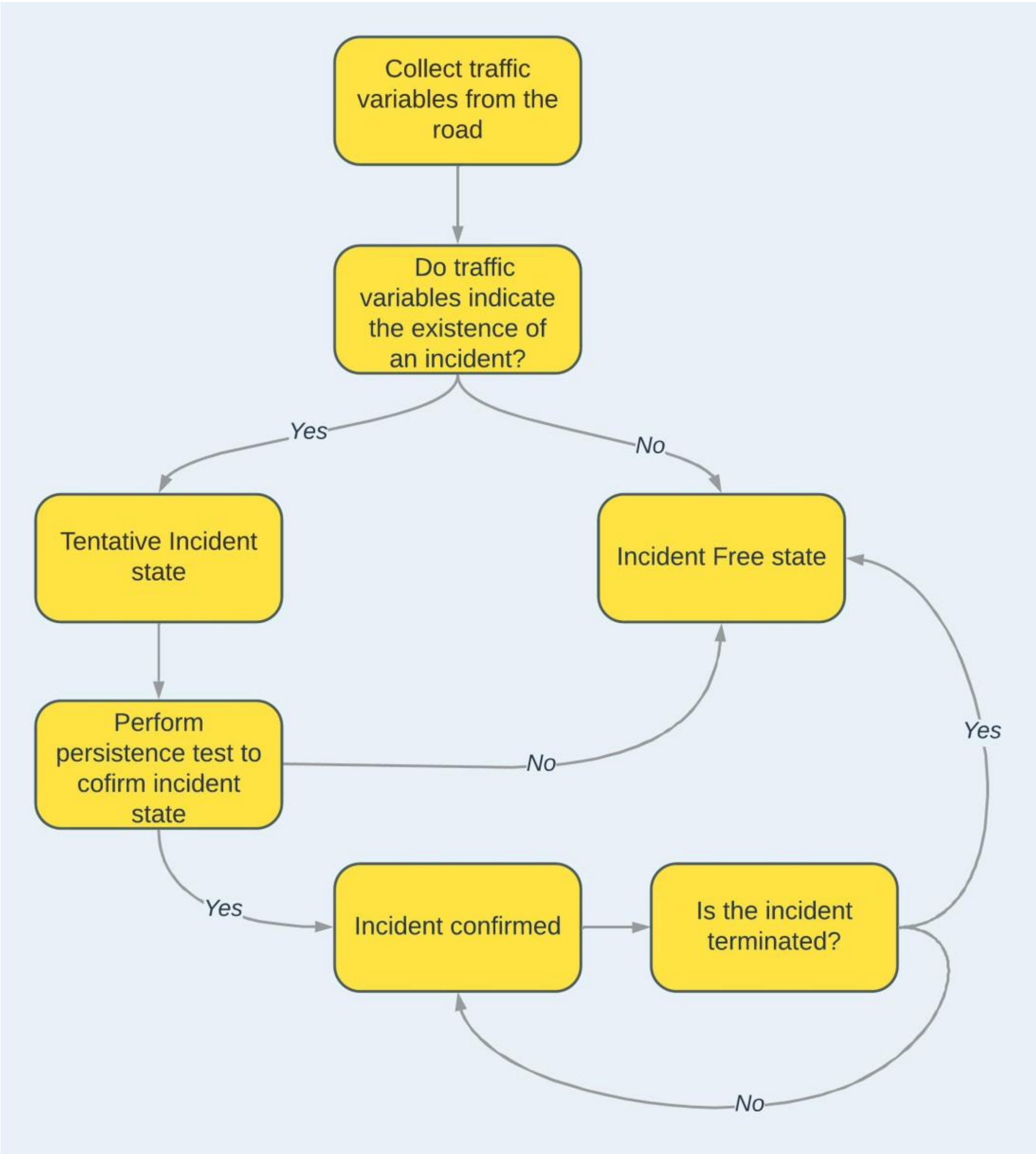

Figure 8 Incident states for APID algorithm [65].

To overcome the weaknesses of California algorithm series, a single station algorithm called McMaster algorithm was developed [66], [67]. Although in some researches, McMaster algorithm is classified as Catastrophe-based algorithm [46], this algorithm is discussed here in this category to compare it detection concept against California algorithm and other algorithms within the same category. The logic of this algorithm is based on the relationship between speed, flow and occupancy. It looks for any sudden change in one of the three variables while the remaining variables have smooth and continuous change [46]. This algorithm uses the data from a single detector station instead of two adjacent stations as in the California algorithm. Using historical data, flow-occupancy and speed-occupancy charts are created, and thresholds of incident conditions are set. If the traffic parameters exceed these thresholds for several consecutive periods, an alarm of potential incident is set off. This algorithm has some advantages over the California algorithm. First, because it uses speed, flow and occupancy instead of the occupancy only, the chances of detecting incidents increased [68]. Second, in this algorithm, an incident alarm is triggered at occupancies that are less than the critical occupancy thus incidents can be detected with larger probability and earlier [68]. Because of that, it has shorter MTTD, compared to California algorithm, that can reach to 1.5 minutes [59]. The third advantage is the number of false alarms caused by the natural variations of occupancy between adjacent stations due to change in grade or geometric characteristics, is reduced since this algorithm depends on a single detector station [59]. However, the algorithm was tested during snowstorm and this resulted in an increase in FAR [46], [59]. This was explained by the fact that snow covered the roads, and the drivers were more cautious and reduced their travelling speeds. Consequently, this reduction in the travelling speed was misclassified by the algorithm and false alarms were produced. Balke [69] performed a comparative analysis of different incident detection algorithms based on their DR, false alarm rate FAR, and mean time to detect MTTD as illustrated in Figure 9, Figure 10 and Figure 11. The analysis of the study indicates that the McMaster algorithm outperformed the other three algorithms in terms of both FAR and MTTD. On the other hand, the APID algorithm achieved the highest DR compared to the other three algorithms. These findings suggest that different incident detection algorithms may have varying strengths and weaknesses depending on the specific performance metrics being considered.

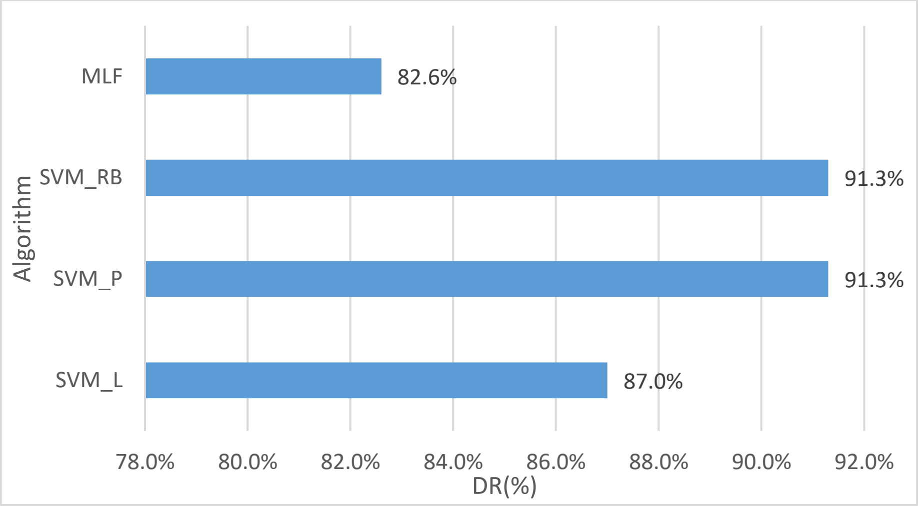

Figure 9 Comparison of the DR of different incident detection algorithms [69].

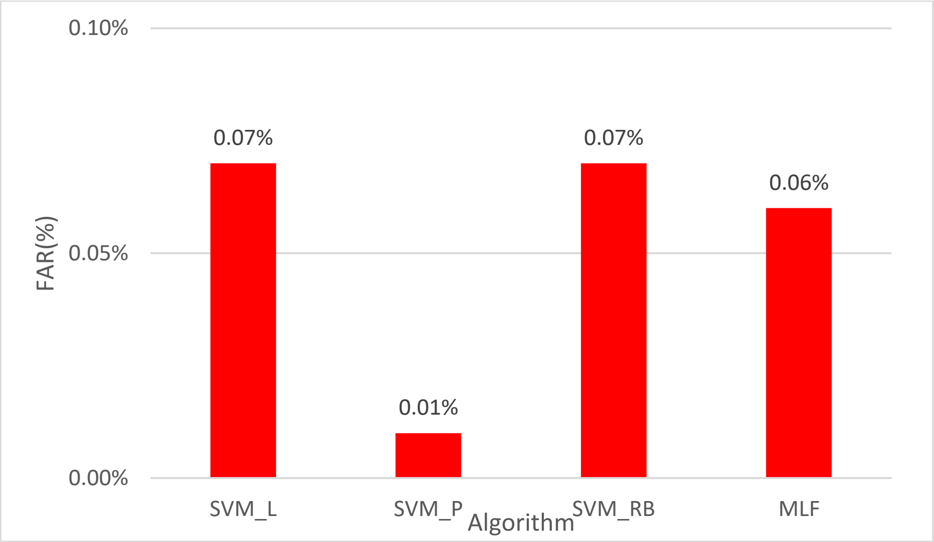

Figure 10 Comparison of FAR of different incident detection algorithms [69].

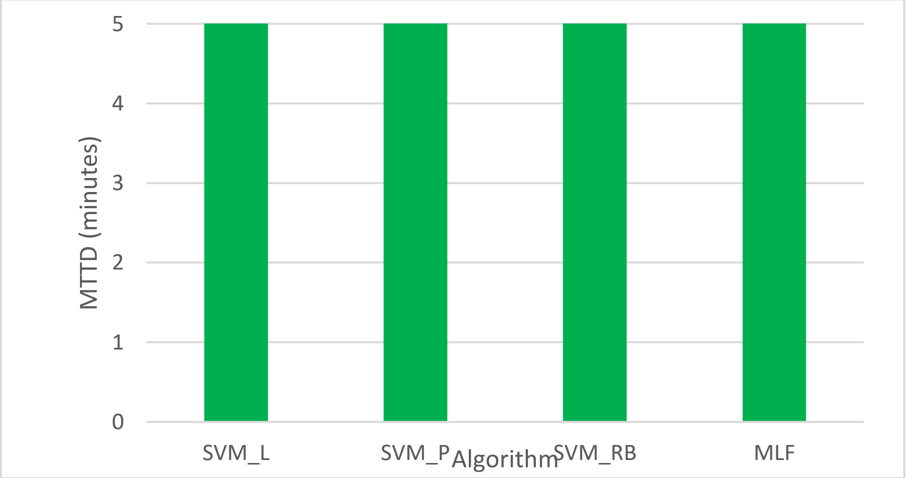

Figure 11 Comparison of MTTD of different incident detection algorithms [69].

A team of researchers from the University of Minnesota developed an incident detection algorithm, which was named Minnesota algorithm. This algorithm evaluates the spatial occupancy difference between adjacent detector stations to detect the occurrence of the incident when this difference changes drastically over a short time period [70]. First, the average occupancies at the upstream, o_t^u , and downstream stations, o_t^d , are collected over 30 second intervals to compute the spatial occupancy difference (o_t^u-o_t^d). The average of the spatial occupancy differences is calculated over five intervals, from the interval of the incident occurrence and 4 intervals after the occurrence of the incident. If the incident occurs at time t-5, then the average spatial difference is calculated as follows.

$$y_t^a = \frac{1}{6}\sum_{k=0}^{5}(o_{t-k}^u - o_{t-k}^d) \quad (11)$$

Then, the average spatial difference is calculated over 10 intervals before the occurrence of the incident from t-15 to t-6 as follows.

$$y_t^b = \frac{1}{10}\sum_{k=6}^{15}(o_{t-k}^u - o_{t-k}^d) \quad (12)$$

The premise of this algorithm is that when an incident occurs at time t-5, y_t^a will increase and it will be greater than y_t^b. If this difference is greater than certain thresholds, which are calibrated with historical data, an incident alarm is declared. The aim of taking the average occupancy measurements over these short time intervals is to smooth up the high- frequency random fluctuations in the traffic data. These random fluctuations can form incident-like patterns (as discussed earlier in the California algorithm) and trigger false alarms. In their study, the University of Minnesota researchers compared their newly developed Minnesota algorithm to four existing algorithms (California algorithm, TSC 7, Standard Deviation, and Double Exponential algorithms) on the same data set. The researchers found that the Minnesota algorithm outperformed all existing algorithms at all levels of DR. At a 50% DR, the Minnesota algorithm produced a 0.1% FAR, while the best-performing existing algorithm produced more than twice the number of false alarms. Similarly, at an 80% DR, the Minnesota algorithm achieved a FAR of approximately 0.3%, which was significantly better than TSC 7’s 0.6% FAR. According to the study, the Minnesota algorithm produced between 40-70% fewer false alarms than the California algorithm, 30-60% fewer than TSC 7, and 70-80% fewer than the Standard Deviation and Double

Exponential algorithms, at various detection rates. Additionally, the study found that the Minnesota algorithm had a relatively low MTTD, which remained very close to the detection times required by the comparative algorithms. The Minnesota algorithm exhibited a competitive response time and could detect an incident faster than the operator. The researchers also noted that the time-to-detect was low at high DRs [71]. However, using the occupancy only as the input traffic feature may cause false alarms during low traffic conditions as the flow of the vehicles will be low and thus the occupancy values. Also, the detection time of this algorithm is three minutes or higher since it uses the occupancy values from five intervals following the occurrence of the incident [71].

2.2.2. Comparative algorithms based on travel time and speed.

Margreiter et al. [72] developed an approach that uses Bluetooth detectors to measure the travel time of the vehicles. The aim of using these Bluetooth detectors is to develop reliable, cost efficient and fast methods for detecting traffic incidents or congestion. The Technical University of Munich (TUM) implemented this method on Bavarian freeway, Germany as part of a project called iRoute. The derived traffic data are used to provide re-routing advice in case of accidents, congestion or capacity overload. Based on data from the detectors, significant variations in travel times or speed could be an indication of incident occurrence. Using Bluetooth detectors instead of inductive loop sensors or traffic radars is more cost efficient as they have lower power, communication, installation and maintenance costs. The Bluetooth detectors were installed near inductive loop detectors to compare the data from inductive loops with the data from the Bluetooth detectors for calibration and validation. Karatsoli et al. [73] studied the optimal detectors distance that achieve the highest performance possible of TUM algorithm in terms of MTTD, DR and FAR. They analyzed real-time traffic data of a 15 km road section of A9 motorway. The data were obtained from three Bluetooth detectors installed on this section for a period of about 45 days. Any incident that occurred at that period was recorded using a Closed-Circuit Television system (CCTV) and compared with the result of the TUM algorithm. In addition to the field data, VISSIM model of the study area was developed considering different detectors layout or locations, different traffic conditions and different incident types to produce a wide range of results. A total of 27 scenarios were simulated with 9 incident scenarios and three traffic volumes (low, medium and high). The travel times resulting from the simulation were processed and used in the TUM algorithm. The results revealed the following:

- The severity of the incidents affects the DR and MTTD. Severe incidents that cause dramatic change in capacity and traffic flow can be detected easier and quicker thus resulting in high DR and low MTTD. However, minor incidents that last less than 30 minutes could not be detected or take long time to be detected.
- Under different traffic volumes, as the distance between detectors increases the DR decreases and vice versa. Since the incident impacts can be smoothed quickly before they reach the downstream detector.
- The location at which the incidents happen impacts the MTTD. If the incident is located closer to the downstream detectors it can be detected quickly and thus has lower MTTD.
- The optimal detectors' distance was found to be between 3 and 4 km based on the DR, FAR and MTTD.

2.2.3. Comparative algorithms based using vehicles' spatial-temporal data.

Khalil et al. [74] proposed a technique to detect incidents on the road by using ultrasonic sensors. The proposed technique for incident detection involves the use of two ultrasonic sensors placed on the front and back side of the car's roof. The sensors measure the distance between the car and its surroundings by sending and receiving reflected sound waves. The threshold distances are predefined for the first and second systems based on the distance between the sensors and the bumpers of the car. When a vehicle or any obstacle collides with the car within the pre-set threshold distance, the respective system turns on and determines the location of the car using a GPS module. The system then sends information to the emergency department through a GSM module. This approach allows for the detection of accidents and prompt communication of relevant information to rescue teams [74]. However, the range and the placement of the sensors have significant impact on the efficiency of the sensors or may cause false alarms. These are the main limitations of this technique. Furthermore, an expert system was developed to collect real-time GPS data from driver's mobile phones or GPS trackers in the vehicles in order detect traffic incidents or congestion. The system consists of three modules. The first one is the pre-processing module that matches GPS traces on digital map and determines travel directions of the vehicles. The second module is the segment traffic classification that assigns traffic state to each segment of the road network based on the average speed and flow of the vehicles that are passing through the segment and the speed limit of that segment. The third module is the traffic alert notification that performs spatiotemporal analysis of the information of each segment to determine if there is an incident or congestion to send traffic alert messages to users indicating the affected area, traffic state and speed of the vehicles on that area. To validate the ability of the system to detect traffic congestion and incidents simulated GPS traces were developed using SUMO (a traffic simulator software) [75]. The performance evaluation of the system showed that it has a DR of 91.6%, FAR of 8.3% and the MTTD traffic incident is about 7 minutes [76]. Similarly, a multilevel traffic incident detection approach was developed to collect real-time GPS data from vehicles to detect abnormal traffic conditions and incidents [1]. The approach consists of two phases, in Phase 1 the road is divided into smaller segments and then a normal average speed range for each segment is assigned. The current average speed of the vehicles on each segment is compared with the assigned normal average speed. Based on these comparisons on multiple segments it is determined if the traffic on a specific segment is slowed due to general congestion or because of a traffic incident. In Phase 2 the GPS data of the individual vehicles is analyzed to identify any abnormal vehicle behavior. Based on that the location of this vehicle is determined and sent to the emergency services. Probe vehicles equipped with a GPS receiver and data transmission devices were used to send real time GPS data in addition to simulated GPS data. The DR and FAR varies based on the number of vehicles and varies from Phase 1 to Phase 2. The DR fluctuated from about 52% to around 79%, depending on the number of vehicles. The DR decreases as the number of vehicles increases or the procedure moves from Phase 1 to Phase 2 because the complexity of the data increases. On the other hand, the MTTD ranged from about 2 seconds to more than 3 minutes based on different road types and conditions [1]. Han et al. [77] availed GPS trajectories of taxis in Hong Kong to detect traffic incidents on Urban roads. GPS trajectories use a finite number of GPS points that consists of timestamp, location and the speed of the vehicles. To detect incidents in the roads, GPS points were aggregated within a single road segment. Then, construct speed vector at a sequence of road segments travelled by the vehicles. The speed vectors are compared with the historical average speed of the same road segments that represent normal speed vectors. A significant difference between the two vectors indicates the occurrence of incidents. Two datasets were used to evaluate the performance in terms of DR, FAR and MTTD. The evaluation results of the first dataset were about 76%, 15% and 2.3 minutes for the DR, FAR and MTTD respectively. While the second dataset has DR about 86%, FAR around 9% and MTTD 1.1 minute.

The fixed detectors have some limitations and weaknesses. For example, the frequent malfunctions and breakages of these detectors can affect the performance of incident detection algorithms [78]. Besides, these detectors may not perform well under different environmental conditions such as fog, rain or snow [79]. In addition, the cost and maintaining loop detectors are high thus they are sparsely placed and this may impact their effectiveness in detecting changes in traffic conditions [80]. Alternatively, Floating Car Data (FCD) has been proposed as an alternative to fixed detectors for incident detection due to the limitations and weaknesses of the latter. Moreover, FCD can also be utilized for traffic estimation and prediction, which can aid in better traffic management. The use of FCD for incident detection and response can also benefit from the availability of real-time and fine-grained data on vehicle movements, which can be used to detect abnormal traffic patterns and incidents as they occur [81], [82]. Further, Houbraken et al. [81] evaluated the feasibility of using FCD method for AID. In this case, the vehicles act as moving sensors and provide connected measurement locations and the generated traffic data is spatially distributed. A real-time FCD was used as input for the AID system. The study was conducted on two Dutch highways which are equipped with AID systems based on installed inductive loop detectors. The installed AID system detects the vehicles that pass over the detectors to calculate their speeds and keeps updating the average speed of the road. Likewise, FCD provides the data of the tracked vehicles to calculate their speeds and the calculated speed triggers AID ON/OFF messages. The performance results of FCD AID were benchmarked against the results of loop-based AID in terms of false positive (FP) percentages, false negative (FN) percentages and hard misses. The hard misses denote the proportion of time the loop-based AID is ON while FCD AID is OFF and remains OFF for more than 60s. The results showed that FCD has false negative percentages of 4-5% (of loop based AID benchmark) thus FCD can sufficiently monitor traffic conditions and can be used in combination with loop detectors to improve the performance of the AID and increase the detection coverage with low cost [81]. In the same way, Liang et al. [82] implemented FCD to detect traffic congestion incident in urban intersections in Ningbo city, China considering the impact of urban traffic signal control on urban traffic congestion incidents using real time FCD and traffic signal cycle data. The proposed method is evaluated in terms of DR and FAR. The DR was about 85.5% while the FAR was around 11%. In addition, the results were compared with the real-time CCTV monitoring system of the city to validate the obtained results. The results showed that the proposed method can accurately detect traffic congestion at urban intersections [82]. Further, Asakura el al. [83] exploited FCD method to develop two incident detection algorithms. In the first algorithm, the corridor is divided into sub-sections (links) with the same lengths. Travel time differences of probe vehicles and the flow rate differences (that is calculated from the number of vehicles passing the probe vehicles) between adjacent links are used to detect bottlenecks caused by traffic incidents. If the travel time differences and flow rate differences between adjacent links exceed predefined thresholds, then an incident is detected. However, this method can't be used when the polling frequency is greater than 30 s because the travel time of the links can't be measured accurately, and the travel time differences will not be reliable. The second algorithm uses shockwave theory and requires three consecutive probe vehicles. The first vehicle i-1 should be the vehicle immediately before the start of the incident, so it doesn't experience the incident while the following two vehicles (i and i+1) experience the incident and pass the bottleneck caused by the incident. Traffic simulation was used to examine the performance of these two algorithms using three measures: DR, FAR, and MTTD at different penetration rates of probe vehicles. At 1% penetration rate of probe vehicles, algorithm 1 achieved DR 50.2%, FAR 0.015% and MTTD 16.5 min. While the results showed that the second algorithm required a higher penetration rate than the first algorithm (at least 5%). Because at low penetration rate the headway of probe vehicles is too large to detect. At 5% penetration rate, algorithm 2 achieved DR 54.8%, FAR $9.5 * 10^{-4}$ % and MTTD 3.5 min.

In general, the FCD is a cost-effective method as the hardware cost and servers' cost are low. However, FCD method has some limitations and flaws, mainly the penetration rate of the tracked vehicles in the road and data latency. Fogue et al. [84] proposed a prototype of a system called *e-NOTIFY* that uses the vehicle to vehicle (V2V) communication and the vehicles to infrastructure (V2I) communication. The purpose of the system is to detect traffic incidents and report them to an external Control Unit (CU) to estimate the severity of the incidents, allocate the required rescue resources to the incident location and reduce the response time. The vehicles should be equipped with sensors and an On-Board Unit (OBU). The OBU collects data from the in-vehicles sensors (such as air-bag triggers, horizontal tilt sensor and accelerometer) to determine if an accident happened and reports that to the nearest external Road-Side Unit (RSU) and the adjacent vehicles that may be affected. The CU receives this information from RSU and pulses of vehicles' acceleration and if they exceed certain thresholds an incident alarm is activated. It takes into consideration the amplitude and duration of the pulses to determine the severity of the incidents and report them to the emergency services to take the necessary actions [84]. This prototype was validated at the Passive Security Department of Applus+ IDIADA Corporation and the results showed that it can detect incidents effectively and report to the CU [84]. Likewise, the European Commission developed an emergency call system called *eCALL*, which aims to improve transportation safety by using in-vehicle sensors to detect incidents automatically through on board sensors in the vehicles [85], [86]. Based on the same concept, various projects such as *COMeSafety2* [87], *OnStar* [88] and *Campus Vehicular Test Bed* [89] that focus on inter-vehicle communications for incident detection were developed by different research organizations and car manufacturers. The proposed system by Patil and Pardeshi [90] complements and enhances these existing incident detection and reporting systems. In addition to the GPS, accelerometer, and camera sensors are used in other systems . The proposed mechanism includes an alcohol sensor, which can help detect drunk driving and prevent accidents caused by impaired driving. This added feature is a significant improvement in the system's ability to prevent accidents, reduce response time in emergencies, and enhance overall road safety. Therefore, the proposed system by Patil and Pardeshi is a valuable addition to the existing incident detection and reporting systems, which can benefit from the system's additional features and improved performance. It should be noted that the availability of communication protocols among different entities, including vehicles and infrastructure, greatly impacts the effectiveness of these incident detection and reporting systems.

Overall, the use of V2I, V2V, and other sensor-based technologies to develop incident detection systems is proving to be a valuable asset in detecting incidents, reducing response times, and enhancing overall road safety and potentially saving lives. However, Further research and development in this area are necessary to ensure that these technologies continue to evolve and remain effective in detecting incidents and enhancing road safety.

2.3. Statistical Algorithms

This category of algorithms uses the temporal characteristics of traffic flow to build models to predict traffic data using statistical methods. Then the collected/ observed traffic data are processed using various statistical analyses to determine the difference between actual or observed data obtained from the road and the predicted or estimated traffic parameters from the statistical model. If the difference is statistically significant, an incident condition is identified [91]. The most common algorithms in this category include the Standard Normal Deviate (SND) algorithm [13], smoothing algorithms (such as double exponential smoothing algorithm) [12] and Bayesian algorithm [17], [92]. In addition, it includes time series algorithms, which assume that the normal traffic follows a predictable pattern over time.

2.3.1. SND algorithm

The SND algorithm was developed in 1974 by Texas Transportation Institute (TTI) to use it on the Houston Gulf Freeway (I-45) [13]. This algorithm calculates the SND of the traffic data to detect anomalies in the traffic data. The SND is calculated using the following equation:

$$SND = \frac{|x_k - x_0|}{\sigma} \tag{13}$$

Where x_k is the observed traffic variable, x_0 is the mean of the traffic data taken over a period of time and σ is the standard deviation of the traffic variable. The calculated SND is compared to a certain threshold. The incident detection of this algorithm is based on the premise that high deviation or change in the traffic variable can indicate the occurrence of traffic incident [46]. To reduce false alarms, consistency test is performed to check if the SND is higher than a specific threshold, for two successive intervals, then an alarm of incident is triggered [18]. However, determining these thresholds is tedious and time consuming. The results of the offsite evaluation of this algorithm showed a DR of 92%, FAR 1.3% and MTTD 1.1 minutes [18]. Although it has high DR and FAR, the main flaw of this method is the presence of outliers. These outliers can inflate the standard deviation and mean values which will lower the threshold values hence decreases the DR, this phenomenon is called masking [93].

Likewise, Ahmed and Hawas [94] developed an incident detection system for urban traffic networks using comparative statistical analyses of traffic parameters. A simulated model of a signalized intersection was developed using NetSim network simulator [95]. Incident scenarios were generated in the NetSim simulation runs. Detectors' data for both incident and non-incident situations were extracted. The considered traffic parameters were the accumulated vehicle counts and the average speeds. The deviation from these two parameters were used to develop a linear regression model to predict incident status. The developed model achieved DR values that ranged from 23% to 87% and FAR ranged from 0% to 20%. The developed model was validated using another set of simulated incident scenarios different than those used to develop the model. However, the limitation of this model is that it has low DR and high FAR at low traffic volumes because incidents didn't have significant impact on the traffic parameters and the detectors readings [94].

2.3.2. Inter-Quartile Distance (IQD) algorithm

To address the SND algorithm's masking issue, Chakraborty et al. [96], [97] developed an Inter-Quartile Distance (IQD) approach for detecting anomalies in traffic speed data that may indicate incidents or congestion. In this approach, the median value, which is the second quartile of the traffic data stream is used instead of the mean and the IQD score is used instead of SND. The IQD is the difference between the third quartile, which represents the 75th percentile, and the first quartile, which represents the 25th percentile, divided by 1.35 [96]. The outlier detection process involves selecting a reference value, which is the sample median, and a measure of variation, which is the IQD. A data point is considered an outlier if the absolute difference between this point and the reference value exceeds the product of the measure of variation (IQD score) and a threshold parameter obtained from historical data. This method doesn't suffer from masking problem, but it suffers from another phenomenon called swamping. In swamping the IQD value becomes zero if more than 50% of the data values have the same value. This causes any value different from the median to be declared as an outlier, regardless of its distance from the median. Hence the FAR will increase in the incident detection algorithm. However, according to Federal Highway Administration (FHWA) guidelines, congestion conditions occur in the freeways when the speed is less than 45 mph [98]. Thus, incident alarms should be triggered only if the speed is less than 45 mph. This can eliminate false alarms during swamping cases. The applicability of this method was tested on speed data from the Des Moines region in Iowa, covering 254 segments and 164 miles over a period of three months. The study computed the median and IQD values for each 15-minute period of each day of the week, based on the historical speed data of each segment. The threshold parameter was determined through Cross-Validation, using speed data observed during incidents. However, determining the threshold parameter from Cross-Validation data requires incident data with accurate start and end times, which is challenging to obtain for new sites where the methodology is applied. Additionally, not all segments experience incidents, making it impossible to determine threshold parameters for all sites. Therefore, the authors noted that the threshold values used were based on data from other sites and may not be suitable for all sites. The method achieved a DR of approximately 97%, a FAR of about 4.8%, and an MTTD of 12.4 minutes [96]. The study had a limitation in that it adopted a persistence test of three minutes before triggering an incident alarm. This contributed to the long MTTD value that was obtained.

2.3.3. Double Exponential Smoothing (DES) algorithm

Double exponential smoothing (DES) is one of the smoothing techniques that are used to remove the noise and heterogeneity form the traffic data to clarify the true traffic patterns to help the system to detect incidents easily and reduce false alarms [18]. The DES was developed by Cook [99]. It uses past and present observations of traffic variables such as volume, occupancy and speed to provide short term forecasting of traffic conditions. It uses exponential functions to assign exponential decreasing weights of the past observations over time and the recent observations are assigned higher weights. Incidents are detected using a tracking signal, which is the sum (to the present minute) of all previous estimate errors between observed and predicted traffic variables divided by the standard deviation. The standard deviation of traffic data is estimated from the mean absolute deviation m (x,t) which is computed from single exponential smoothing of the absolute values of the predication errors [12], [46]. In the normal condition, if there is no incident, the tracking signal should dwell around zero because the predicted and observed traffic variables should be similar or the predications compensate for errors in succeeding time periods. Similar to the SND algorithm, large deviation of tracking signal from the zero, exceeding a threshold deviation, indicates the occurrence of an incident. DES were evaluated over 13 different traffic variables and the results showed that volume and occupancy yield the best performance of the algorithm [12].

2.3.4. Time series algorithms

Time series algorithms have been considered by some researchers as a separate category [18], [47] while other researchers consider them as a subset of statistical algorithms [46], [91]. However, it makes more sense to be included as part of statistical algorithms to reduce the confusion for the readers since they are using statistical techniques for incident detection. Time series algorithms assume that the normal traffic follows a predictable pattern over time. They are used to predict these normal traffic conditions and if the observed data is significantly different than the normal conditions, incidents are detected [47]. This category includes the Autoregressive Integrated Moving-Average (ARIMA) model [100]–[102] and High Occupancy (HIOCC) algorithm [55].

Li et al. [103] proposed comparative AID algorithm for urban expressways in China. Urban expressways extend over 5% of the total road area of the central area of Shanghai and serve about 20% of the total traffic flow. Therefore, traffic accidents on these roads cause sever traffic congestion and loss of time in addition to significant increase in transportation cost. Most of the existing AID algorithms are developed based on the transverse time series of traffic parameters. However, the main problems in transverse time series are the high fluctuations of the traffic volume which results in higher FAR and lower robustness. The proposed AID algorithm used the lengthways time series of traffic parameters as the historical data that represents the basis. The lengthways time series of traffic parameters provides good stability as the data are collected over a week interval at the same time and at the same site. Therefore, the lengthways time series can overcome the disadvantages of the transverse time series. The lengthways time series was utilized to estimate the traffic parameters in normal traffic status for each moment by using the moving range method. When a traffic incident occurs, the traffic flow and the speed decrease while the occupancy ratio increases upstream of the incident location. The researchers developed the concept of traffic status variability Index (I) to compare the actual traffic status at a certain moment with the normal traffic status at the same moment. When the variability index exceeds certain threshold values this indicates abnormal traffic status. Based on that it can be concluded if there is a traffic incident or not [103].

2.4. Artificial Intelligence Based Algorithms

Incident detection can be viewed as a pattern classification problem, where the traffic patterns of incidents and non-incident are to be recognized and classified [104]. The classification is based on the changes of the spatial and temporal traffic patterns during incidents since the traffic maintains consistent pattern at upstream and downstream segments during normal conditions. Accordingly, these algorithms exploit Artificial Intelligence (AI) to develop efficient AID systems. Machine Learning (ML) is a subset of AI which consists of algorithms that model the optimal mapping between set of inputs and set of outputs [105]. The ML models algorithms include Artificial Neural Networks (ANN), Fuzzy Logic (FL), Support Vector Machine (SVM) or a combination multiple models [18], [46]. The ML models have many applications in the transportation sector. They have been used for traffic planning and management [106]–[111] predicting the duration of traffic incidents [112] and for Incident detection [113]–[116]. The advantage of using ML models is that they learn from the input data and deduce the pattern without relying on a predefined mathematical model. Such models can learn incident and non-incident patterns and then recognize these patterns and classify traffic data as incident or non-incident.

2.4.1. Artificial Neural Networks (ANN) based algorithms.

ANN is composed of interconnected artificial neurons that are used for pattern recognition in a set of data and to solve other AI problems [117]. This technique is inspired by the human's brain neural network [118], [119]. The ANN was proposed by Warren McCulloch, and Walter Pitts in 1943 [116, 117], to mimic the way that biological neurons in the human's brain process the information. It has several applications such as pattern recognition, clustering, classification and function fitting. With the development of computers and machine learning, many algorithms and methods have been proposed for AID using ANNs.

Multilayer Feedforward (MLF) seems to be one of the first ANN use for incident detection [122] and it has been widely used due to its high DR and FAR [54], [123]–[125]. Cheu and Ritchie [54] availed ANN models to develop a method to detect lane-blocking incidents. First, three types of ANN models were investigated, which are

MLF, Self-Organizing Feature Map (SOFM) and Adaptive Resonance Theory 2 (ART2). These models were developed with simulation data from INTRAS, a microscopic freeway traffic simulation model, which included some lane-blocking incidents with different severity, various traffic flow conditions and at different locations. A separate set of INTRAS simulation data was assembled to assess the relative performance of the MLF, SOFM, and ART2 models. An analysis of incident detection performance was conducted on this data set, which contained incidents of different severity in terms of the number of lanes blocked under a range of peak and off-peak traffic volumes. The incident detection algorithm declared an incident alarm if the traffic state changed from state 1 (no incident) to state 2 (incident condition) and persisted for a specified time interval. Persistence tests were incorporated to minimize false alarms, with up to three intervals used. DR, FAR and MTTD were used to evaluate the performance of the incident detection algorithm. The DR values for the MLF model ranged from 80% to 51% for the 0 to 3 interval persistence tests, with decreasing DR values as the persistence intervals increased. The DR values for the SOFM model ranged from 59% to 39%, while those of ART2 ranged from 56% to 35% [54]. Regarding the FAR, the MLF model yielded FAR values ranging from 1.5% to 0.17% for the 0 to 3 interval persistence tests, with decreasing FAR values as the persistence intervals increased. The FAR values for the SOFM model ranged from 0.4% to 0.14%, while those of ART2 ranged from 0.96% to 0.19%. For MTTD, the MLF model yielded MTTD values ranging from 194 seconds to 297 seconds for the 0 to 3 interval persistence tests, with increasing MTTD values as the persistence intervals increased. The MTTD values for the SOFM model ranged from 252 seconds to 352 seconds, while those of ART2 ranged from 282 seconds to 387 seconds. Following the determination that the MLF model was the best ANN, it was further compared against the TSC 8, McMaster, and Minnesota filtering algorithms. These algorithms operate at 30 second intervals and take in input from loop detectors such as occupancy and/or volume. The aforementioned algorithms were tested using the same dataset, with the TSC 8 achieving a DR range of 59% to 49%, the McMaster algorithm achieving a DR range of 41% to 21%, and the Minnesota algorithm achieving a DR range of 78% to 37%. The FAR of the TSC 8 ranged from 0.4% to 0.22%, the McMaster algorithm ranged from 2.5% to 0.2%, while the Minnesota algorithm ranged from 2.92% to 0.2%. MTTD values for the TSC 8 algorithm ranged from 252 seconds to 288 seconds, while the

McMaster algorithm ranged from 258 to 344 seconds, and the Minnesota algorithm ranged from 223 to 316 seconds. The results demonstrated that the MLF model exhibited the most superior performance in detecting lane-blocking incidents amongst the aforementioned algorithms. This was evident in the model's ability to achieve the highest DR, while simultaneously having the lowest FAR and MTTD [54].

Ki el al. [20] developed an ANN model to recognize certain traffic pattern in order to detect traffic incidents. Traffic incidents create congestion upstream and reduce the traffic flow downstream, thus leading to speed difference between the upstream and downstream traffic flow. The proposed ANN model uses this feature to detect traffic incidents. The ANN model used real world data of the traffic conditions and actual incidents that were recorded by a traffic management center in Seoul and Uiwang cities in South Korea. The model showed a DR of 77.3%, which is higher than existing algorithms in the literature such as California model. However, the FAR was relatively high as in some cases traffic may exhibit incident-like patterns when there is no incident. However, in this case they didn't mention what FAR is for their model which is necessary to assess the reliability of the model.

In another case, the normal conditions were set as the baseline traffic parameters and minor incidents (with durations less than or equal to 30 minutes) were generated at different locations and with different traffic intensities. The purpose of this simulation model is to develop an AID algorithm that can effectively detect minor incidents at low traffic volumes. The experimental design considered two traffic volumes and three different locations for the incident. The model was trained using data obtained from the regulated network section. The input parameters used to train the model were the were the deviations from the normal conditions of the cars' quantity, speed, and occupancy over one-minute time intervals. The database was divided into three datasets, including a training dataset, a testing dataset, and a control dataset. The developed MLF architecture was a 9-4-2, consisting of 9 input parameters, 4 hidden neurons, and 2 output neurons. The results indicated that the developed model had a DR of 97.6% for the training dataset, 94.4% for the testing dataset, and 88.9% for the validation dataset. It is worth noting that the study did not evaluate the performance of the model using FAR and MTTD metrics, which are crucial factors in evaluating the efficiency of incident detection models [126].

One of the machine learning methods, Long-and Short-Term Memory Network (LSTM), was utilized to develop an AID framework. Unlike other approaches that have been developed based on image data sets, the proposed LSTM uses videos captured by CCTV surveillance cameras that are installed in the road network. The procedure extracted frames from these videos and used two different LSTMs to extract spatial and temporal features from the frames and detect if an accident has taken place or not in the video. The LSTMs are enhanced Recurrent Neural Networks (RNN) that have an advantage over conventional feed-forward ANN and RNN, which is remembering patterns for long durations of time. A dataset which contains real-world traffic scenes was used to train and test the model. The test data included 48 videos (24 incident videos and 24 non-incident videos). The model was able to classify videos without any manual annotations and the performance of the model was evaluated using four metrics, which are Accuracy, Precision, Recall and F-score. The model achieved an Accuracy of 95% in detecting incidents, with a Precision of 75%, a Recall of 84%, and an F-score of 84% [127].

The imbalance of non-incident and incident samples in the training dataset reduces the performance of machine learning models for incident detection. Real-world traffic data contains significantly higher non-incident samples than incident samples so that the model cannot classify non-incident samples accurately. An incident detection framework was proposed to solve the problem of incident samples scarcity by using Generative Adversarial Networks (GNAs), which is a class of deep learning models [128], [129]. In these models’ new incident samples can be generated based on the features of real traffic data. These generated incident samples are combined with real non-incident samples to improve the diversity of the samples and balance the data. Based on some traffic parameters, a support vector machine classifier was applied to detect traffic incidents. To evaluate the proposed method, real-world traffic parameters were collected from I-80 Interstate highway in California and were used as input data. The performance of the developed traffic incident detection models was evaluated based on three measures: Accuracy, DR, and FAR. The results showed that using the real-world data in combination with the generated incident samples improved the accuracy of the model from 89.89% to 91.87 and increased the DR from 87.48% to 90.68%. In addition, the FAR decreased from 12.76% to 7.11% [129].

Recent studies have shown the effectiveness of ANN based algorithms in detecting traffic incidents using data from various sources such as traffic sensors and cameras. However, the potential of ANN algorithms to enhance traffic incident management can be further extended by integrating them with other emerging technologies. For example, Philip and Saravanaguru [130] proposed a system that utilizes a combination of ML algorithms and blockchain technology to provide real-time incident detection and response in an Internet of Vehicles (IOV) environment. The system integrates data from multiple sources, such as vehicle sensors, traffic cameras, and social media, to provide accurate incident classification and severity assessment. The use of blockchain technology ensures the authenticity and integrity of incident reports and evidence. While the implementation of the system poses several challenges, such as the need for standardization of incident reporting and data sharing protocols, the proposed system has the potential to significantly enhance incident management and improve road safety. However, one limitation of the proposed system is that it does not specify the exact type of ML algorithm to be used for incident classification and severity assessment. The authors suggest that the system can utilize various ML techniques, such as decision trees, neural networks, and support vector machines, to train and classify incidents based on their severity levels. These ML models will be discussed in more detail in the next sections, where some of the recent studies that have used these techniques in incident detection will be reviewed.

To further demonstrate the potential of deep learning algorithms in traffic incident detection, recent work by Zhang and Yang [131] presented a novel dataset and a deep learning-based approach for real-time traffic cones detection. The TraCon dataset contains over 10,000 images captured from various angles and lighting conditions, with annotations of the location and size of traffic cones. The proposed deep learning algorithm, which combines a Convolutional Neural Network (CNN) with a Region Proposal Network (RPN), achieved high accuracy and fast processing speed in detecting traffic cones. The authors suggest that their approach can be further extended to detect other types of traffic-related objects, such as road signs and pedestrians. By extension, the detection of vehicles using similar deep learning algorithms can also be utilized to detect incidents. For instance, the detection of stopped or slow-moving vehicles can indicate a possible incident, which can then trigger incident alarm. This work highlights the potential of deep learning-based algorithms to enhance incident detection and response, and the importance of developing comprehensive and diverse datasets for training and testing such algorithms.

2.4.2. Fuzzy Logic (FL) algorithms.

Fuzzy logic (FL) was first developed by Zadeh in 1960s [132]–[134]. The FL is a method of reasoning that imitates human reasoning in making decisions, it is utilized in case of incomplete, ambiguous or in accurate inputs [135]–[137]. The FL approach takes into consideration all possibilities between true and false to handle the concept of partial truth. Hence, the output value can be any real number between 0 and 1. In contrast to the Boolean logic that it is used in computer systems, where it takes a precise input and yields a definite output, there are only two possible values for the input either true or false represented as 0 or 1 [136], [138]. The FL mimics human reasoning and cognition by using Fuzzy Inference System (FIS) that is based on human interpretable rules instead of abstract mathematics' rules [139]. The FL consists of three main steps; first the input values are transformed into fuzzy variables through a fuzzification process, where they are expressed as linguistic terms. Second, a set of rules developed by experts is used to make inferences about these variables. Then, a membership function is used to represent the fuzzy set graphically and maps the resulting outputs in the fuzzy step through a defuzzification process, where they are converted eventually into crisp outputs [140], [141]. Figure 12 [142] summarizes the processes of FL system.

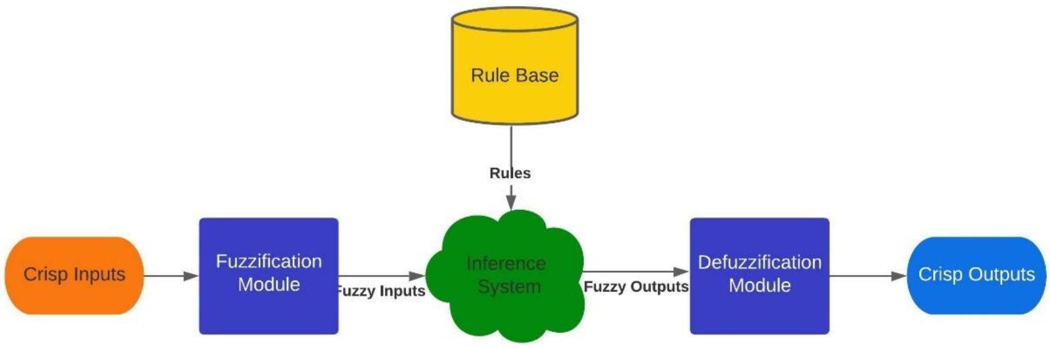

Figure 12 Architecture of fuzzy logic system [142].

Because of the nature of the FL, it is used to deal with the complex and stochastic nature of traffic variables. When it is used in incident detection, it doesn't give a clear signal of incident or no incident, it gives the likelihood for an incident. Several incident detection algorithms were developed based on FL and it showed good performance in terms of DR, FAR and MTTD measures [78].

Rossi et al. [143] developed an incident detection system based on FL. The proposed system consists of two FIS, which simultaneously analyze traffic data collected from two detectors placed upstream and downstream the road segment. Similar to comparative algorithms, they compare the observed traffic parameters to preset threshold values, which represent normal traffic conditions. However, in these systems the difference between observed traffic conditions and normal conditions is computed on the fundamental diagram of traffic flow. The two FIS compare the observed traffic conditions upstream and downstream with normal traffic conditions. If the output of the two FIS is in an abnormal state for two consecutive time intervals, then an incident alarm is triggered. The input variables for the two FIS are density and speed obtained from simulation data and the output is the state of the road section at a certain time period. The membership functions of the fuzzy sets are identified with K-means algorithms, where the input variables are clustered into k clusters with the nearest mean to minimize the within-cluster sum of squares [144]. To assess the effect of traffic flow conditions on the performance of the systems at different traffic conditions, four flow rates (1000, 2000, 3000 and 3500 vehicles per hour) were loaded into the systems. In addition, the effect of distance among detector stations were assessed by simulating three distances scenarios (100, 150 and 1000 meters). The performance of the systems was evaluated in terms of three indices DR, FAR and MTTD. The DR values of the model ranged from 51% to 100%, with the highest value achieved at a Traffic volume of 3000 and Distance among loops detectors of 1000. The FAR values varied from 0.03% to 0.96%, with the lowest value obtained at a traffic volume of 2000 and distance among loops detectors of 1000. The MTTD values ranged from 1.10 to 4.67 minutes, with the highest value achieved at a traffic volume of 2000 and distance among loops detectors of 1000 [143]. Based on the results, the DR of the model is generally high, except for low traffic volume conditions with a distance of 1000m between loop detectors. The FAR is very low, less than 1% in all cases, making it an ideal condition for the problem of freeway incident detection. The MTTD is on average lower than 5 minutes, and it increases as the distance among loops increases, starting from a minimum of 1.10 minutes. The analysis of some factors showed that the distance among loops is the most important factor affecting the AID system's performance. MTTD increases for all traffic conditions as the distance among loops increases, and DR decreases for low traffic volumes. Additionally, traffic flow conditions have an impact on the system's performance, with higher flow rate values associated with better DR and lower MTTD [143].

Ahmed and Hawas [145] utilized FL to develop an urban incident detection model. A microsimulation model was developed for a typical 4-leg signalized intersection, where incident and no incident scenarios were generated. The incidents generated were lane blocking incidents that last for at least 6 minutes to be more challenging to detect. Incident status was evaluated over various signal cycle times (60, 80 or 100 seconds), approach link lengths (300, 500 or 1000 m) and hourly traffic volumes combinations. Three detectors were fixed at upstream, midstream and downstream of each link. The traffic parameters that were collected are the speeds and vehicles count at the detector's locations for each link at every time step. Thus, three deviation parameters of vehicles count, and three deviation parameters of speed measures resulted and were used as independent variables in the Fuzzy logic model (FLM) to predict the incident status. Statistically significant test using ANOVA was used to determine the most significant independent variables. The test revealed that the most significant independent variables are those related to upstream and downstream only. These four independent variables were used as input variables of the FLM. A neuro-fuzzy system was used to predict incident status based on the inputs. The input variables go through the three steps of FL system as depicted in Figure 12 and the final output value (dependent variable) is either "yes", which is an incident case, or "no" which represents a normal traffic condition. The FLM allocated the central value of 1 for an incident and the central value of 0 for a non-incident status. A threshold value of 0.5 was utilized in this model, where if the dependent variable is greater than 0.5 an incident is indicated, otherwise it's non-incident condition. Based on the results, the average DR of the model was 64.3% and the average FAR was 7%. The results revealed that the proposed FLM had the worst performance (low DR and high FAR) at low traffic volumes scenarios.

In the same way, FL was implemented to develop an incident detection algorithm to detect traffic incidents in urban areas using detectors' cumulative count differences [146]. The PTV VISSIM microsimulation software was used to develop a model of a four approach-intersection to generate traffic data. Each approach consists of three lanes and each lane is equipped with three detectors upstream, midstream and downstream. The differences in the detectors' readings were used as input variables to the model. These input variables were then transformed into three linguistic terms: Zero, Positive and Very Positive. Then, they went through the processes of the FL system as illustrated earlier in Figure 12. The output of this model is an index that represents the possibility of the incident, either Low (low probability of incident) or High (high probability of incident). Three different data sets were generated from VISSIM and used to develop three Fuzzy models (FM1, FM2 and FM3) with different traffic volumes and incident durations. To test and validate the models developed they were run on three different validation datasets at which incidents were generated in these datasets. The performance of the three models were evaluated by calculating the difference between actual incident status and detected incident status and three measures were used for evaluation over the simulation intervals: good DR (which is represented by the difference of 0), no DR (represented by the difference of 1) and FAR (represented by the difference of -1). The percentage of good DR ranges from 54% to 79 while the percentage of no detection ranges from 9% to 20% and the percentage of the FAR ranges from 1% to 37%.

Lee et al. [147] implemented FL in developing incident detection algorithm for a signalized diamond interchanges. Based on four traffic measures (lane traffic volumes, queue length, speed and occupancy), the algorithm detects lane blocking that deteriorates traffic conditions and then reports these incidents to the system to determine the appropriate strategies to control the signal. The algorithm compares the current values of the traffic parameters with the past five minutes average values to determine any abnormal traffic conditions that indicate incident-induced congestion. The algorithm consists of four modules: (1) normality inference module, (2) incident location inference module, (3) incident severity assessment module; and (4) incident termination inference module. These modules run sequentially to check for abnormal conditions and detect any incidents. Each module has an FIS that takes two inputs: the five-minute average for each traffic measure and the percent of change between this average and the recent minute data. The membership function and the rules are used to convert these inputs into either abnormal or normal conditions through the FL process [147]. To evaluate the performance of the algorithm, a simulation model of urban diamond interchange was generated by TexSIM, a traffic simulation software. Incidents at different locations and three different traffic volumes were generated. The performance was evaluated in terms of DR, FAR and MTTD. The overall DR of the model was 73%. Interestingly, the DR varied according to the traffic volume: it was 62% for light volume conditions and increased to 79% for heavy volume conditions. Moreover, the severity of the incident also had an impact on the DR, with higher rates observed for incidents that blocked more lanes: 42% for 1 lane blocked, 86% for 2 lanes blocked, and 93% for 3 lanes blocked. The model produced 34 false alarms out of a total of 6120 decisions, resulting in a false alarm rate of 0.56%. As traffic volume increased, the false alarm rate also tended to increase: 0% for light volume, 0.74% for medium volume, and 0.93% for heavy volume. In terms of MTTD, it was found that the average time was 4.1 minutes. As the traffic volume increased, the MTTD tended to decrease, with times of 4.6, 4.0, and 3.7 minutes for light, medium, and heavy volumes, respectively. The severity of the incident also had an impact on the MTTD, with shorter times observed for more severe incidents: 4.4, 4.1, and 3.9 minutes for light, medium, and heavy volumes, respectively. This is because incidents that are more severe have a greater impact on traffic conditions, leading to more rapid congestion build-up that is easier for the model to detect [147].

2.4.3. Support Vector Machine (SVM) algorithms

The SVM is a supervised ML algorithm that was developed by Cortes and Vapnik [145] and can be used for classification and regression analysis. The SVM algorithm is given a set of training data that consists of labelled instance pairs each belonging to one of the two classes, so the algorithm learns the pattern of each class. The training vectors are selected to lie closer to the boundaries of the classes and they are called support vectors. The SVM finds the best line, hyperplane, which spilt these support vectors and has the maximum margin, i.e., the maximum distance between data points of the two classes. Then the SVM can assign new points to one of the two categories. Accordingly, SVM models have been used to detect traffic incidents[148]–[150] due to its great learning potential and because it is computationally efficient classifier that can detect incidents with high DR and low FAR [24], [151]. However, the accuracy of the SVM is highly dependent on the kernel function selected to handle non-linear classification problems. However, selecting an appropriate kernel function is complex and complicated. If a high dimension kernel function is used this might generate too many support vectors which can significantly decrease the training speed of the model. In addition, the SVM is not appropriate for large data sets because it will make the training process very time-consuming.

Motamed [24] examined the use of SVM to develop real-time incident detection during peak hours. Using field data, the speed, volume and occupancy of vehicles at upstream and downstream detector stations over a specific time step was used to recognize incident patterns and detect the occurrence of incidents. The proposed model was tested on a new site, and it achieved very satisfying results with DR around 89%, FAR about 7% hence the results showed that SVM can successfully classify traffic conditions as incident or non-incident. Likewise, Yuan and Cheu [37] investigated the application of SVM models in incident detection. Three SVMs with different were developed and applied for incident detection in arterials and freeways. Linear SVM denoted as SVM_L, and SVMs with polynomial denoted as SVM_P and SVM with radial basis kernel function denoted as SVM_RB. The performance of these SVMs were compared against MLF and Probabilistic Neural Network (PNN) models that were trained by the same training data set and tested with the same testing dataset. The performance was evaluated in terms of DR, FAR and MTTD as shown in Figure 13, Figure 14 and Figure 15.

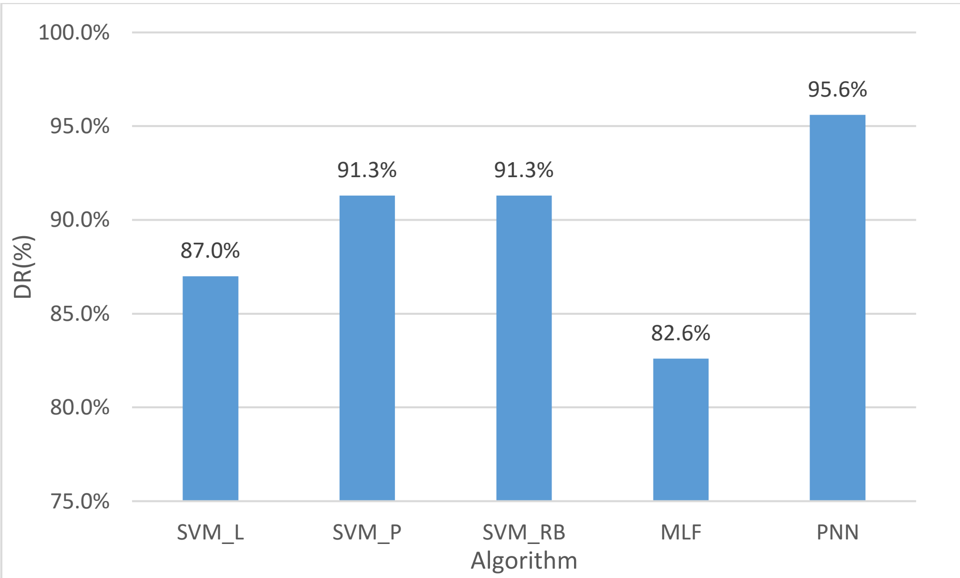

Figure 13 DR values of SVM, SVM_P, SVM_RB , MLF and PNN [37].

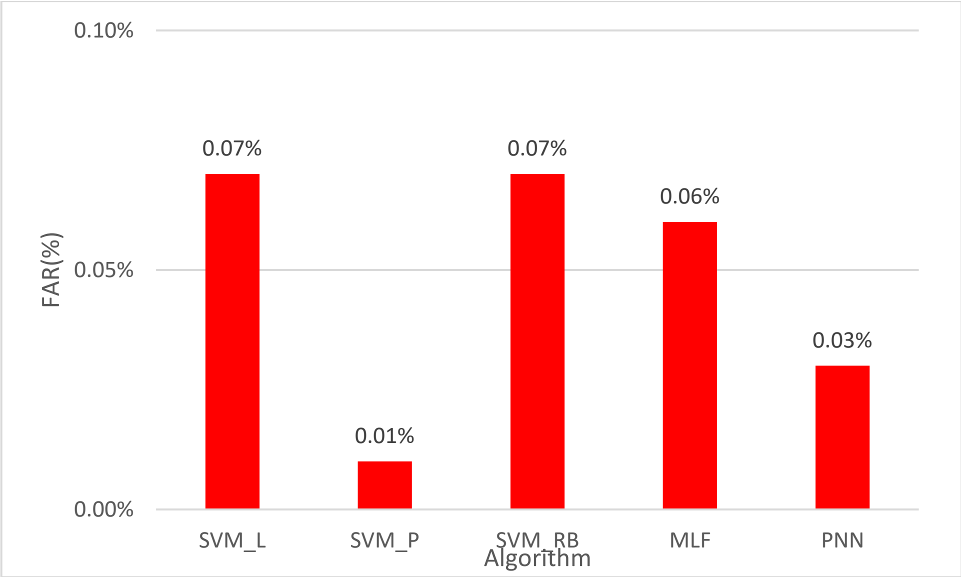

Figure 14 FAR values of SVM, SVM_P, SVM_RB , MLF and PNN [37].

Figure 15 MTTD values of SVM, SVM_P, SVM_RB , MLF and PNN [37].

Figure 13, Figure 14 and Figure 15 indicate that the PNN model has the highest DR of 95.6%, but also has a higher FAR (0.30 %) compared to MLF and SVMs. Also, it can be noted that three models, SVM_L, SVM_RB, and MLF have similar FAR values (0.06 or 0.07), while SVM_RB has the highest DR (91.3%) among them but a longer TTD of 327 seconds. The study showed that SVMs can achieve high DRs while maintaining low FARs similar to the MLF, thus SVMs can perform at least as good as MLF. Thus, the results showed the potential of applying SVM in incident detection.

2.4.4. Ensemble learning algorithms.

Ensemble learning is a branch of machine learning where multiple learning algorithms are combined to obtain better learning and predictive performance than single learning algorithms. Random forest classifier is the most widely used ensemble learning algorithms [152].

Random forest (RF) classifier is an ensemble learning algorithm that operates by constructing a forest of several decision trees that operates as an ensemble [153]–[155]. These decision trees can be either classification trees or regression trees hence RF can be used for both classification and regression problems. During the training phase, the RF uses sampling with replacement (bootstrap sampling) from the original data to create multiple subsets with the same number of observations as that of the original data [155]. Bootstrap sampling helps to improve the classification performance and the depths of the decision trees. At the testing phase, RF gets class prediction from each individual decision tree and the final predication output is based on majority voting this step is called bagging as illustrated in Figure 16. The advantage of RF that it creates uncorrelated (or relatively low correlated) individual decision tree classifiers, so it has superior performance than any of the individual constituent decision trees[155], [156]. The higher the number of decision trees in RF, the higher accuracy and robustness of the algorithm.

Liu et al. [158] constructed Naïve Bayes classifier ensemble for incident detection to improve the detection efficiency of the standard Naïve Bayes (NB). In addition, the performance of the standard NB depends mainly on the selection of the optimal threshold and tuning parameter. This is a challenging task since there is no structured method to select the optimal threshold and parameters of NB which can be determined by trial and error. So, the developed NB classifier ensemble was proposed to avoid the burden of choosing the optimal threshold of NB and tuning its parameters. The NB classifier ensemble trains several individual NB classifiers on incident and non-incident traffic patterns to construct classifier ensemble that was used to classify incident and non-incident state. Additionally, the authors combined Naïve Bayes and decision tree (NBTree) to develop another incident detection algorithm. The study evaluated the performances of three algorithms using both the I-880 Freeway dataset in San Francisco, California, and a simulated dataset with noisy data.

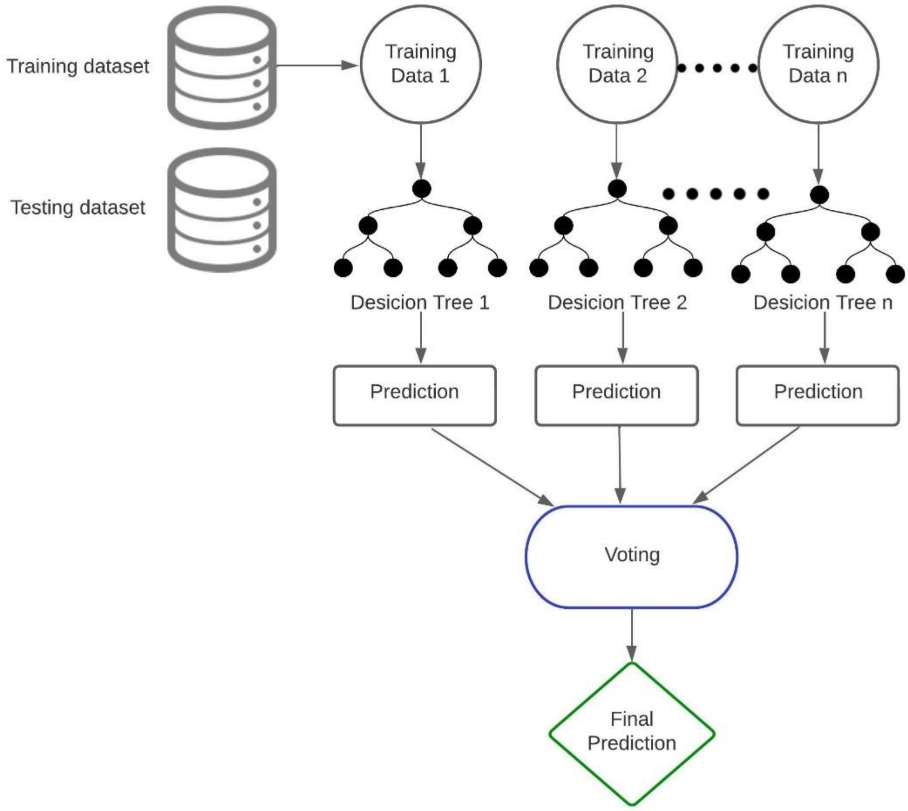

Figure 16 Random Forest algorithm [157].

These datasets included volume, occupancy, and speed data at upstream and downstream sites collected every 30 seconds for both incident and non-incident traffic conditions. The results for the I-880 dataset showed that the DR of the standard NB classifier was approximately 82%, with a FAR of around 0.04% and an MTTD of approximately 1.3 minutes. For NBTree, the DR was about 81%, with a FAR of around 0.01% and an MTTD of approximately 1.4 minutes. In comparison, the DR values for NB ensemble ranged from 82% to 89%, with a FAR ranging from 0.03% to 0.042%, and the MTTD varied from 1.6% to 1.8%. For the simulated data, the DR of the standard NB classifier was approximately 71%, with a FAR of around 0.05% and an MTTD of approximately 1.4 minutes. For NBTree, the DR was about 73%, with a FAR of around 0.03% and an MTTD of approximately 1.2 minutes. Similarly, the DR values for NB ensemble ranged from 64.6% to 79.6%, with a FAR ranging from 0.05% to 0.06%, and the MTTD varied from 1.4% to 1.9%. Overall, the study revealed that the performance of the NB ensemble was better than that of the standard NB and NBTree in terms of DR, FAR, and MTTD indexes [158].

Dogru and Subasi [149] used three machine learning algorithms ANN, SVM and RF to develop traffic incident detection models. Using SUMO traffic simulator, the speed, and the location of vehicles during normal and incident situation were collected and used as input of these algorithms. The performance of the three algorithms was compared in terms of DR and FAR. The ANN model achieved a DR of 86.1% and FAR of 8%, while the RF model achieved a DR of 94% and FAR of 0.203%. The SVM achieved a DR of 88% and FAR 4.2%. It can be observed that the RF model showed superior performance in terms of both DR and FAR. However, it is worth noting that the MTTD of these algorithms was not measured in this study. In addition, the proposed models use simulated data collected from Vehicular Ad-Hoc Networks (VANETs) based on the speeds and coordinates of the vehicles to detect incidents. However, their models make several assumptions that may limit their applicability in real-world scenarios. For instance, they assume that only probe vehicles are equipped with V2V communication devices, and that these vehicles broadcast their position and speed every second. Moreover, the models assume that probe vehicles are able to calculate the position of transferring vehicles using signal processing and antenna techniques and their own location, and that they can aggregate microscopic traffic values for each vehicle over the last 10 seconds.

Similarly, Ahuja [93] developed four algorithms to detect congestion and incidents using the sensor speed and occupancy data collected from Interstate Freeways I-74 of the Davenport region, in Iowa, USA. Two of these algorithms were based on Decision tree and Random Forest models, while the other two were based on statistical methods using IQD. The first IQD-based algorithm, IQD_speed, used only speed as a parameter, while the second algorithm, IQD_speed&occupancy, used both speed and occupancy. The performance of each algorithm was measured in terms of DR and FAR. The DR of IQD_speed ranged from 91% to 94%, while the DR of IQD_speed&occupancy ranged from 88% to 92%. The DR of the decision tree algorithm ranged from 94% to 97%, and the DR of the RF algorithm ranged from 94% to 97%. The FAR of IQD_speed ranged from 3.8% to 5.4%, while the FAR of IQD_speed&occupancy ranged from 2% to 4%. The FAR of the decision tree algorithm ranged from 1% to 3%, and the FAR of the RF algorithm ranged from 1% to 3%. Overall, the decision tree and RF algorithms outperformed the IQD-based algorithms in terms of both DR and FAR, but the IQD-based algorithms still achieved relatively high levels of accuracy [93].

Iqbal et al. proposed model stacking, an ensemble learning approach, to classify incidents and their associated severity level simultaneously [11]. This approach combines multiple classifiers to improve the performance and accuracy of the classification. Three incident types: accident, congestion, and reckless driving with their associated three severity levels, i.e., low, medium, and high were considered, which results in nine pairs. The model stacking consists of two levels, the first level classifies the incident severity, and the second level classifies the incident type. The performance of the proposed was compared against existing individual incidents classifiers in terms of severity and type classifications. The model achieved performance gains of 5%–56% for incident severity classification and 1%–14% for incident type classification when applied with different classifiers [11].

In 2022, Xie et al. developed a novel incident detection framework using Self-Adaptive Synthetic Oversampling(SASYNO), RF and Random Subspace K Nearest Neighbour (RSKNN) ensemble algorithm [25]. First, traffic flow, speed and occupancy were collected from highway loop detectors. Second, 57 initial variable sets are selected based on four aspects of spatial-temporal characteristics. Third, the SASYNO method is used to generate traffic incident samples and expand the amount of these samples to tackle imbalanced sample database. Then, RF is used for variables importance ranking to select feature top 30 variables for AID. Finally, the RSKNN ensemble algorithm was used to classify incident and non-incident samples. To validate the effectiveness of the developed model, empirical analyses were conducted using two real-world traffic datasets collected from the I-205 and I-880 highways. The results demonstrated the superior performance of our model, achieving a DR of 97.3% and FAR of 0.061% on the I-205 dataset and a DR of 94.7% and FAR of 0% on the I-880 dataset [25]. However, using any oversampling technique can potentially affect the reliability of the results because it artificially generates traffic incident samples. This could potentially result in unrealistic samples that do not represent the actual traffic incidents. This could lead to false positives or overestimation of the incident detection performance. This, in turn, may cause the model to perform well on the training data but poorly on new and unseen data, which is a sign of overfitting. To mitigate this problem, it is important to use appropriate techniques such as Cross-Validation and regularization to prevent overfitting and ensure the generalizability of the model to new data.

2.5. Video Image Processing Algorithms

These algorithms use a video processing software to analyse videos of real-time traffic captured by surveillance cameras to extract traffic parameters such as traffic volume, speed, occupancy or queue length to detect incidents [18], [47], [91].

A traffic incident detection system using image processing technology consists of two main modules, the first module is vehicle detection and the second one is moving vehicles tracking module. The vehicle detection module is based on background subtraction method and Gaussian background modelling. After setting the background model each pixel of the following video frames is compared to detect vehicles. Moving vehicles tracking module uses CamShift to judge the state of vehicles by analysing speed vector and other parameters and then setting certain thresholds. If the values of these parameters reach the thresholds, this will trigger an alarm and the system will identify the occurrence of a traffic incident. However, the algorithm can lose track of vehicles when there are too many moving vehicles or they move too fast [159]. Therefore, the procedure is combined with Kalman filter algorithm, which is a popular algorithm that has various applications that include tracking moving objects, where it can be used to estimate and predict the position and the speed of the vehicles over time [160]–[163]. Based on that, the Kalman filter algorithm was utilized to reduce the impact of this flaw and improve the robustness of this algorithm. This algorithm was tested and the DR varied from about 51% to about 75%. While the omission rate (the ratio of undetected vehicles in the video to the total number of vehicles in the video) ranged from 25% to about 49% and the FAR was between 12% to 18%. The results revealed that when the traffic flow is large or not stable this can affect the efficiency of the detection system [159].

Likewise, a video-based detecting and positioning of traffic incidents was developed [164]. The method used cameras installed above a road segment to monitor the road and extract traffic data. Each lane in the monitored segment is divided into a number of cells and each cell has a unique ID. The framework of this method consists of two main stages. The first stage is time-space information extraction of traffic flow, and the second stage is detecting and positioning of incident points. In the first stage, modified Gaussian Mixture Model (GMM) with adaptive update rate is used to model background image of the road segment and to extract traffic objects from the background. This modified GMM solves the problem of misidentifying suspended vehicles as background. Then for each vehicle, its coordinates are extracted using Kalman tracker. Using the extracted vehicle coordinates and its Frame IDs along its trajectory points in the video sequence, the space-time diagram of trajectories along the stream direction for each lane is generated. In the second stage, traffic parameters including traffic flow rate, average travelling speed and space occupancy rate are calculated from space-time diagrams and used to constitute a three-input fuzzy identification model that is used to identify the traffic state in each cell. The traffic states are classified into three states: smooth, slightly congested and congested. Next, a SVM classifier is used to detect and locate traffic incidents. The dataset used to train and test the SVM model consists of both real data and simulated data generated using VISSIM traffic simulator. In addition, oversampling technique was used to generate artificial incidents sample to balance the datasets used to train and test the model. The traffic state of a congested cell is taken together with states of upstream and downstream neighboring cells in the same lane to construct a feature vector. If the traffic state in the neighboring cells is smooth, then this is normal condition of queuing congestion otherwise the cell is judged to be an incident point. If the state in that cell remains congested for two successive periods, an incident is detected and its position is calculated using that cell's ID. Figure 17 summarizes the method [164]. A combination of traffic videos and simulation scenes were used for training and testing the SVM based detector. The results of this method were compared with the results of three typical existing algorithms: California algorithm, McMaster algorithm and AGANN algorithm which uses three layered backpropagation ANN to recognize incidents patterns form traffic parameters. Figure 18, illustrate the performance of each algorithm in terms of DR, FAR and MTTD.

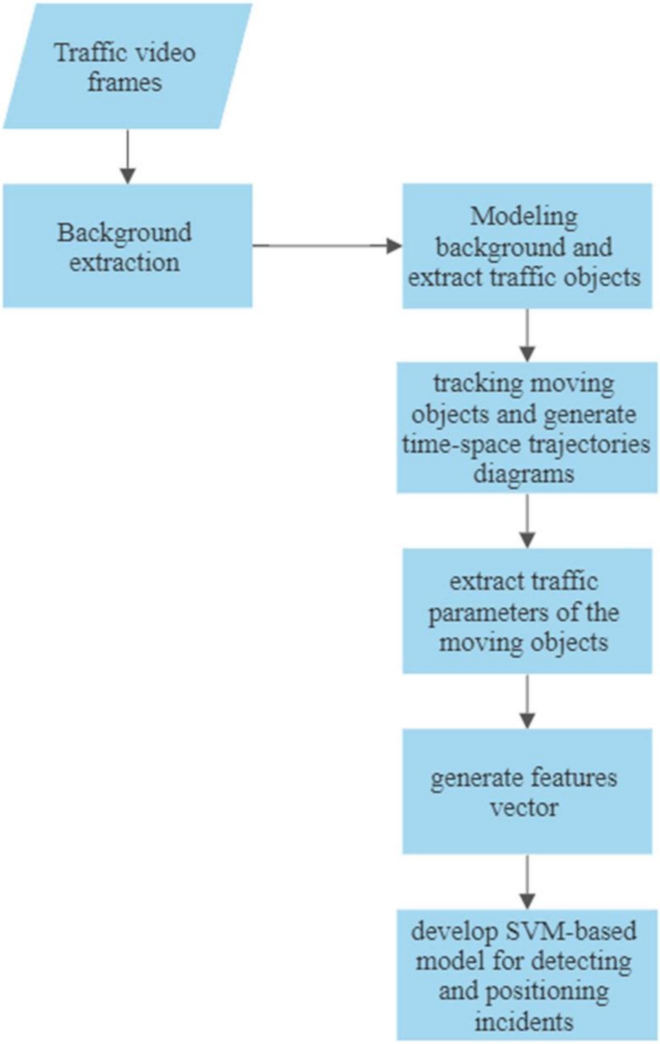

Figure 17 Detection and positioning of traffic incidents based on video processing [26 , 160].

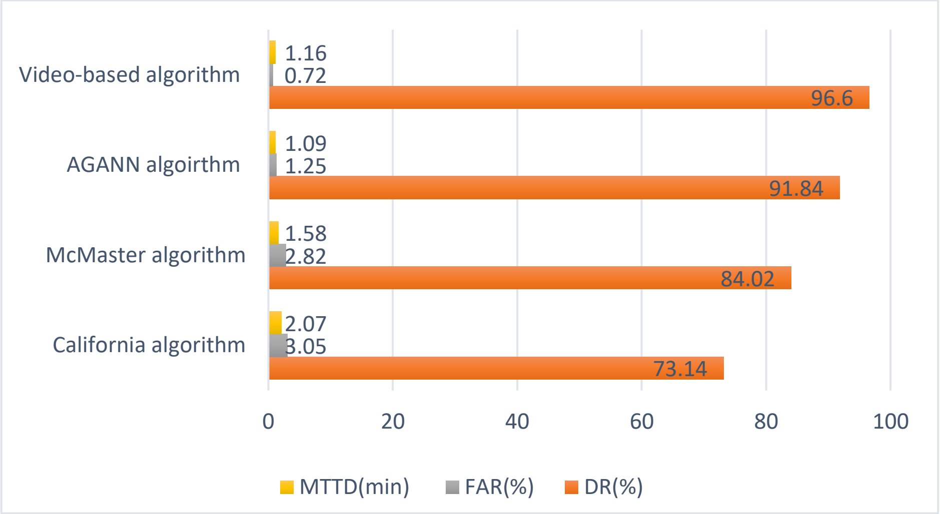

Figure 18 Performance comparison of the video-based algorithm and other algorithms [164].

It can be noted from Figure 18, the performance of this method was superior to the performance of the other algorithms except the MTTD of this method was slightly higher than that of AGANN. However, the lighting conditions, extreme weather conditions and coverage range of the camera that is used to capture traffic video are the main limitations of when using this procedure [164]. Additionally, using oversampling technique to generate artificial incident scenarios is not realistic and causes overfitting, as discussed before.

Table 2 below summarizes the incident detection algorithms categories.

Table 2 Summary of Literature-Reviewed incident detection algorithms and their pros and cons.

Category		Features & Advantages	Limitations and drawbacks
1. Comparative algorithms	California algorithms	• Looks for discrepancies in traffic parameters between adjacent loop detectors to identify the presence of an incident. • Good DR, tolerable FAR, well-known & recognized.	• Costly calibration, requires multiple detectors, prone to false alarms caused by natural changes in occupancy or equipment malfunctions, delayed detection time.
	McMaster algorithm	• Uses speed, flow, and occupancy to increase incident detection probability and reduce false alarms. • It has shorter MTTD, compared to California algorithm.	• Performance affected by weather conditions (e.g., snowstorms), limited testing/validation.
	Minnesota Algorithm	• Evaluates spatial occupancy difference between adjacent detector stations to detect incidents. • Collects average occupancies at upstream and downstream stations over 30 second intervals.	• May cause false alarms during low traffic conditions due to low occupancy values. • Detection time is three minutes or higher due to use of occupancy values from five intervals

		• Smooths high-frequency random fluctuations in traffic data by taking averages over short time intervals. • Produces the lowest false alarms compared to other algorithms.	following incident occurrence. • Uses only occupancy as input feature, potentially limiting its effectiveness in detecting certain types of incidents.
	Bluetooth based algorithms	• Uses Bluetooth detectors to measure travel time of vehicles. • More cost-efficient than inductive loop sensors or traffic radars. • Provides reliable, cost-efficient, and fast method for detecting traffic incidents or congestions.	• Minor incidents that last less than 30 minutes may not be detected. • Optimal detector distance varies based on traffic volume and location of incident.
	GPS based algorithms	Utilizes driver's mobile phones or GPS trackers in the vehicles to establish spatial-temporal traces of the vehicles to detect traffic congestion and incidents.	Range and placement of sensors affect efficiency and may cause false alarms.
	FCD based algorithms	• Uses probe vehicles to have real-time traffic data and detect the occurrence of incidents. • Real-time and fine-grained data on vehicle movements. • Cost-effective method. • Can be used in combination with loop detectors to	Limitations include penetration rate of tracked vehicles and data latency.

		improve incident detection performance.	
	V2V and V2I based algorithms	Uses V2V and V2I communications to monitor traffic and detect incidents and congestions	Impacted by the availability of the communications' protocols among different entities (vehicles and infrastructure)
2. Statistical Algorithms	SND algorithm	• Calculates the SND of traffic data to detect anomalies and incidents. • Uses consistency test to reduce false alarms. • High DR	• The presence of outliers can decrease DR and increase FAR. • tedious and time-consuming to determine thresholds.
	IQD based algorithm	• Uses the median instead of the mean, which avoids the masking problem of SND algorithm. • Uses IQD score, which is less sensitive to outliers than standard deviation.	• Suffers from the swamping problem when more than 50% of data stream values are equal, which can increase FAR. • Thresholds need to be obtained from historical data, which may not be readily available.
	DES algorithm	• Removes noise and heterogeneity from traffic data. • Provides short-term forecasting of traffic conditions. • Uses tracking signal to detect incidents.	• Assumes traffic will follow predicted pattern over time. • Requires extensive computational efforts. • Not extensively evaluated for different traffic scenarios.

	Time Series Algorithms	• Uses historical data of traffic variables for forecasting. • Detects incidents by identifying significant deviations.	• Time-consuming and requires extensive computational efforts. • Assume the traffic follows a predictable pattern over time. • May not perform well for different traffic scenarios if not evaluated properly.
3. AI Algorithms	ANN algorithms	• Can model complex and nonlinear relationships between input and output variables. • Can learn from historical data to detect incidents in real-time. • Can handle noisy and incomplete data. • Can be trained to detect specific types of incidents.	• The accuracy of the algorithm depends on the performance of the model that needs optimization and tuning. • May require extensive computational resources and time for training and prediction. • May suffer from overfitting or underfitting and may be prone to misclassification due to imbalanced data or class overlap. • There is no rule to determine the structure of the network, the appropriate structure is achieved through trial and error.

	Fuzzy logic algorithms	• Can handle imprecise or uncertain information. • Can model non-linear relationships. • Can incorporate expert knowledge through linguistic rules. • Can handle real-time and dynamic data streams.	• Require a significant amount of expert knowledge to create the rule base. • Can be difficult to interpret the output due to the use of fuzzy sets and linguistic rules. • May not perform as well in situations where the underlying data distribution is not well-understood
	Support Vector Machine	• Can learn complex decision boundaries for classification, which makes it a powerful machine learning tool. • Can handle high dimensional data well, which is important for processing large datasets. • Provides a computationally efficient nonlinear classifier that can be used in real-time incident detection, making it a suitable option for time-sensitive applications.	• The performance of the SVM is sensitive to kernel function used. However, choosing an appropriate kernel function is complex and difficult. • SVM is not suitable for large dataset.
	Ensemble learning algorithms	• Can combine the strengths of multiple models to achieve better overall performance. • Reduces overfitting and increases accuracy by	• Requires more computational resources than using a single model. • Choosing the right ensemble technique and

		aggregating the predictions of multiple weak models. • Can handle complex relationships between variables and large amounts of data.	parameters can be challenging. • Ensemble models can be difficult to interpret compared to a single model.
4. Video Image Processing Algorithms		Analyzes the motion of objects in a video stream to identify patterns of flow and movement, which can be used to detect incidents such as traffic jams or accidents thus they can be used in real-time.	• Can be sensitive to changes in lighting conditions or other environmental factors or low-quality video feeds. Which can decrease their accuracy. • May produce false alarms in response to non-incident changes in the background. • can be computationally intensive and require significant computing power.

Chapter 3. Models' Description and Methodology

The development of an efficient and reliable traffic incident detection system requires collecting traffic data to train the model to differentiate between normal conditions and abnormal conditions, which can indicate the occurrence of traffic incidents. The first section of this chapter discusses the challenges in this research. The second section discusses data source selection and the reasons for selecting it over other sources. The third section presents the study area that will be considered to develop the models. Then the experimental design for this dissertation is discussed where different machine learning algorithms are tested such as ANN and RF to select the most suitable algorithms.

3.1. Research Challenges

The first challenge is the process of collecting traffic data. There are two sources for traffic data, real data source and simulated data source. Due to the stochastic nature of the traffic, there is a need to monitor and collect the traffic data continuously to obtain detailed information and capture the variations in traffic variables during the day. Therefore, collecting real traffic data is a time and money consuming process. Instead, simulated data is used to overcome these difficulties and develop the system. In addition, traffic simulation enables the generation of various scenarios at different traffic flows and different incident scenarios. Hence, simulated data is used to develop the proposed incident detection models. The second challenge is to develop incident detection models that can detect incidents quickly with high DR and low FAR at the same time. This is one of the main challenges as these two rates are directly proportional, high DR increases FAR as discussed earlier in Chapter 2. Additionally, traffic flow is a nonlinear, stochastic, and very complex dynamic system and traditional modeling approaches struggle to capture its intricacies. For that reason, ML models are used as they can learn directly from data, without relying on a pre-existing model or equations, making them suitable for traffic analysis. This allows for the identification of complex and non-linear relationships among traffic variables, which can be used to predict future traffic conditions and incidents. By training ML models on historical traffic data, it is possible to identify patterns associated with normal traffic flow, as well as anomalies that may indicate incidents, such as accidents or congestion. These models can then be used to classify new observations in real-time, helping to improve traffic management and reduce the risk of accidents and delays. Moreover, these models can adapt to changes in traffic patterns over time and be fine-tuned with new data, making them suitable for both real-time traffic prediction and long-term planning. [165]. Also, they have adaptive learning and fault-tolerance capabilities [165]. However, the prediction accuracy of some of these models depends on the parameters and hyper-parameters of the model that needs optimization and tuning, which is another challenge that will be faced during the development of the models. Moreover, there are several factors can affect the efficiency of automatic incident detection algorithms, including but not limited to the severity of the incident, the distance between detectors or sensors, the incident location in relation to the detectors, and the traffic condition or congestion level [166]. These factors are complex and pose a challenge for considering all of them in a single model. None of the existing models have been designed to consider all of these factors together [78]. In this dissertation, these factors will be considered together to develop generic and effective incident detection models that can simultaneously take multiple variables into account, and the impact of these variables on incident detectability is studied.

3.2. Data Source

As mentioned before, there are two main sources of traffic data. The first source is real data that is collected from sensors and cameras. However, this source has some disadvantages. First, it utilizes expensive equipment in addition to the cost of maintaining this equipment. Also, it has limited coverage, so the data is available on where cameras and sensors have been installed. Moreover, the accuracy of this equipment is highly affected by weather conditions. Furthermore, real traffic data is not easily accessible to this research. Although databases such as Performance Measurement System (PeMS) [161] and Virginia 511 [162] databases provide real-time traffic information and are available to the public, they do not always provide detailed information about incidents such as their severity, location, and exact time of occurrence. However, the aim of this dissertation is not only developing accurate and efficient AID models, but also considering the various factors that can impact their performance. Therefore, it is important to investigate the impact of these factors on the performance of the AID models being developed. By doing so, A better understanding of the limitations and potential biases of these models can be gained, and strategies can be developed to mitigate their impact on their accuracy and effectiveness. The second source is simulated data generated by traffic simulators. There are several benefits of using simulated traffic data generated by a traffic simulator as the main source of traffic data instead of real data collected from sensors and cameras. Some of the potential benefits of using simulated data include:

- Cost-effectiveness: Simulated traffic data can be generated relatively inexpensively compared to the cost of installing and maintaining sensors and cameras to collect real traffic data.
- Flexibility: Simulated traffic data can be easily modified to reflect different scenarios or conditions, allowing researchers to explore a wide range of potential traffic patterns and situations. To effectively train ML models, a diverse and comprehensive dataset is necessary. This includes a variety of incidents of different severity, under different traffic flow conditions, and occurring in different locations. Unfortunately, such a dataset is not available and collecting it would be an extensive and impractical effort. As a result, simulation is being utilized to generate data that reflects these various conditions.
- Control: Simulated traffic data can be generated under controlled conditions, allowing researchers to precisely control variables such as traffic volume and road conditions. This can be useful for testing and evaluating the performance of different traffic management strategies.
- Privacy: While it is true that traffic sensors typically do not collect personal data, some forms of traffic data, such as video feeds or GPS data, may contain identifiable information about individual drivers, such as their license plate number, location, speed, direction of travel, and even their identity if they are captured on camera. This can raise concerns about privacy and data protection. By using simulated traffic data, researchers can avoid potential privacy concerns associated with the collection and use of real-world traffic data, while still being able to study and analyse traffic patterns and behaviour in a realistic and controlled setting. This can help ensure that research is conducted in an ethical and responsible manner, while still providing valuable insights into traffic flow and management.
- Another advantage of using simulated data is that the actual occurrence time of incidents (T0) is known. In real-world traffic data, the occurrence time of an

incident may not always be accurately recorded, as there may be a lag between the time an incident occurs and the time it is reported. This lag time can affect the accuracy of incident detection models, as they may rely on the assumption that the reported time of an incident is the same as the occurrence time [167]. Using simulated data allows researchers to control this variable and ensure that the occurrence time of an incident is accurately recorded.

- Moreover, simulated data is a recorded reality. This means that researchers can go back and review the simulated data in detail to better understand any unexpected results that may occur during analysis. Being able to visually inspect the simulation scenarios can be especially useful, as it allows researchers to see exactly what was happening in the simulated environment at different points in time. This can help researchers identify patterns or trends that may not be immediately apparent from analysing the data alone. While it is true that researchers can also review real-world traffic data in detail, even when traffic data is available, accessing and using that data can be complicated by issues related to privacy, data protection, and data sharing agreements. In some cases, access to traffic data may be restricted or controlled by government or other regulatory agencies, and obtaining the necessary permissions and approvals can be a lengthy and complex process. In contrast, simulated traffic data can be generated and analysed more easily and without the same level of regulatory or logistical challenges.

It is worth noting that simulated data is not a perfect substitute for real-world data, and there may be situations where real-world data is more useful for understanding certain phenomena and simulated traffic data may not always be as accurate as real traffic data collected from sensors and cameras. Real traffic data reflects the actual conditions and patterns of vehicle movement in a specific location, whereas simulated data is based on models and assumptions that may not always accurately reflect real-world conditions. As a result, simulated data may not always be as reliable or accurate as real traffic data in certain situations. However, simulated data can be a valuable tool for testing and evaluating traffic management systems and strategies in a controlled environment and can provide valuable insights into how these systems and strategies may perform in the real world. Therefore, traffic data generated from traffic simulation software is used to develop the proposed models. It should be noted that the preference of using simulated data in this study is due to its ability to provide controlled and repeatable experiments that are crucial for developing and testing new machine learning models. But the proposed models can also work with real data collected from roads using any of data collection methods identified in the literature and the collected data can be used as input data and the models should properly classify them. But the quality of the data being collected will impact the quality of the models' performance.

3.3. Study Area

The study considers a section of a major freeway in Dubai, United Arab Emirates, which is Sheikh Mohammad Bin Zayed Road (E311). The considered section has varying numbers of lanes, ranging from six to seven lanes per direction, and includes six junctions, consisting of two right-in-right-out junctions, a single point interchange, and two full-cloverleaf junctions with additional ramps as shown in Figure 19. These different types of junctions require distinct patterns of lane changes and weaving manoeuvres, which can create traffic perturbations and pose additional challenges for the developed AID models. The posted speed limit is 110 kph, and drivers are allowed to drive up to 130 kph without receiving speeding tickets. The length of the section being analysed is approximately 125 km.

Figure 19 The study area.

3.4. Experimental Design

In this section, a detailed description is provided of the experimental design used in this dissertation to develop generic incident detection models. To accomplish this goal, the VISSIM microscopic traffic simulation software developed by PTV is utilized to create a microscopic model of the proposed study area [168]. The developed model is then implemented to simulate normal traffic conditions and incident cases within the study area. During this simulation, the average speed of the vehicles, traffic flow rates, and detectors' occupancy at the upstream and downstream stations are collected every 30-second interval over a one-hour period for both normal and incident scenarios.

To generate a diverse dataset and give more challenging data to the ML models, four factors are considered when generating normal traffic and incident scenarios. These factors include the congestion level, the distance between the upstream and downstream stations, the location of the incident relative to the stations, and the severity of the incident.

The primary performance measure used for highways or freeways is typically the Volume-to-Capacity ratio (V/C) [169]. This conventional level-of-service measure compares roadway demand, represented by vehicle volumes, with roadway supply, represented by carrying capacity. The V/C ratio is a useful tool for measuring congestion on the road. However, it should be noted that in cases where the demand of vehicles exceeds the capacity of the freeway, the conventional V/C ratio may not accurately reflect the true congestion levels. In order to address this, the Demand to Capacity ratio (D/C) ratio is utilized as an alternative measure in this dissertation. The D/C ratio takes into account both scenarios and provides a more comprehensive representation of the congestion levels. Four D/C ratios are considered to represent different traffic congestion levels, including 0.6, 0.8, 1.0, and 1.2. These ratios represent uncongested conditions, congested conditions, very congested conditions, and oversaturated conditions, respectively. For each of these D/C ratios, three detector spacings are considered, which were 500m, 1000m, and 1500m. Additionally, three incident location cases are simulated, which include being closer to the upstream detector (quarter of the distance between the detectors), closer to the downstream detector (three quarters of the distance between the detectors), and in the middle between the two detectors. For the severity of the incidents, three incident severities are simulated, including light blockage (by blocking one lane out of the seven lanes), medium blockage (by blocking two lanes out of the seven lanes), and extreme blockage (by blocking five lanes out of the seven lanes). Furthermore, to create a more diverse dataset, four additional D/C ratios of 0.4, 0.5, 0.9, and 1.1 are added. In these scenarios, either the spacing scenarios, the location of the incident, or the severity of incident scenarios, or all of them are varied from the previous scenarios simulated for the first four D/C ratios. For the D/C ratios of 0.4 and 0.5, the three detector spacings of 500m, 1000m, and 1500m are also used, but the location of the incident is fixed to be 250 meters from the upstream station in all scenarios for these two D/C ratios. Moreover, two lane blockage patterns are considered, which are blocking three lanes and blocking four lanes, representing medium blockage. For the D/C ratios of 0.9 and 1.1, two detector spacings are used, which are 750 meters and 1250 meters, respectively. The incident location is simulated at 250, 375, and 625 meters from the upstream station. For the blockage patterns, three lane blockage patterns are simulated, which are two lanes, three lanes, and four lanes, representing light, medium, and high blockage. Overall, by varying the pattern of detector spacing, incident severity, and location of the incident, a diverse dataset is created that could challenge the ML models. These efforts are aimed at developing generic incident detection models that can differentiate between normal and abnormal traffic patterns in various scenarios.

In the following subsections, the generated normal condition and incident scenarios will be described in detail.

3.4.1. Generation of normal conditions.

In this section, the generation of normal traffic conditions is described. To simulate normal traffic scenarios, the combinations of D/C ratios and detector spacing that were, previously mentioned, are used. These scenarios are simulated without any incidents.

To ensure reliable results, each scenario in this dissertation is simulated for multiple runs with different seed numbers due to the stochastic nature of microsimulation. The number of runs varied across studies, with some using five [170]–[173], ten [174]–[176], or even twenty runs [177] per scenario. In this dissertation, normal scenarios are simulated for ten runs, while incident scenarios are simulated for five runs. Simulating normal scenarios for ten runs helped establish a baseline for comparison with incident scenarios and capturing a broader range of typical traffic conditions that are stable and predictable. However, incident scenarios are simulated for fewer runs due to their unpredictable nature, aiming to capture a range of possible outcomes rather than a comprehensive picture of traffic behaviour. Then, a trimmed average approach is used to average the results from the simulation runs, reduce the effect of randomness in the simulation results and remove possible outliers [172], [178].

Traffic parameters, including traffic speed, flow, and occupancy of the upstream and downstream stations, are collected during these scenarios. Each run is simulated 5400 seconds, with the first 900 seconds serving as a warm-up period to fill up the network with vehicles and establish normal traffic flow. The traffic parameters are collected over the following hour at 30-second intervals. The last 900 seconds of each run are used to ensure that the traffic flow continued until the end of the simulation period, resulting in a stable traffic flow during the middle 3600 seconds (one hour). This allowed for the collection of representative data during normal traffic conditions.

3.4.2. Generation of incident conditions.

VISSIM is a collision-free traffic simulator, and there is no direct way to create traffic incidents in VISSIM [179]. In this dissertation, incidents are generated by scheduling vehicles to make full stops in predetermined locations for 20 minutes (1200 seconds) from second 1800 to second 3000. These stopped vehicles blocked one or more lanes to create different lane blockage patterns and severity cases. For the D/C ratios of 0.6, 0.8, 1.0, and 1.2, the stopped vehicle blocked 1 lane, 3 lanes, and 5 lanes to represent light, medium, and extreme blockage, respectively. However, for the D/C ratios of 0.4 and 0.5, as well as 0.9 and 1.1, the blockage pattern is varied. Specifically, the stopped vehicle blocked 3 and 4 lanes for the D/C ratios of 0.4 and 0.5, and 2, 3, and 4 lanes for the D/C ratios of 0.9 and 1.1. This decision is made to create more diverse and challenging incident data for the models to learn from. By varying the blockage pattern among different D/C ratios, the model is exposed to a wider range of incident scenarios, which allows for a more robust and accurate performance during training and testing.

Several scenarios are simulated where an incident took place. Each scenario is a combination of one D/C ratio, one distance between detectors, one incident location case, and one incident severity case. Each scenario is simulated for five runs using different random seed numbers [170]–[173].

The traffic parameters during incident cases are collected from upstream and downstream detectors to represent incident conditions. In addition, the traffic parameters are collected before and after the occurrence of the incident and considered as normal conditions. The data collected before and after the incident is used to train the models to accurately identify the transition from normal to incident conditions and from an incident to normal conditions. This allows the models to quickly flag the occurrence of an incident, as well as the end of the incident, and to correctly classify the traffic situation as normal once the incident has been cleared. Additionally, the use of disturbed data in the training process helps the models to become more robust and adaptable to real-world scenarios where data may be incomplete or noisy.

3.4.3. Data preparation.

A total of 150 scenarios were simulated, of which 22 did not include any incidents, while the remaining 128 scenarios had incidents simulated. The traffic parameters were collected at 30-second intervals, resulting in 120 intervals per scenario, and a total of 18,000 intervals across all 150 scenarios. The data was collected before, during, and after the occurrence of incidents. This resulted in a total of 12,880 intervals representing normal conditions, while 5,120 intervals represented incident conditions. ML models typically involve three stages: training, validation, and testing. During the training stage, the ML algorithm learns from the input features and fits the model using a set of data known as the training dataset [180]. The validation dataset is a separate dataset that is held back from the training set and used to evaluate the model's fitness and skill in an unbiased manner, as well as to tune its parameters [181]. The test dataset is another separate dataset that has not been used for training or parameter tuning and is used for the final evaluation of the model's performance [182]. One method to split the full dataset into these three sets is to use a training set for training, a validation set for parameter tuning, and a test set for final evaluation. Another method is to use Cross-Validation, which involves dividing the full dataset into multiple folds or subsets, and using each fold in turn for validation while the other folds are used for training [183]. In this dissertation, a 5-fold Cross-Validation method is used, with 80% of the total number of intervals in the full dataset allocated to the Cross-Validation set, and the remaining 20% allocated to the testing set. Table 3 summarizes all incident scenarios that are simulated, and each cell indicates the location of the incident in meters relevant to the upstream station. The incident scenarios in this study are grouped based on their D/C ratios. Each group includes incident scenarios with one D/C ratio, as well as all possible incident locations, stations' spacings, and incident locations relative to the upstream stations. From each group, a subset of scenarios is selected for use as a dataset for testing. This dataset contains all variations in D/C ratios, lanes blockage, incident locations, and station spacing, as shown in Table 3. This approach ensured that the selected scenarios are representative of the range of possible incident scenarios and allowed for a thorough evaluation of the model's performance. It is important to note that the highlighted cells in Table 3 represent the scenarios that are used in the testing, while the remaining cells are used for Cross-Validation dataset.

Table 3 Summary of incident scenarios simulated.

Demand to capacity ratio	Blocked lanes	Spacing /distance of the incident from upstream detector				
		500	750	1000	1250	1500
0.4	3	250		250		
	4	250		250		
0.5	3	250		250		
	4	250		250		
0.6	1	125		250		375
		250		500		750
		375		750		1125
	3	125		250		375
		250		500		750
		375		750		1125
	5	125		250		375
		250		500		750
		375		750		1125
0.8	1	125		250		375
		250		500		750
		375		750		1125
	3	125		250		375
		250		500		750
		375		750		1125
	5	125		250		375
		250		500		750
		375		750		1125
0.9	2		375		625	
	3	250		250		
	4		375		625	
1.0	1	125		250		375
		250		500		750
		375		750		1125
	3	125		250		375
		250		500		750
		375		750		1125
	5	125		250		375
		250		500		750
		375		750		1125

1.1	2		375		625	
	3	250		250		
	4		375		625	
1.2	1	125		250		375
		250		500		750
		375		750		1125
	3	125		250		375
		250		500		750
		375		750		1125
	5	125		250		375
		250		500		750
		375		750		1125

The following section illustrates the use of RF to develop incident detection models.

3.5. Implementation of Random Forest for Incident Detection

RF is a powerful ensemble learning algorithm that combines multiple decision trees to classify the data sets. In this dissertation, RF is selected as the method to classify input traffic parameters and factors as normal or incident cases due to several reasons. Firstly, RF is a popular and powerful ensemble learning algorithm that has shown promising results in various classification tasks, including traffic incident detection. Secondly, RF is well-suited for handling high-dimensional data sets with a large number of features, which is the case for traffic data. Thirdly, RF is a non-parametric method that does not assume any underlying distribution of the data, making it robust to outliers and noise in the data. Finally, RF has the ability to handle both categorical and continuous variables, which is an essential feature for traffic data that may contain a mixture of both types. The RF model is built using several key variables that are considered to have an impact on the target variable. These input variables include the traffic flow, speed, and occupancy of the upstream and downstream stations at 30-second intervals over a 1-hour period, the spacing between the stations, the difference in flow, speed, and occupancy between the upstream and downstream stations, as well as the relative differences in flow, speed, and occupancy at the upstream station and the downstream station.

The RF model during the training phase also creates multiple subsets with the same number of observations as that of the original data. To ensure that the performance results obtained would be the same if the developed model was run on any computer machine, a certain seed number is selected for all three processes (data splitting, Cross-Validation, and RF). In this case, a seed number of 100 is used. The use of RF in this dissertation offers several advantages such as high accuracy, robustness and the ability to handle large datasets. Furthermore, RF is able to handle missing data, outliers and noisy data, which is important in real-world traffic incident detection scenarios.

The RF model utilizes 16 input variables to predict incident detection. These variables include:

1. Flow Rate at Upstream Station – The flow rate measurement at the station located upstream of the incident location.
2. Flow Rate at Downstream Station – The flow rate measurement at the station located downstream of the incident location.
3. Flow Rate Difference between Upstream and Downstream Stations – The difference in flow rate measurements between the upstream and downstream stations.
4. Flow Rate Difference Relative to Upstream Station – The difference in flow rate measurement at the downstream station relative to the upstream station.
5. Flow Rate Difference Relative to Downstream Station – The difference in flow rate measurement at the upstream station relative to the downstream station.
6. Speed at Upstream Station – The speed measurement at the station located upstream of the incident location.
7. Speed at Downstream Station – The speed measurement at the station located downstream of the incident location.
8. Speed Difference between Upstream and Downstream Stations – The difference in speed measurements between the upstream and downstream stations.
9. Speed Difference Relative to Upstream Station – The difference in speed measurement at the downstream station relative to the upstream station.
10. Speed Difference Relative to Downstream Station – The difference in speed measurement at the upstream station relative to the downstream station.
11. Occupancy at Upstream Station – The occupancy measurement at the station located upstream of the incident location.
12. Occupancy at Downstream Station – The occupancy measurement at the station located downstream of the incident location.

13. Occupancy Difference between Upstream and Downstream Stations – The difference in occupancy measurements between the upstream and downstream stations.
14. Occupancy Difference Relative to Upstream Station – The difference in occupancy measurement at the downstream station relative to the upstream station.
15. Occupancy Difference Relative to Downstream Station – The difference in occupancy measurement at the upstream station relative to the downstream station.
16. Distance between Upstream and Downstream Stations – The distance between the upstream and downstream stations.

These 16 input variables capture traffic characteristics and enable the models to detect changes in traffic flow and speed that are indicative of an incident. By incorporating these variables into the models, they will be able to accurately predict the occurrence of an incident and in a timely manner. For the implementation of the RF model, RapidMiner is used, which is an open-source data science platform that provides a user-friendly interface for developing and optimizing ML models. The results of this model are presented and discussed in Chapter 4.

3.5.1. Hyperparameter optimization for Random Forest model.

ML models have two types of variables: namely parameters and hyperparameters.

- Parameters: These are variables of the models that are estimated by fitting the given data to the model and they change continuously during the training process of the model, and they are required to make predictions. They are not set manually, the final parameters are determined at the end the training process of the model [184]. Examples of model parameters include the weights and bias in ANN or slope and intercept in linear regression.
- Hyperparameters: These are the parameters whose values are set before the training process of the model. They can't be learned by fitting the model to the data, they have to be set manually [184]. In addition, they are required to estimate the model parameters. Examples of hyperparameters include learning rate of ANN, the number of trees in RF and the kernel function in SVM. Setting very high values of hyperparameters can cause overfitting

while setting very low values of hyperparameters can cause underfitting. Thus, tuning hyperparameters is necessary to achieve the best classification accuracy. The optimal hyperparameters are impossible to determine ahead of time. They are determined based on trial and error. Identifying the values for the hyperparameters is called hyperparameter optimization.

The most widely used strategy to search for optimal hyperparameters is the grid search [185]. In grid search a subset of the hyperparameters is specified and then the performance of the model is evaluated over different combination values of hyperparameters. Usually Cross-Validation is used to evaluate the performance of the model [185]. Obtaining the optimal hyperparameters allows for an unbiased evaluation of the model's performance and helps to prevent overfitting and underfitting. In this dissertation, the optimization of the hyperparameters of the RF model is approached in a stepwise manner. Optimizing all the hyperparameters simultaneously can be computationally expensive, especially with large datasets. Therefore, to avoid this issue, the hyperparameters are optimized one at a time. However, optimizing hyperparameters one at a time can result in local optimal or near-optimal values instead of the global optimal values. The main hyperparameters that are optimized in this dissertation are the number of trees, the maximal depth, the subset ratio, and the confidence parameter of pruning. The number of trees in the RF model determines the number of decision trees that are created and used in the final prediction [155]. Increasing the number of trees can improve the accuracy of the model, but it can also increase the computational time and complexity. The maximal depth hyperparameter controls the maximum depth of each decision tree in the forest [186]. The deeper the tree, the more complex the model becomes and the higher the risk of overfitting. Therefore, it is important to balance the complexity of the model with the need for accurate predictions. The subset ratio hyperparameter controls the number of randomly selected attributes used in each step of creating decision trees in the RF algorithm. This parameter plays a crucial role in reducing the likelihood of overfitting, which can occur when decision trees are too complex and fit the training data too closely. By introducing a degree of randomness in the attribute selection process, the subset ratio helps to ensure that the decision trees are diverse and generalize well to new data, ultimately improving the overall accuracy of the model [187]. A higher subset ratio implies that more features will be picked randomly at each split, whereas a lower subset ratio indicates the selection of fewer features. However, using a higher subset ratio may result in overfitting, where the model becomes too complicated and tightly fits the training data, leading to poor performance on new data. On the other hand, using a lower subset ratio may result in underfitting, where the model is too simplistic and fails to capture the underlying data patterns. The confidence hyperparameter of pruning controls the minimal standard deviation that is allowed between the predicted and the actual values [155]. If the deviation is below the confidence level, the branch of the decision tree will be pruned. This helps to prevent overfitting by removing branches that do not contribute to the accuracy of the model. By optimizing each of these hyperparameters one at a time, the model can be fine-tuned to achieve the best balance between accuracy and computational efficiency. Each step of hyperparameter optimization, where multiple evaluations of hyperparameters are performed, will be referred to as an 'Iteration', while the sub-steps within each iteration where a single set of hyperparameters is evaluated, will be referred to as 'Evaluation Steps'.

RF Model Optimization procedure:

In order to optimize the hyperparameters of the RF model, the following optimization process is followed:

The optimization process for the hyperparameters of the RF model involves several iterations to determine the near optimal values for different parameters. In the first iteration, the range of 100-500 is considered to find the near optimal number of trees that maximizes the F-score of the model. The "Optimize Parameters (Grid)" operator in RapidMiner is utilized to find the local optimal values of the selected hyperparameters for the operators in its subprocess. In the next iteration, after determining the near optimal number of trees, the same operator is used to find the maximal depth of the trees. The maximal depth of the trees is searched within the range of 25-100 to maximize the F-score. Then, the subset ratio that achieved the highest F-score in Cross-Validation is searched for within the range of 0.05 – 0.95. In the following iteration, the confidence value that maximizes the F-score is searched for using the number of trees, maximal depth, and subset ratio that resulted from the previous iterations. This is done within the range of 1.00E-07 – 0.5. Throughout the optimization process, the F-score is used to evaluate the performance of the model. The resulting F-score is compared with the previous F-score. If the F-score improves, the previous iterations of the optimization process are repeated multiple times until the final hyperparameters of the model are found. However, if the F-score decreases or stops improving for consecutive iterations, the optimization process is terminated. Figure 20 below illustrates the flowchart of the optimization process used to fine-tune the hyperparameters of the RF model.

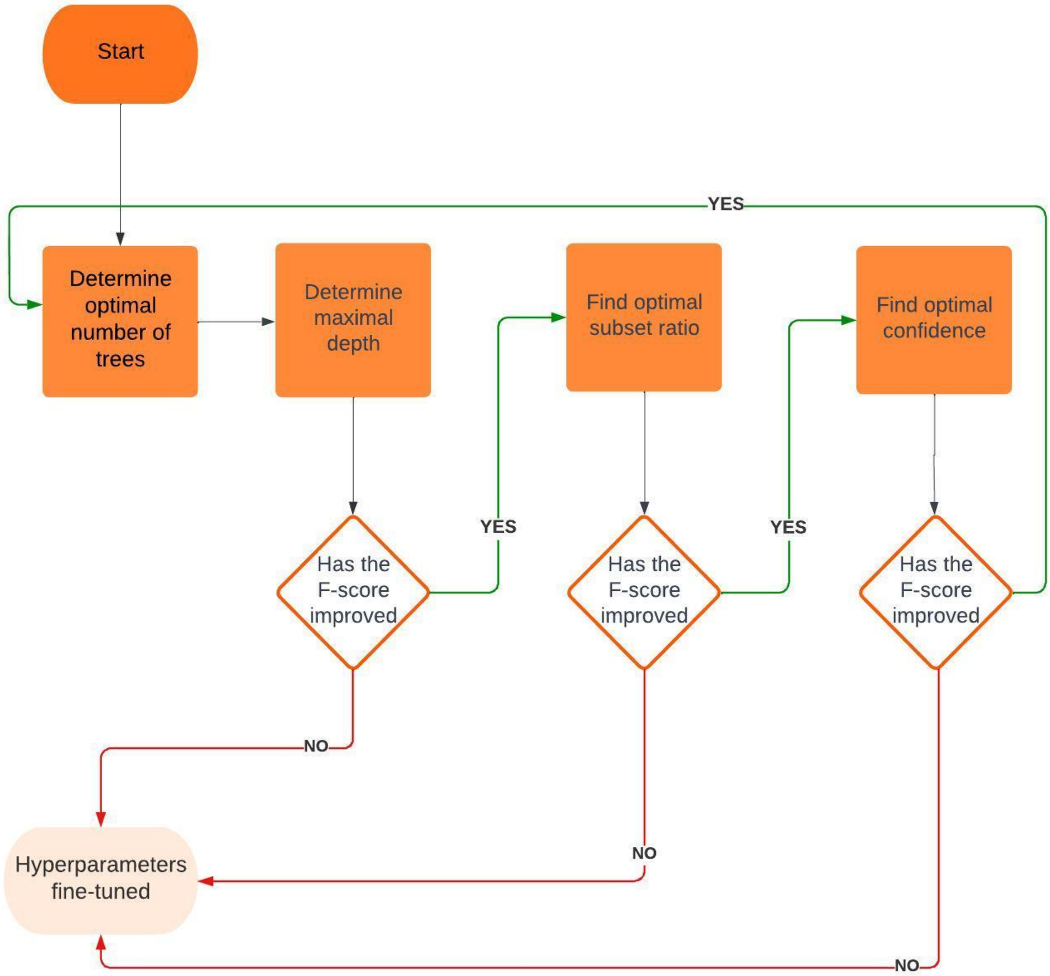

Figure 20 Optimization process for RF model.

3.6. Implementation of Artificial Neural Network for Incident Detection

As mentioned in section 2.4.1, Artificial Neural Networks (ANN) based algorithms. ANNs are computational models that are designed to mimic the neural network of the human brain. ANNs are used in a wide range of applications, including incident detection. One of the most used types of ANN is the Multilayer Feedforward ANN, also known as MLF. The MLF is used for modelling classification and regression problems [188], [189].

MLF is a popular type of artificial neural network used in various applications including incident detection. The structure of an MLF consists of an input layer, one or more hidden layers, and an output layer as shown in Figure 21. The input layer receives the input data, which is multiplied by the weights assigned to the connections. The weighted sum is then passed through an activation function, which transforms the result to output values that are used as inputs for the neurons in the next layer. This process, known as feedforward propagation, is repeated over and over again until the output layer is reached [190].

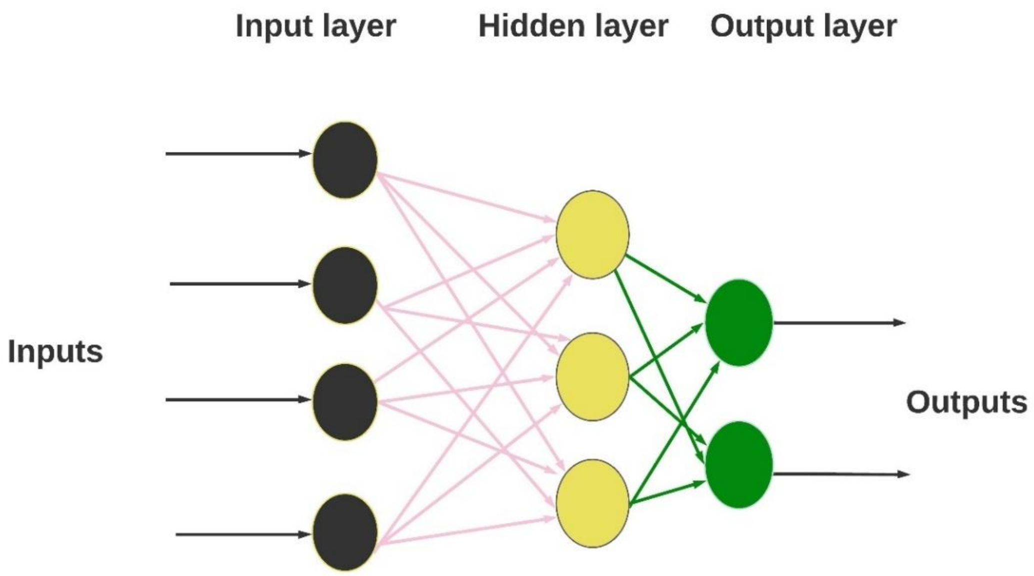

Figure 21 Structure of MLF neural network [191].

During the training phase of an MLF, the model learns from the data by adjusting the weights and bias iteratively through an optimization algorithm. The most widely used optimization algorithm is Stochastic Gradient Descent (SGD), which minimizes the loss function of the model and makes it as close to zero as possible [192]. The loss function measures the difference between the predicted outputs and the true labels for the inputs. The two most common loss functions used in MLFs are Mean Squared Error (MSE), which is the average of squared differences between the predicted values and the actual values [193], and Cross-Entropy, which measures the difference between the output probabilities of the network and the true values[194], [195].

Backpropagation is the most commonly used algorithm to train MLFs [196]. During training, the data with its labels are fed into the model in the input layer, and the model classifies the inputs and predicts their labels. At the end of each epoch, the loss or the error of the model's predictions is calculated, and the weights and bias are updated accordingly [197]. This is achieved by calculating the slope or the gradient of the loss function with respect to each weight at the end of each epoch during training and updating the weights and bias accordingly along the opposite direction of the gradient to minimize the loss [116].

MLFs also use activation functions, which introduce non-linearity into the model. One of the most commonly used activation functions is the Rectified Linear Unit (ReLU), which returns the input value if it is positive and zero otherwise. Another popular activation function is the Sigmoid function, which returns a value between 0 and 1, and it is often used in the output layer to normalize the outputs to a probability distribution over predicted output class. SoftMax function is another commonly used activation function in the last layer to normalize the outputs of the network so that the probabilities add up to 1 [198], [199].

Based on the literature, MLF models have been widely used in several studies to develop AID systems, showing promising results in terms of DR, FAR, and MTTD. Moreover, the MLF model has several advantages over other ML models. For example, it is able to generalize well to new, unseen data, making it ideal for applications where the input data distribution is likely to change over time. Additionally, it can handle high-dimensional input data and is capable of extracting features automatically from raw input data, reducing the need for manual feature engineering. Given these advantages, it is clear why the MLF model is a good choice for developing an AID system in this dissertation. By leveraging the strengths of the MLF model and using it to classify traffic status based on input traffic measurements, the resulting AID model is likely to be highly accurate, reliable, and capable of detecting incidents quickly and efficiently.

In this dissertation, MLF is used to develop an AID model that takes input variables, which are traffic measurements, and classifies the traffic status as either normal or incident condition. The developed MLF model serves as a benchmark to compare the results of the RF model, as both are trained and tested on the same dataset and use the same input variables as mentioned in Section 3.5. Furthermore, the DR, FAR, and MTTD of the developed MLF model are compared to those of other MLF models in the literature, in addition to comparing the structure and input variables used. RapidMiner is used to develop and optimize the MLF model.

3.6.1. Hyperparameter optimization for Artificial Neural Network model.

This section discusses the hyperparameters of the MLF model and the optimization process. The performance of the MLF model depends largely on its parameters and hyperparameters, and it is important to understand their impact on the model's accuracy. Hyperparameters are the settings used to train the model and can be adjusted to optimize the model's performance. The hyperparameters that are optimized in this dissertation are the number of hidden layers, the size of hidden layers, the learning rate, momentum, error epsilon, and training cycles.

- Number of hidden layers: This refers to the number of layers between the input and output layers in the neural network. Increasing the number of hidden layers can improve the model's ability to learn complex patterns in the data, but it also increases the risk of overfitting [193], [198], [200]–[206].
- Size of hidden layers: This refers to the number of neurons in each hidden layer. Larger hidden layers can capture more complex features in the data, but also require more computational resources and may lead to overfitting [193], [198], [200]–[206].
- Learning rate: This determines how much the weights of the neural network are updated during training. A larger learning rate can result in faster convergence but may also cause the model to overshoot the optimal weights and perform poorly [193], [198], [200]–[206].
- Momentum: This is a technique used to accelerate gradient descent by adding a fraction of the previous weight update to the current update. This can help the model avoid getting stuck in local minima during training [193], [198], [200]–[206].
- Error epsilon: This is a small value used to determine when the neural network has converged to the optimal weights. If the difference in error between iterations is less than the error epsilon, training is stopped [193], [198], [200]–[206].

- Training cycles: This refers to the number of times the entire training dataset is used to update the weights of the neural network. Increasing the number of training cycles can improve the model's accuracy, but also increases the risk of overfitting [193], [198], [200]–[206].

To optimize these hyperparameters, a stepwise approach is adopted. First, a range of values for each hyperparameter is specified. Then, the "Optimize Parameter (grid search) "operator in RapidMiner is used to search among this range for the optimal value of each hyperparameter. By testing different combinations of hyperparameters, the optimal settings for the MLF model are identified. The description and results of the hyperparameter optimization is discussed in this section.

MLF Optimization procedure:

The optimization of the MLF model is approached in a stepwise manner. First, the number of hidden layers and the size of each layer must be determined. This is done by testing different combinations of one and two hidden layers, with each layer's size varying from 5 to 50 neurons. The model's performance is then evaluated based on the F-score to determine the number of hidden layers and their sizes that yield the highest F-score. Once the near optimal number and size of hidden layers have been identified, the learning rate is optimized using the "Optimize Parameters (Grid)" operator in RapidMiner. The learning rate is tested within a range of 0.001 to 0.95 to find the value that maximizes the F-score. Next, the error epsilon value is optimized within a range of 1.00E-10 to 0.1 to identify the value that maximizes the F-score. This value is used to train the MLF and improve its performance. Finally, the training cycles are optimized by testing different values ranging from 50 to 1000. The value that yields the highest F-score is selected as the near optimal training cycle. At each iteration of the optimization process, the F-score is used to evaluate the performance of the MLF. If the F-score improves, the previous iterations of the optimization process are repeated multiple times until the final hyperparameters of the model are identified. However, if the F-score decreases or stops improving for consecutive iterations, the optimization process is terminated. Figure 22 summarizes the optimization process of MLF model.

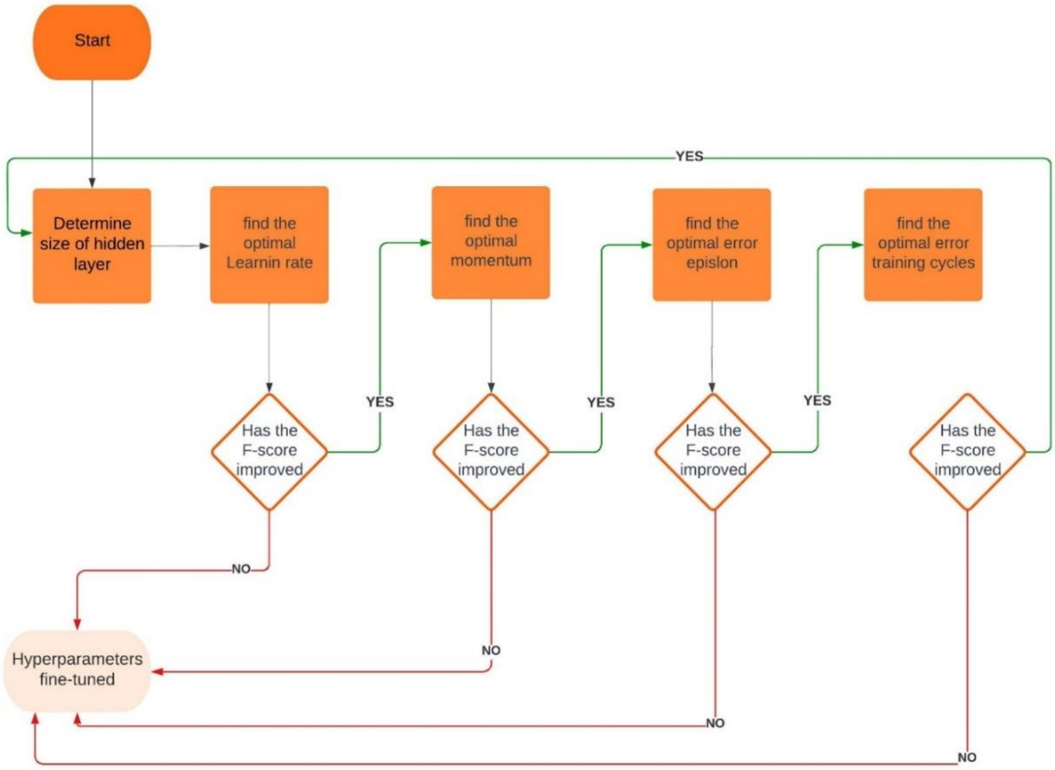

Figure 22 Flowchart of optimization process for MLF model.

Chapter 4. Results and Discussion

In this Chapter, the results of the optimization process of the RF and the MLF models developed in this dissertation are presented. In addition, this chapter provides a comprehensive examination of the developed models' performance, including the evaluation of the DR, FAR and MTTD. The confusion matrices of the Cross-Validation and testing processes are presented and discussed in detail to better understand the strengths and limitations of the models. Additionally, a comparison is made between the performance of the models and some of the existing AID models. The insights gained from this analysis serve as the foundation for future improvements and developments in this field.

4.1. Analysis and Discussion of Random Forest Model Results

In this section, the outcomes of the analysis and discussion of the RF model results are presented. Firstly, the hyperparameters of the RF model are fine-tuned, and the results of this process are reported. Subsequently, the optimized hyperparameters are utilized to perform Cross-Validation and testing phases, and the results of these stages are analysed. The primary objective of this analysis is to evaluate the performance of the model and examine the impacts of the factors considered in this dissertation on the model's efficacy.

4.1.1. Optimization results of random forest.

As previously discussed, the performance of ML models is heavily influenced by the choice of hyperparameters. In this section, the results of the hyperparameters' optimization process for the RF model are presented. The objective of this process is to identify the near optimal values for the hyperparameters that maximize the model's performance metrics. To achieve this objective, a systematic approach is employed, as illustrated in Figure 20, to explore different combinations of hyperparameters and evaluate their impact on the F-score of the model. The impact of the hyperparameters on the performance metrics is assessed using the Cross-Validation techniques to ensure that the optimization results are robust and not overfitting to the training data. The resulting values of the hyperparameters at the end of each iteration and the confusion matrix of the model are reported. Additionally, the final optimized values for the hyperparameters and the confusion matrices of the model during Cross-Validation and testing processes using the optimized hyperparameters are also presented. The following steps have been performed to achieve the optimized hyperparameters:

1. The initial model was set up with 100 trees, maximal depth of 30, pruning option disabled, guess subset ratio option enabled, and criterion hyperparameter set to information gain criterion. Table 3 presents the confusion matrix of the initial model. The Recall of the normal class is 99.01%, which means that the model correctly identifies 99.01% of actual normal cases, and only 0.99% of normal cases are falsely predicted as incident cases (false negatives). The Recall of the incident class is 85.05%, which means that the model correctly identifies 85.05% of actual incident cases, and 14.95% of incident cases are falsely predicted as normal cases (false positives). The Precision of the normal class is 94.64%, which means that out of all predicted normal cases, 94.64% are actually normal, and 5.36% of predicted normal cases are actually incident cases (false positives). The Precision of the incident class is 96.98%, which means that out of all predicted incident cases, 96.98% are actually incidents, and only 3.02% of predicted incident cases are actually normal cases (false negatives). The Accuracy of the model is 95.2%, which means that the model correctly predicts 95.2% of all cases (both normal and incident). However, Accuracy can be misleading when the classes are imbalanced, as mentioned before. The resulted F-score, which is the harmonic mean of Precision and Recall, of the Cross-Validation is 90.62% as shown in Table 4.

Table 4 Confusion matrix of the initial unoptimized model.

	True normal	true incident	class precision
pred. normal	10455	592	94.64%
pred. incident	105	3368	96.98%
class recall	99.01%	85.05%	
Accuracy	95.2%	F-score	90.62

2. The near optimal number of trees is found to be 250 through a grid search conducted within the interval of [100-500] trees, with an increment of 10, using the optimize parameter operator. This resulted in a slight increase in the F-score

to 90.82%. Table 5 displays a sample of the values tested for the number of trees parameter using the optimize parameter operator. The full range of values tested, from 100 to 500, can be found in Appendix A -Table A 2.

Table 5 Sample of Number of Trees Used in Grid Search for Model Optimization.

Evaluation step	Number of trees	Accuracy	Precision	Recall	F-score
1	100	0.9519284	0.96977425	0.850253	0.906059
2	110	0.9525482	0.97038637	0.85202	0.907331
3	120	0.9525482	0.97012561	0.852273	0.90735
4	130	0.9528237	0.97042655	0.85303	0.907912
5	140	0.9524105	0.96984095	0.85202	0.907081
6	150	0.9524105	0.97010913	0.851768	0.907057
7	160	0.9525482	0.97066892	0.851768	0.907299
8	170	0.952686	0.97068089	0.852273	0.907598
9	180	0.9529614	0.97123523	0.852778	0.908132
10	190	0.9526171	0.9706647	0.85202	0.907457
11	200	0.9524793	0.97064776	0.851515	0.907161
12	210	0.9528237	0.97122716	0.852273	0.907841
13	220	0.9528237	0.97094991	0.852525	0.907865
14	230	0.9527548	0.97121522	0.85202	0.907694
15	240	0.9528926	0.97150159	0.852273	0.907961
16	250	0.9530303	0.97179118	0.852525	0.908232
17	260	0.9528926	0.97177763	0.85202	0.907937
18	270	0.9528926	0.9717781	0.85202	0.907941
19	280	0.952686	0.97148229	0.851515	0.907524
20	290	0.952686	0.97067347	0.852273	0.907599
21	300	0.9526171	0.97066313	0.85202	0.907451
22	310	0.9526171	0.97120259	0.851515	0.9074
23	320	0.9526171	0.97093886	0.851768	0.907425
24	330	0.9527548	0.97150396	0.851768	0.907663
25	340	0.9526171	0.97093661	0.851768	0.907419
26	350	0.9527548	0.97149964	0.851768	0.907664
27	360	0.9527548	0.97150027	0.851768	0.907666
28	370	0.9528237	0.97150728	0.85202	0.907812
29	380	0.9527548	0.9712237	0.85202	0.907689
30	390	0.952686	0.97148548	0.851515	0.907518
31	400	0.9527548	0.97149365	0.851768	0.907664
32	410	0.9525482	0.97120127	0.851263	0.907249
33	420	0.9528926	0.97178059	0.85202	0.907932
34	430	0.9524793	0.97119383	0.85101	0.907102
35	440	0.9524793	0.97119383	0.85101	0.907102
36	450	0.9525482	0.97120562	0.851263	0.907252
37	460	0.9526171	0.97093785	0.851768	0.907423
38	470	0.9528237	0.97123535	0.852273	0.907838
39	480	0.9526171	0.97121291	0.851515	0.907397
40	490	0.9525482	0.97147455	0.85101	0.907225
41	500	0.9525482	0.97147455	0.85101	0.907225

3. The local optimal maximal depth is found to be 29 within the search interval of [25-100], with an increment of 2, when the number of trees is set to 250. No change in the F-score is noticed after this iteration.
4. The local optimal subset ratio of 0.95 is found using the optimize parameter operator within the search interval of [0.05-0.95], with an increment of 0.05, when the number of trees is set to 250 and the maximal depth is set to 29. This resulted in a slight increase in the F-score to 90.84%.
5. The local optimal confidence hyperparameter of 8.00E-07 is found through a grid search within the interval of [1.00E-07 – 0.5]. This value achieved the highest F-score value within the interval as illustrated in Table A 8. This is done when the number of trees is set to 250, maximal depth is set to 29, and subset ratio is set to 0.95, the hyperparameters' values resulted from the previous iterations. This led to an increase in the F-score to 90.86%.
6. The previous steps are repeated until the F-score stops improving for consecutive iterations. After 8 iterations, the optimal hyperparameters are determined to be the number of trees = 380, maximal depth = 27, subset ratio = 0.8, and confidence = 2.00E-07. The model with these hyperparameters achieved an F-score of 90.94% and an accuracy of 95.34%. At each iteration of the optimization process, the F-score is used to evaluate the performance of the model, and if the performance is improved, the previous iterations of the optimization process are repeated multiple times until the final hyperparameters of the model are found.
7. The search for near optimal hyperparameters is terminated at iteration 9 as the F-score decreased, indicating that the optimization process had reached its limit.

Table 6 provides a summary of the hyperparameters optimization process for the RF model. After applying the optimization process for 9 iterations the final confusion matrix is shown in Table 7.

Table 6 Summary of Hyperparameters Optimization for RF Model.

Hyperparameter		Value	F-score
Iteration 1	Number of Decision Trees	100	90.61
		200	90.72
		250	90.75

		400	90.77
		500	90.72
Iteration 2	Maximal Depth	25	90.820
		29	90.823
		40	90.823
		70	90.823
		100	90.823
Iteration 3	Subset Ratio	0.05	90.42
		0.2	90.72
		0.4	90.70
		0.95	90.84
		1.0	90.83
Iteration 4	Confidence	1.00E-07	90.82
		8.00E-07	90.86
		4.00E-04	90.83
		0.006	90.84
		0.5	90.77
Iteration 5	Number of Decision Trees	100	90.74
		200	90.80
		380	90.86
		450	90.83
		500	90.79
Iteration 6	Maximal Depth	25	90.85
		27	90.87
		55	90.85
		85	90.84
		100	90.85
Iteration 7	Subset Ratio	0.05	90.42
		0.35	90.71
		0.5	90.78
		0.8	90.92
		1	90.86
Iteration 8	Confidence	1.00E-07	90.89
		2.00E-07	90.94

		8.00E-05	90.91
		0.006	90.92
		0.5	90.83

Table 7 Cross-Validation confusion matrix of the optimized RF model.

	True normal	true incident	class precision
pred. normal	10453	569	94.84%
pred. incident	107	3391	96.94%
class recall	98.99%	85.63%	
Accuracy	95.34%	F-score	90.94

The confusion matrix of the testing phase of the optimized RF model is shown in Table 8.

Table 8 Testing confusion matrix of the optimized RF model.

	True normal	true incident	class precision
pred. normal	2267	80	96.59%
pred. incident	53	1080	95.32%
class recall	97.72%	93.10%	
Accuracy	96.18%	F-score	94.20%

The F-score results of the evaluation steps within each iteration of the grid search for the optimal hyperparameters and the resulting confusion matrices of the Cross-Validation for each iteration are provided in Appendix A. In the aftermath of presenting the confusion matrices of the Cross-Validation and testing processes in Table 7 and Table 8, a comprehensive analysis and discussion of these results are be provided in the next sections.

4.1.2. Results of Cross-Validation of random forest.

In this section, the details of incident scenarios in the Cross-Validation dataset are presented. The calculations of DR, FAR, and MTTD of this dataset are illustrated. The confusion matrices presented in Table 7 and Table 8 from the Cross-Validation and testing phases require further analysis to calculate three performance measures: DR, FAR, and MTTD, which are used to evaluate the performance of the developed RF model. During both the Cross-Validation and testing phases, the model's predictions are compared against the actual status of the traffic for each interval in both datasets. Based on this comparison, the model's prediction is classified as true, false alarm, or missed. A sample of this comparison during one of the scenarios used in the Cross-Validation dataset is shown in Table 9.

Table 9 Sample comparison of model predictions and actual status during Cross-Validation.

Interval	Actual Status	Model's Prediction	Comparison Result
31	incident	normal	Missed
32	incident	incident	True
33-37	incident	normal	Missed
38-39	incident	incident	True
40	incident	normal	Missed
41-42	incident	incident	True
43-45	incident	normal	Missed
46-48	incident	incident	True

The observations made during this analysis are as follows:

1. Fluctuations in the incident alarm are observed after the incident had occurred within the following 40 intervals as illustrated in Table 9 . It is noticed that one or more intervals are labeled as normal by the algorithm, even though they are incidents. This fluctuation is found to be caused by the differences and relative differences in volume, speed, and occupancy between upstream and downstream stations, which are among the input variables used by the model to detect incidents. When an incident occurs, it can cause a significant difference in the traffic parameters between the upstream and downstream stations, which the model can easily detect. However, when the incident remains on the road for several intervals, these differences start to decrease until there is only a slight or no difference in the volume, occupancy, and speed of the upstream and downstream stations. At this point, the model may not be able to detect the

incident and incorrectly label it as a normal condition. Therefore, it is important to note that the detection is counted at the first alarm produced by the model, as once the operators receive an alarm, they will check the cameras to verify its authenticity.

2. It is observed that some false alarms occurred at consecutive intervals. Table 10 presents a sample of consecutive false alarms that occurred in one of the scenarios.

Table 10 Sample of consecutive false alarms in a scenario.

Interval	Actual Status	Model's Prediction	Comparison Result
20-26	normal	normal	True
27-30	normal	incident	False Alarm
31-36	incident	incident	True

Treating these consecutive alarms as separate incident alarms is unrealistic and affect the reliability of the model if the traffic operator has to check the road to validate the existence of an incident for these consecutive intervals and finds that there are false alarms. An assumption regarding false alarms has been made in this dissertation to consider real-life conditions when calculating the FAR, where consecutive false alarms are considered as a single false alarm if they are continued for four or fewer intervals (i.e., within two minutes or less). However, if there is a gap of more than four intervals between two false alarms, they are treated as separate false alarms. This assumption considers the real-life situation and assists in the performance of persistence tests and reduction of false alarms. Upon receipt of an incident alarm by a traffic operator, its authenticity is verified. Appropriate action, such as calling an emergency team, is taken if the alarm is found to be true. On the other hand, if the alarm is false and persists for two minutes or less, it is ignored by the operator. If the false alarm disappears and another alarm occurs after two or more minutes, the operator will check the cameras on the road to determine its authenticity. This assumption better represents actual operational conditions and assists in improving the performance of persistence tests and reduction of false alarms. However, it is important to note that this assumption can affect the detection time of an incident, as an incident may occur in the two minutes that the alarms were ignored. Therefore, an assumption of four intervals was made to avoid extending the MTTD of incidents that may occur after a false alarm.

Based on these observations, it can be concluded that simply calculating the DR and FAR from the confusion matrices alone can be inaccurate or unrealistic, as this approach does not take into account the fluctuations in incident alarms or the assumption regarding false alarms. Therefore, further analysis of the model’s results is necessary to obtain realistic DR, MTTD, and FAR values. This analysis focuses on calculating the Time to Detect (TTD) for each incident scenario. To calculate TTD, the system records the first interval in which the incident is detected, with the actual incident occurrence at interval number 31. It is important to note that the detection is counted at the first alarm produced by the model, as once the operators receive an alarm, they will check the cameras to verify its authenticity. Next, the DR of the model is calculated by evaluating how many incidents are successfully detected by the model.

In addition, the number of false alarms for each incident scenario and normal condition scenario is counted using the assumption that was discussed previously. As mentioned before in Section 2.1, there are two methods to calculating the FAR either by using Equation (6) or by using Equation (7). In this dissertation, FAR is calculated using the second method Equation (7), which divides the number of false alarms by the total number of applications of the model. Because it provides a more accurate representation of the performance of the incident detection system. In the first method, the FAR is calculated as the ratio of the number of false alarms generated by the model to the total number of alarms generated by the model. However, this method does not take into account the number of times the model was applied. Therefore, it can lead to an overestimation of the performance of the system if the number of incidents in the data is low. In contrast, the second method of calculating FAR takes into account the total number of times the model is applied. This provides a more accurate measure of the system’s performance because it reflects the proportion of times the model reported a false incident during the monitoring period. Also, it considers the correct classification of normal conditions as a success of the applied model. Therefore, the second method is considered more appropriate for incident detection systems, where the system's performance is measured by its ability to accurately detect the traffic conditions (normal or incident). Therefore, the total number of false alarms will be divided by the total number of intervals as mentioned in Equation (7) to calculate the FAR. By analysing these results, a more comprehensive understanding of the model's performance can be obtained.

The data shown in Table 11 illustrates a set of intervals that were misclassified within one hour during an incident scenario in the Cross-Validation dataset. Table 11 displays the actual traffic status during an incident that took place 250 meters away from the upstream station, with the distance between the upstream and downstream detector stations being 1000 meters. The incident occurred at 0.6 D/C ratio, and Table 11 presents the model’s predictions for each interval, along with a comparison between the actual status and the predictions. It is noteworthy that the model was able to identify the incident after one interval, with a TTD of 1 interval or 30 seconds, as it did not detect the incident at interval 31. Among the intervals that the model classified as an incident, two were falsely classified, and their actual status was normal. These two false alarms were not consecutive, so they were considered as two false alarms in an hour.

Table 11 Sample calculation of DR, MTTD, and false alarms for Cross-Validation dataset.

Intervals	Actual Status	Model’s Prediction	Comparison Result
9	normal	incident	False Alarm
31	incident	normal	Missed
32	incident	incident	True
71	normal	incident	False Alarm

The Cross-Validation dataset utilized in this analysis consists of 121 scenarios, comprising 22 normal conditions and 99 incident scenarios. The 22 normal scenarios in the dataset contain a total of 2640 intervals, while the 99 incident scenarios contain 11880 intervals. In Table 12, the number of false alarms for both sets of scenarios is presented, alongside the number of detected incidents and the total TTD required to identify these incidents.

Table 12 Performance comparison of normal and incident scenario sets for RF model results from Cross-Validation dataset.

Scenario Set	Detected Incidents	Number of False Alarms	MTTD (minutes)
Normal	N/A	10	N/A
Incident	96	80	1.05

As shown in Table 12, out of the 99 incident cases included in the dataset, the model is unable to detect 3 incidents. This gives a DR of 96.97%. The Cross-Validation dataset consists of 121 hours of traffic data, with traffic state checked every 30 seconds, resulting in 14,520 model's applications. During these applications, the model generated a total of 90 false alarms. This resulted in a FAR of 0.62%. Therefore, the model's performance during Cross-Validation can be reported as DR = 96.97%, MTTD = 1.05 minutes, and FAR = 0.62%. In order to provide a more comprehensive analysis of the model's performance in detecting incidents of varying severity levels, Table 13 summarizes its DR, FAR, and MTTD for each level of incident severity.

Table 13 Performance of RF Model by Incident Severity Level.

Incident severity	DR(%)	FAR (%)	MTTD (minutes)
1 lane	88.46	0.641	2.5
2 lanes	100	0.833	1.25
3lanes	100	0.613	0.588
4 lanes	100	0.972	0.5
5 lanes	100	0.699	0.565

Next, a sensitivity analysis is conducted to assess the impacts of four factors, namely D/C ratio, incident severity, location of the incident, and distance between detectors, on DR, FAR, and MTTD. Each factor is analysed individually to evaluate its effect on the performance metrics. It should be noted that the 0.4 and 0.5 D/C ratios use different blockage levels (2 or 4 lanes) and have constant incident location relative to the upstream detector. Also, the 0.9 and 1.1 D/C ratios are applied for different spacing between the detectors, different severity of the incident (2, 3, or 4 lane blockage), and different location of the incident. Therefore, these D/C ratios (0.4, 0.5, 0.9, and 1.1) are not considered in the sensitivity analysis. To ensure a fair comparison, incident scenarios with the same severity levels, locations, and detector spacings are selected for the analysis. Specifically, incident scenarios with D/C ratios of 0.6, 0.8, 1.0, and 1.2 are chosen for the analysis.

4.1.2.1. *Impacts of D/C ratio on performance metrics*

To investigate the impact of D/C ratio on the performance metrics of the incident detection system. The variation in DR, FAR, and MTTD, when the D/C ratio is varied between four ratios (0.6, 0.8, 1.0 and 1.2) is analysed. Figure 23 depicts the relation between the DR and the D/C ratio for the RF model during Cross-Validation phase.

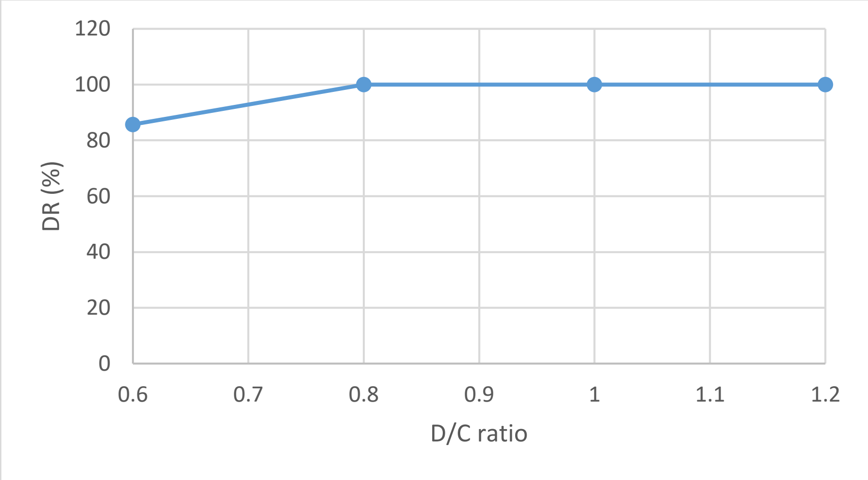

Figure 23 DR vs. D/C ratio for RF model during Cross-Validation.

Figure 23 shows that the DR reached 100% for all D/C ratios, except for 0.6, where it reached 85.71%. The low DR at D/C of 0.6 can be attributed to the fact that three incidents out of the total are not detected by the model. These three incidents occurred at low traffic flow, and one lane blockage out of the seven lanes of the road. Therefore, it is believed that they are minor incidents that did not have significant effect on the traffic performance and were harder to detect, which resulted in a reduced DR of the model.

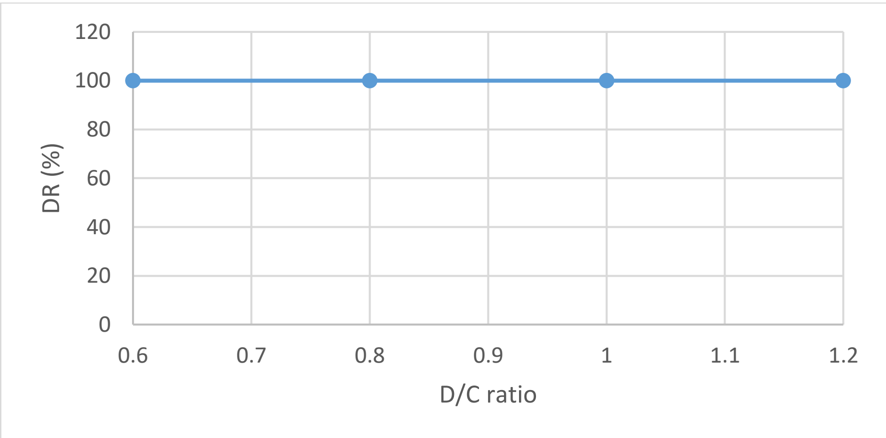

Figure 24 DR vs. D/C ratio for RF model during Cross-Validation excluding minor incidents during low traffic volume.

To confirm this justification for low DR at D/C ratio of 0.6, Figure 24 shows the relationship between DR and D/C ratio without the one lane blockage incidents that occurred at 0.6 D/C ratio. As shown in Figure 24, the DR is 100% for all D/C ratios after excluding the minor incidents during low traffic volume. This suggests that the model's performance can be influenced by the occurrence of minor incidents during low traffic volume, and excluding such incidents can lead to a higher DR. Therefore, it is important to take into account the severity of incidents and the volume of traffic at the time of their occurrence, when assessing the performance of the model in detecting incidents. The relationship between the FAR and the D/C ratio is shown in Figure 25.

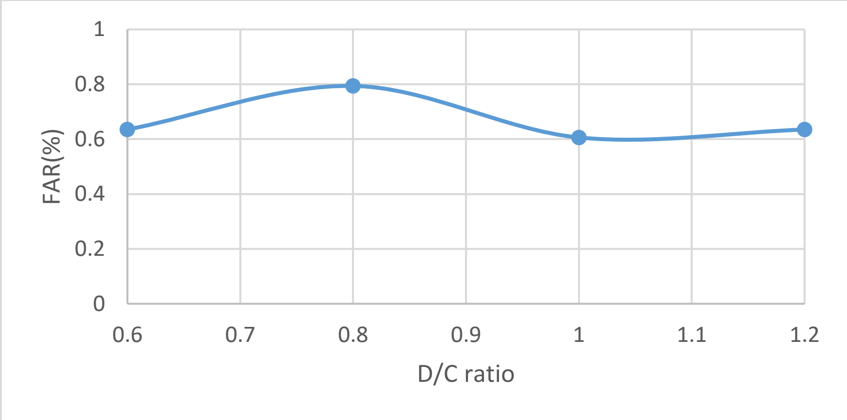

Figure 25 FAR vs. D/C ratio for RF model during Cross-Validation.

Figure 25 shows that there is a variation in FAR with respect to D/C ratio. While the FAR values show some fluctuations, there is an overall trend of decreasing FAR with an increase in D/C ratio from 0.8 to 1.2. To further understand the impact of minor incidents during low traffic volume on the FAR, Figure 26 shows the relationship between FAR and D/C ratio excluding the one lane blockage incidents that occurred at 0.6 D/C ratio.

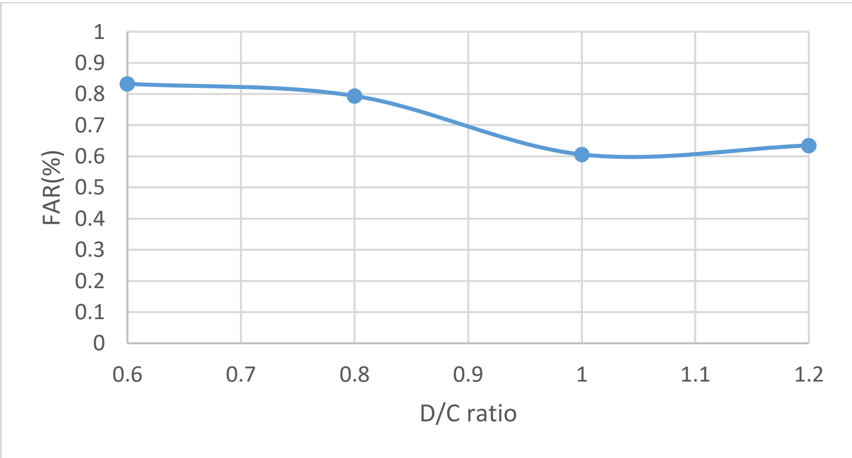

Figure 26 FAR vs. D/C ratio for RF model during Cross-Validation excluding minor incidents during low traffic volume.

After excluding the minor incidents during low traffic volume, it becomes evident that the FAR decreases with the increase in D/C ratio. Specifically, the FAR remains relatively stable for D/C ratios of 1.0 and 1.2, with values of 0.606% and 0.635%, respectively. The decreasing trend of FAR with an increase in D/C ratio can be attributed to the fact that as the congestion level increases, vehicles are more likely to be closely spaced and moving slowly, which can result in more consistent behavior and patterns that are easier for the incident detection system to recognize. In contrast, in case of lower traffic volumes, vehicles may be more widely spaced and traveling at varying speeds, making it more difficult for the system to distinguish between normal and abnormal traffic behavior. This is consistent with the literature review's earlier finding that suggested that the flow of the vehicles during low traffic conditions can be low, leading to low occupancy values and a higher chance of false alarms [70], [94].

Figure 27 shows that the MTTD values appear to increase as the D/C ratio increases from 0.8 to 1.2. This suggests that the model is more efficient at detecting incidents when the D/C ratio is low. One possible explanation for the increase in MTTD is that at higher traffic volume and congestion, there are queues at the blocked sections of the road, which can cause delays in traffic flow. As a result, incidents that occur may take longer to have a detectable impact on the traffic performance, as the travel time between the detectors increases with congestion. Therefore, the detection of incidents can be further delayed.

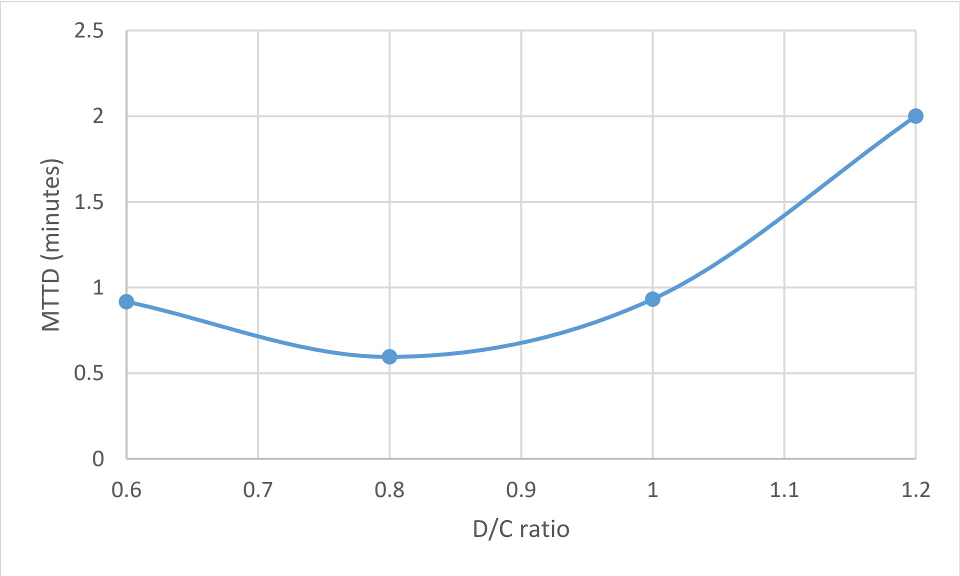

Figure 27 MTTD vs. D/C ratio for RF model during Cross-Validation.

Figure 28 shows the relationship between MTTD and D/C ratio without the one lane blockage incidents that occurred at 0.6 D/C ratio.

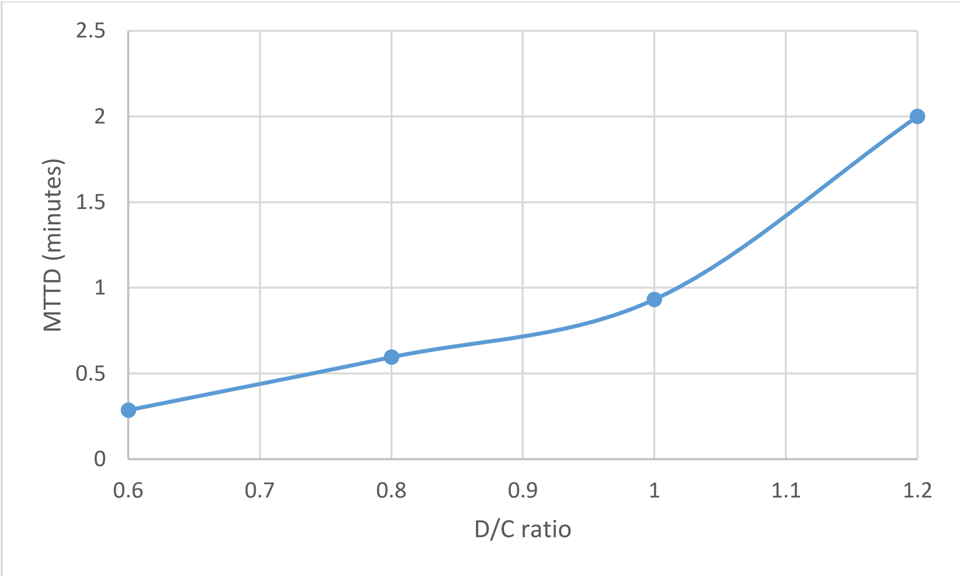

Figure 28 MTTD vs. D/C ratio for RF model during Cross-Validation excluding minor incidents during low traffic volume.

Figure 28 confirms that the inconsistent trend in Figure 27 is mainly because of the existence of minor incidents during low traffic volume as shown in Figure 28, the MTTD consistently increases with the increase in D/C ratio. It should be noted that Figure 23, Figure 25, and Figure 27 included incidents with one lane blockage occurring at a D/C ratio of 0.6, which the model failed to detect some of them due to their minor impact during low traffic volume. As a result, the trends of DR, FAR, and MTTD in these figures were not consistent. On the other hand, Figure 24, Figure 26 and Figure 28 show a clearer trend for the DR, FAR, and MTTD, after the exclusion of one lane blockage incidents that occurred at 0.6 D/C ratio. Therefore, these incident scenarios will be excluded in the upcoming analysis of the other factors to obtain a clear picture of the impact of the factor of interest on the performance measures. The results provided in Figure 24, Figure 26 and Figure 28 show that the DR remains 100% for all D/C ratios, but the FAR and MTTD are highly dependent on the D/C ratio. The FAR decreases as the D/C increases. Meanwhile, the MTTD increases as the D/C ratio increases, with a sharp increase at D/C ratio 1.2. These findings suggest that D/C ratio is an important factor to consider when evaluating and improving the performance of traffic incident detection models.

4.1.2.2. *Impacts of incident severity on performance metrics*

An analysis is conducted to examine the variation in DR, FAR, and MTTD as the lane blockage varied from 1 lane to 5 lanes blockage. The DR of the incident detection system shows a constant value of 100%, for all incident cases except for the one lane blockage at D/C ratio of 0.6, indicating that the system is highly accurate in detecting incidents with a significant impact on traffic flow. However, when considering incidents with one lane blockage at a D/C ratio of 0.6, the DR drops to 88.46%. This result is in line with the findings of previous research [54], [143], indicating that the DR tends to be lower for minor severity incidents that occur during periods of low traffic volume.

Figure 29, shows a trend where the FAR remains lower than 1% for all lane blockage categories.

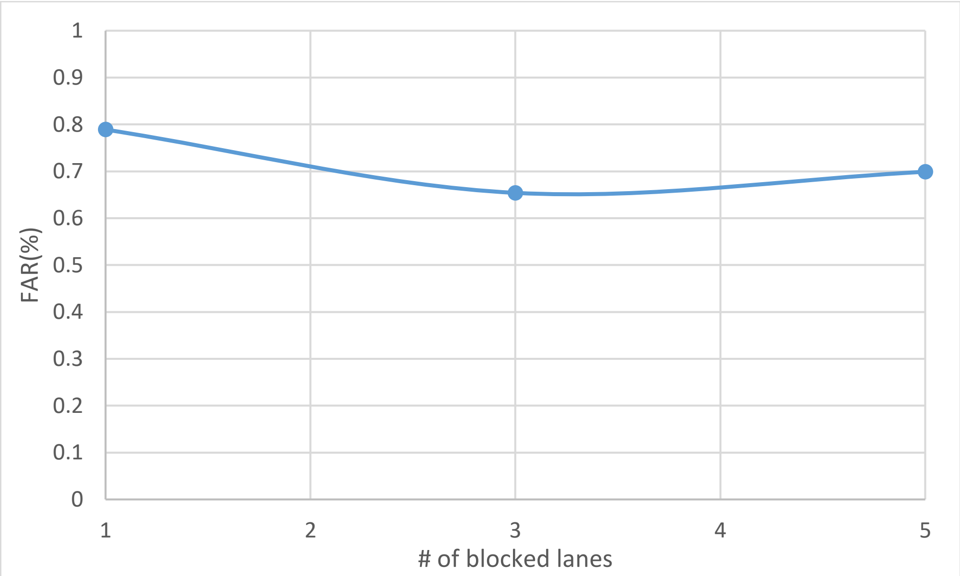

Figure 29 FAR vs. lane blockage for RF model during Cross-Validation.

It can be observed that the FAR is slightly affected by the number of lanes blocked. The results indicate that the number of lanes blocked has a minor impact on FAR as there is a very small variation. Specifically, the FAR for one lane blocked scenarios was found to be 0.789%, while the FAR for three and five lanes blocked scenarios were 0.654% and 0.699%, respectively. These results suggest that the incident detection system is not significantly impacted by the incident severity, except for minor incidents during low traffic volume. The lower FAR values across all lane blockage categories are a good indicator of the efficacy of the detection model in distinguishing actual incidents from normal traffic conditions. It should be noted that when the incidents with one lane blockage at D/C of 0.6 are included, the FAR is 0.64, which is lower than the case without such incidents. The system did not create any false alarms nor detect any incidents during such scenarios. Therefore, when these scenarios were excluded, the FAR increased as the number of false alarms is compared to a smaller number of application intervals.

According to Figure 30, it can be observed that the MTTD decreases as the number of lanes blocked increases. Specifically, the MTTD for one lane blockage is the highest at 2.37 minutes, while the MTTD reduces to 0.589 and 0.565 minutes for three and five lanes blockage, respectively. This could be due to the fact that incidents with more severe lane blockage are easier to detect as they have a great impact on the traffic flow, allowing the system to detect such incidents in a shorter time. The current findings are in agreement with prior study of Cheu and Ritchie [52].

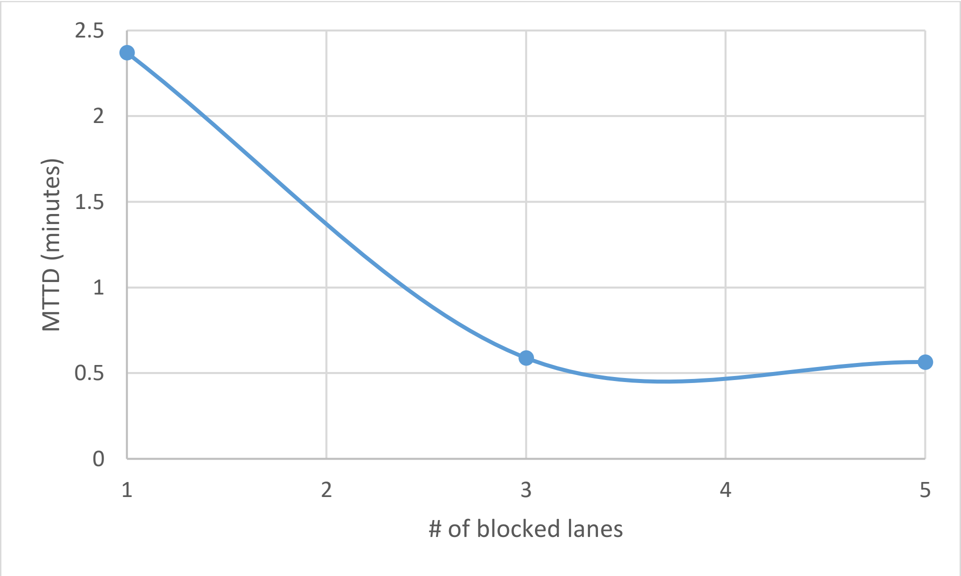

Figure 30 MTTD vs. lane blockage for RF model during Cross-Validation.

These results suggest that the severity of an incident, as represented by the number of blocked lanes, has a significant impact on the MTTD. However, it has no impact on the detectability of the incidents and a minor impact on FAR. The DR of the system is consistently high for all lane blockage, except for one lane blockage at D/C of 0.6. The FAR was marginally affected as the number of lanes blocked changes. The MTTD of the system decreases as the severity of the incident increases, indicating that the system is faster at detecting severe incidents with higher lane blockage. By underscoring the impact of incident severity on the efficacy of detection systems, these findings emphasize the need to incorporate such factors into the design and assessment of these systems.

4.1.2.3. Impacts of incident location on performance metrics

The incident location varies to be 0.25, 0.5, and 0.75 of the distance between the detectors, from the upstream detector. These values are considered to examine their effect on DR, FAR, and MTTD. The DR remained 100% for all three incident locations, indicating the system's high accuracy in detecting severe incidents that involve lane blockages and have a significant impact on traffic flow. It should be noted that this analysis excludes one lane blockage incidents at 0.6 D/C ratio, as these incidents were found to be less detectable due to their minor influence during low traffic volume, as mentioned previously. It is worth noting that when the incidents with one lane blockage at D/C of 0.6 are included, the DR values are 93.1%, 94.12% and 100% for 0.25, 0.5 and 0.75 incident locations respectively.

Figure 31 depicts the relation between the FAR and the incident’s location.

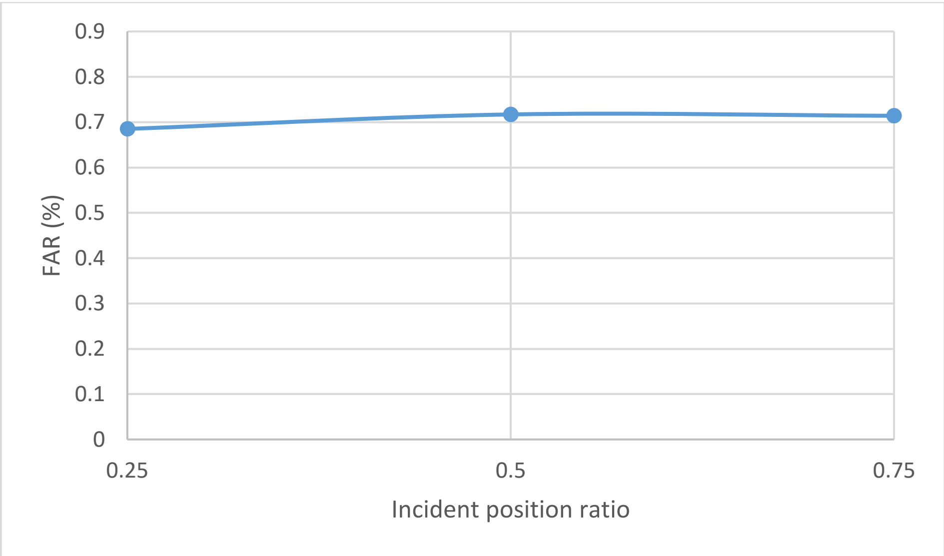

Figure 31 FAR vs. incident's location for RF model during Cross-Validation.

Figure 31 shows that the FAR is almost constant as the incident moves further from the upstream detector station, with values of 0.685, 0.717, and 0.714 observed for incident position ratios of 0.25, 0.5, and 0.75, respectively. This result indicates that the incident location has almost no impact on the FAR. With the inclusion of incidents that involve a single lane blockage at D/C of 0.6, the FAR values for incident locations of 0.25, 0.5, and 0.75 are 0.92%, 1.08%, and 0.37%, respectively.

Figure 32 illustrates the relationship between the MTTD and the incident's location.

Figure 32 MTTD vs. incident's location for RF model during Cross-Validation.

According to Figure 32, the MTTD values increase as the incident moves further from the upstream detector station. This can be attributed to the fact that as the incident moves further away, the time it takes for its effects to propagate to the upstream detector increases. As a result, the detection of the incident is delayed, leading to higher MTTD values. The impact of varying the incident's location, relative to the upstream station detector, on the DR, FAR, and MTTD of the incident detection system is investigated in Figure 31 and Figure 32. Based on the analysis of the DR, FAR, and MTTD of the incident detection system, it can be concluded that the incident location has a significant impact on the MTTD. The system is highly effective in detecting incidents at all locations, with a DR of 100%. The FAR is slightly impacted by the incident location, while the MTTD of the system is consistently increasing, as the incident location moves further away from the upstream detector. This trend indicates that the incident location relative to the detector plays a vital role in the MTTD, with incidents occurring closer to the upstream detector being detected faster. By demonstrating the impact of incident location on the performance of detection systems, these results highlight the need to take into account the spatial distribution of incidents when designing and evaluating such systems. Neglecting incident location as a factor can compromise the efficacy of detection algorithms and undermine their overall effectiveness.

4.1.2.4. Impacts of distance between detectors on performance metrics

The distance between the two detectors varies from 500 to 1500 meters, and the DR, FAR, and MTTD are evaluated to assess the impact of this variation on the model's performance. The results show that DR is constant at 100% for all three spacing values, 500m, 1000m, and 1500m excluding minor one lane blockage incidents at 0.6 D/C ratio. The inclusion of such incidents leads to a reduction in DR values to 96.97% and 92.86% for detector distances of 1000m and 1500m, respectively. Notably, of the three minor incidents at 0.6 D/C ratio that were not detected, one occurred at 1000m detector spacing and the remaining two occurred at 1500m spacing.

The trend in FAR values, presented in Figure 33, indicates that the FAR increases as the distance between detectors increases. This trend can be attributed to the increased likelihood of detecting false incidents due to the longer travel time of vehicles between the detectors.

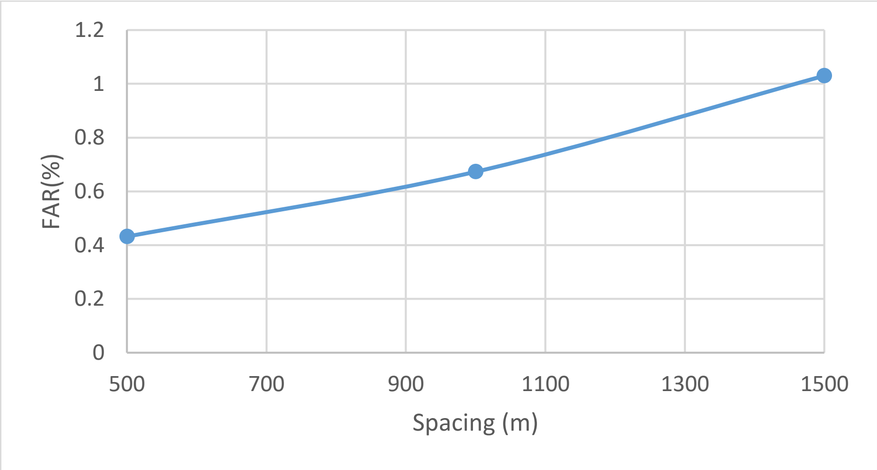

Figure 33 FAR vs. detectors' spacing for RF model during Cross-Validation.

For instance, the 1500-meter spacing achieved a DR of 100%, but it also has a relatively higher FAR value compared to smaller spacings. The increase of the FAR with the increase of distance between detectors may be due to the longer travel time of vehicles between detectors. Longer travel time can cause fluctuations in traffic measurements, such as volume, speed, and occupancy, at both the upstream and downstream detectors. An increase in travel time can result in a higher volume of vehicles being detected at the upstream detector and a lower volume at the downstream detector. Additionally, the downstream detector may record a lower speed than the upstream detector due to the longer travel time, leading to fluctuations in the speed measurement. Lastly, the occupancy measurement may be affected if the travel time between detectors increases. The longer travel time can cause vehicles to occupy the detector for a longer period, leading to fluctuations in occupancy measurement between the two detectors. These fluctuations in traffic measurements can cause false triggers of incident alarms.

Figure 34 shows the association between the FAR and detectors' spacing.

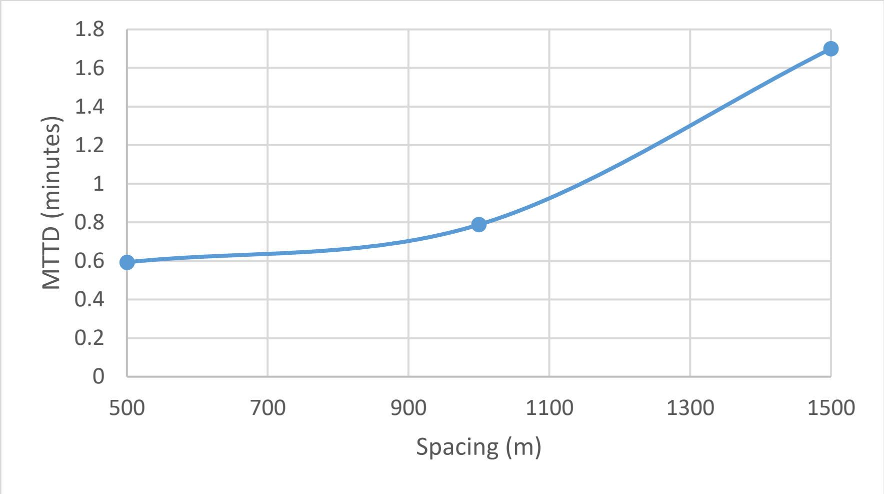

Figure 34 MTTD vs. detectors' spacing for RF model during Cross-Validation.

The trend in MTTD values shown in Figure 34 illustrates that larger detector spacings are associated with longer detection times. This can be attributed to the increased travel time of vehicles between the detectors, as mentioned previously, which can cause delays in detecting incidents. This is align with the findings of Rossi et al. [143] who reported that an increase in the distance between detectors results in higher MTTD values. After conducting an analysis of the impact of the distance between the upstream and downstream detectors on the performance measures of an incident detection system, it can be concluded that smaller detector spacings result in better performance in terms of FAR and MTTD. This is due to the closer proximity of the detectors, which reduces the detection time and traffic measurement fluctuations between the detectors. The DR remains 100% in all cases, indicating that the system is capable of detecting all incidents regardless of the distance between the detectors. In contrast, larger detector spacings are associated with longer detection times and higher FAR values. This is because vehicles take longer to travel between the detectors, leading to more fluctuations in traffic measurements, which can be falsely detected as incidents. Larger detector spacings also result in higher MTTD values due to the increased travel time between the detectors. However, while smaller detector spacings are more effective in detecting incidents, they may not be practical due to installation and maintenance costs. On the other hand, larger detector spacings may be more cost-effective and easier to maintain but may result in higher FAR and MTTD values, which is consistent with the findings of Karatsoli et al. [73]. They reported that the distance between detectors and traffic volumes are positively related to the FAR and MTTD, meaning that as the distance between detectors increases, the FAR and MTTD values increase and vice versa. Therefore, the decision to select an appropriate detector spacing should consider specific application requirements, available resources, and trade-offs between the performance (in terms of DR, FAR, and MTTD) and the associated costs. An optimal balance between these factors is necessary to ensure an effective and cost-efficient incident detection system.

The impacts of each of the four factors on the performance measure have been thoroughly investigated, as presented and discussed in detail throughout this section. The DR, TTD, and number of false alarms of each scenario in the Cross-Validation dataset are calculated, and the results are tabulated in Appendix B-Table B 1.

4.1.3. Results of testing phase of random forest.

In this section the DR, MTTD, and FAR of the testing dataset are calculated using the same method as in the previous section. A sample of these calculations is shown in Table 14, which depicts the actual traffic status during a one -lane blockage incident that occurred 250 meters from the upstream detector station, with the distance between the upstream and downstream detector stations being 500 meters. The incident occurred at 0.8 D/C ratio, and the table displays the model's predictions of each interval along with a comparison between the actual status and the predictions.

Table 14 Sample calculation of DR, MTTD, and false alarms for testing dataset.

intervals	Actual Status	Model's Prediction	Comparison Result
25-30	normal	normal	True
31	incident	normal	Missed
32	incident	normal	Missed
33	incident	normal	Missed
34-70	incident	incident	True
71	normal	incident	False Alarm
72	normal	normal	False Alarm
73-80	normal	normal	True

As observed from Table 14, the model is able to detect the incident 3 intervals after it occurred, with a TTD of 3 interval or 1.5 minutes, as the incident is detected at interval 34. Additionally, two false alarms were generated at intervals 71, and 72. As the false alarms at intervals 71 and 72 occur at consecutive intervals, they were counted as only 1 false alarm in an hour. In Table 15 , the DR, TTD, and number of false alarms for each scenario in the testing dataset is presented.

Table 15 Performance of incident scenario sets for RF model results from testing dataset.

Number of blocked lanes	Incident location	Spacing	D/C ratio	Detection Status	TTD (intervals)	TTD (minutes)	False alarms
3	250	500	0.5	Detected	0	0	1
3	250	1000	0.5	Detected	1	0.5	2
1	125	500	0.6	Detected	0	0	0
3	250	500	0.6	Detected	0	0	0
3	250	1000	0.6	Detected	1	0.5	1
1	500	1000	0.6	Detected	1	0.5	1
5	750	1000	0.6	Detected	0	0	0
5	750	1500	0.6	Detected	4	2	2
1	250	500	0.8	Detected	3	1.5	1
1	250	1000	0.8	Detected	3	1.5	1
3	375	500	0.8	Detected	2	1	1
3	375	1500	0.8	Detected	2	1	4
5	500	1000	0.8	Detected	1	0.5	1
3	1125	1500	0.8	Detected	1	0.5	0
4	375	750	0.9	Detected	1	0.5	2
2	625	1250	0.9	Detected	1	0.5	2
1	125	500	1	Detected	1	0.5	0
1	500	1000	1	Detected	2	1	1
3	750	1000	1	Detected	1	0.5	1
3	750	1500	1	Detected	1	0.5	1
5	1125	1500	1	Detected	0	0	1
4	375	750	1.1	Detected	3	1.5	3
2	625	1250	1.1	Detected	0	0	0
5	125	500	1.2	Detected	30	15	0
1	375	500	1.2	Detected	0	0	0
1	375	1500	1.2	Detected	1	0.5	0
1	750	1000	1.2	Detected	1	0.5	2
1	750	1500	1.2	Detected	1	0.5	0
3	1125	1500	1.2	Detected	6	3	1
					2.3	1.17	29

As shown in Table 15, the testing dataset includes 29 incidents, all of which are correctly detected by the model which resulted in a DR of 100. It should be noted that the testing datasets contain two incidents that blocked one lane and occurred at 0.6 D/C ratio. During the Cross-Validation phase, the model was unable to detect three out of seven incidents with one lane blockage that occurred during 0.6 D/C ratio. However, during the testing phase, the model was able to detect both incidents with one lane blockage that occurred at the same traffic volume as the incidents in the Cross-Validation phase. This could be due to the fact that the model learned from its performance during the Cross-Validation phase and made improvements. Additionally, the location and spacing of the detectors used during the Cross-Validation phase may have contributed to the model's inability to detect these incidents. Therefore, the location and spacing of the detectors used during the Cross-Validation phase, as well as the learning and improvements made by the model, may have contributed to the differences in the model's performance between the Cross-Validation and testing phases.

The MTTD for these 29 incidents is determined to be 1.17 minutes. During the testing phase, traffic measurements are collected for 29 hours, resulting in the application of the model 3480 times. Within these 3480 intervals, the model produced 29 false alarms, FAR of 0.862%.

4.1.4. Discussion of random forest model results

The results of the Cross-Validation and testing phases indicate a high level of accuracy in detecting incidents with a low FAR. In the testing phase, all 29 incidents are correctly detected, resulting in a DR of 100%. The DR in the Cross-Validation phase is found to be 96.97%. The size and complexity of the dataset in each phase may have contributed to this difference, with the testing phase having a smaller dataset and potentially fewer complex incidents. The MTTD in the testing phase is 1.17 minutes, whereas it is 1.05 minutes in the Cross-Validation phase. The FAR is found to be 0.862% in the testing phase and 0.62% in the Cross-Validation phase. This aligns with previous literature, which suggests that the FAR increases with the increase in DR [52], [54]. Despite slight variations in results between the Cross-Validation and testing phases, both show a high level of accuracy in detecting incidents and a low FAR. These findings indicate that the algorithm has a high level of performance and reliability in detecting incidents. Overall, based on investigating the impact of each factor on the performance measures, it can be concluded that the limitations and effectiveness of the incident detection system, as observed in the analysis of DR, FAR, and MTTD, can be attributed to a combination of factors such as traffic volume, incident severity, distance between the detectors, and distance from the detector station. These factors impact the system's ability to detect and respond to incidents accurately and quickly.

After analysing the impact of each of the four factors individually, the impacts of all four factors together on the detectability, TTD, and FAR of incidents in the Cross-Validation and testing phases is discussed. A selection of incident scenarios that are not detected or have relatively long TTD are analyzed to understand the combined impact of all four factors in the Cross-Validation and testing phases. During the Cross-Validation phase, it was previously noted that the model failed to detect three incidents that blocked one lane and occurred during low traffic volume at a D/C ratio of 0.6. These incidents have taken place at a detector spacing of 1000 and 1500 meters respectively. The model's inability to detect these incidents can be attributed to three key factors: traffic volume, incident severity, and detector spacing, which have been discussed earlier. The low severity of these incidents, combined with their occurrence during low traffic volume and the significant distance between the detectors, all contributed to the model's inability to detect them. The model was able to detect an incident where only one lane was blocked, and the D/C is 0.6. The incident occurred 750 meters from the upstream station with a spacing of 1000 meters between the stations. However, the model took 9.5 minutes to detect the incident, which is not a good indicator of the detection performance. This delay in detection can be attributed to the low severity of the incident, the low D/C ratio, and the distance between the detectors. These factors affected the model's ability to detect the incident in a timely manner. This finding supports the earlier conclusion that minor severity incidents during low traffic volume can be hard to detect or undetectable. This observation is consistent with the findings of Cheu and Ritchie [54], Ahmed and Hawas [94] and Rossi et al. [143] who reported that incidents involving one lane blockage during low traffic volume are associated with low DR and high MTTD values. At a D/C ratio of 1.2, two incidents with high TTDs were observed during the Cross-Validation phase. These two incidents are minor one lane blockage incidents located at distances of 125 and 1125 meters from the upstream station, respectively. The distances between the detector stations for these two incidents are 500 and 1500 meters, respectively. The model took 6 and 18 minutes, respectively, to detect them. The extended TTDs of the two incidents during the high D/C ratio of 1.2 can be attributed to a combination of factors. Firstly, the severity of the incidents played a role in the delay of detection. As indicated in Figure 30, minor incidents are associated with higher MTTD. Additionally, the congested conditions during a D/C ratio of 1.2 may have also contributed to the long MTTD, as illustrated in Figure 28. During periods of high demand, such as a D/C ratio of 1.2, the road network experiences heavy traffic congestion. This results in more vehicles on the road, which tend to travel in platoons or sometimes wait in queues, due to the lack of available space. The existence of these platoons can cause slower travel speeds, which in turn can cause a delay in incident detection by the model. Moreover, the presence of platoons makes it more difficult for the model to detect changes in traffic patterns that could indicate an incident. To illustrate this, Figure 35 depicts a section of a road at four-time intervals t_0, t_1, t_2, and t_3. The starting time (t_0) represents the time interval at which the incident occurs, while t_1 to t_3 represent the intervals after the occurrence of the incident. Before t_0, there were already queues of vehicles due to congestion, and there were no significant variations in traffic parameters between the upstream and downstream detectors stations. When an incident occurs, it blocks the vehicles and adds to the upstream queues of blocked vehicles. At the meantime, there are vehicles queued between the incident location and the downstream detector. The queue starts to discharge from t_0 until sometime later t_3. At t_3, the platoon of blocked vehicles is completely discharged, and the vehicles start to spread out. It is only after this time interval that the downstream detector will sense the change in traffic parameters caused by the occurrence of the incident. As congestion increases and the number of vehicles in the platoon increases, the time it takes to clear the platoon also increases. This leads to a longer TTD. Moreover, the distance between the upstream and downstream detectors, which is represented by "D" in Figure 35, is another factor that affects the TTD. The greater the distance between these detectors, the more time it will take to clear the platoon, resulting in a longer TTD. Hence, the difference in TTD between the two incidents can be attributed to the varying distances between the detectors. In the first incident, the detectors are spaced 500 meters apart, while the detectors are spaced 1500 meters apart in the second incident scenario. It is worth noting that these two incidents with higher TTDs contributed to the high MTTD value at 1.2

D/C ratio that was observed in Figure 28. Likewise, during the testing phase, a severe five-lane blockage incident occurred at 1.2 D/C ratio, and it occurred 125 meters from the upstream station, with a spacing of 500 meters between stations. The TTD for this incident is high. The model took 15 minutes to detect it. The TTD for this incident is high, which can be attributed to the oversaturated condition of the road network. As discussed earlier, the two incidents of the Cross-Validation at 1.2 D/C ratio also had a similar impact on the TTD. In this incident, the demand was already above the road's capacity by 20%, and the vehicles on the roads were already queued due to congestion. Therefore, when this severe incident occurred, it further exacerbated the congestion, resulting in a further build-up of queued vehicles. However, the impact of this incident would take some time until the queues of vehicles ahead of the incident location are discharged, so that the impact of this incident reached the downstream station as illustrated in Figure 35. Therefore, the combination of a high D/C ratio, the severity of the incident, the distance between the detectors and the incident location resulted in a long TTD of 15 minutes.

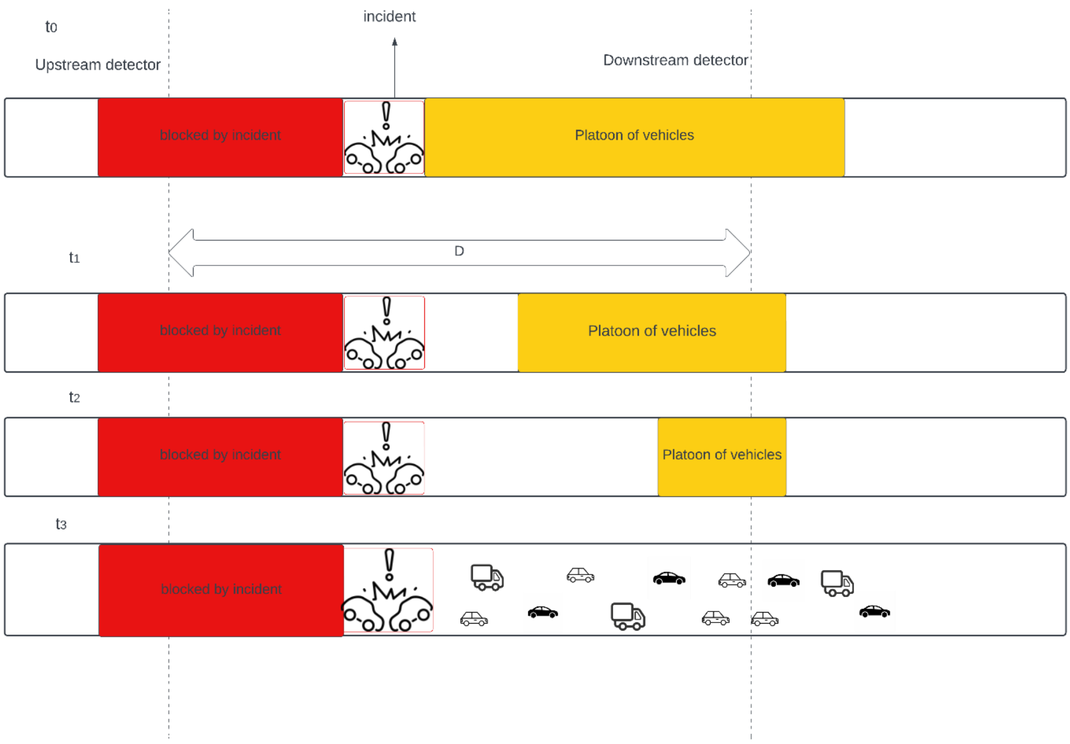

Figure 35 Visualization of incident detection delay during high traffic volume with platooning.

Notably, during the evaluation of the incident detection system, it was observed that false alarms were produced in certain instances. Most of these false alarms were generated in the 1-2 minutes range post-incident termination. This is due to the residual impact of the cleared incident, which can cause fluctuations in traffic performance that may be misidentified as a persistent issue by the model. In such cases, the operator can adopt a wait-and-see approach, allowing for a brief time to assess whether the situation is truly an ongoing incident or just a residual effect. This approach can help mitigate the occurrence of false alarms, ensuring the efficiency and accuracy of the incident detection system.

In conclusion, it is highlighted by the analysis of the Cross-Validation and testing phases that considering various factors such as severity of the incident, traffic conditions, and location and spacing between the stations is crucial. The detectability and MTTD of incidents can be significantly influenced by these factors individually or together, ultimately impacting the accuracy of the system's ability to detect them. Thus, the potential of the incident detection system to be a useful tool for real-world incident detection is demonstrated by the results. The succeeding section discusses the results of the Neural Network model in the same manner.

4.2. Analysis and Discussion of Neural Network Model Results

In this section, the results of the analysis and discussion of the MLF model are presented. Firstly, the hyperparameters of the MLF model are fine-tuned, and the outcomes of this process are reported. Subsequently, the optimized hyperparameters are utilized to perform the Cross-Validation and testing phases, and the findings of these stages are analysed in a similar procedure as the one presented in Section 4.1.

4.2.1. Optimization results of neural network

This section presents the outcomes of the hyperparameter optimization process conducted for the MLF model, aiming to determine the ideal values of hyperparameters that optimize the model's performance metrics. A systematic method is applied, as shown in Figure 22, to explore diverse combinations of hyperparameters and evaluate their influence on the Accuracy, Precision, and Recall of the model. To ensure the optimization outcomes were reliable and not overfitted to the training data, Cross-Validation techniques are employed to assess the impact of the hyperparameters on the performance metrics. The results include the hyperparameters' values at the conclusion of each iteration, as well as the model's confusion matrix. Additionally, the final optimized hyperparameters and the corresponding confusion matrices of the model during both Cross-Validation and testing stages are presented.

1) The number of hidden layers and their size are optimized in the first step of the optimization process. This involved testing different combinations of one and two hidden layers with varying sizes of 5 to 50 neurons. Initially, a single hidden layer was used with varying sizes ranging from 5 to 50 neurons. The sizes were increased by 5 neurons to cover a wide range of possibilities. Once the performance of the model was evaluated with different sizes of a single hidden layer, all possible combinations of two hidden layers were examined. Each hidden layer in the combination had a size ranging from 5 to 50 neurons, incremented by 5 neurons. The F-score is used to determine the near optimal combination, and the network with one hidden layer and 25 neurons achieved the highest F-score of 89.59%, as shown in Table 16.

Table 16 Confusion matrix of iteration 0 for MLF model.

	true normal	true incident	class precision
pred. normal	10411	626	94.33%
pred. incident	149	3334	95.72%
class recall	98.59%	84.19%	
Accuracy	94.66%	F-score	89.59%

2) As shown in Table 17, the optimization process continued with the testing of different learning rates ranging from 0.001 to 0.95 as shown in Table 17. Within this range, the learning rate varied with an increment of 0.001 from 0.001 to 0.009, then with an increment of 0.01 from 0.01 to 0.09, and then with an increment of 0.1 from 0.1 to 0.95. The local optimal learning rate was found to be 0.04, which increased the F-score to 89.83%. Table 17 displays a sample of the values tested for learning rate hyperparameter using the optimize parameter operator. The full range of values tested, from 0.001 to 0.95 is presented in Appendix C-Table C 3.

Table 17 Sample of learning rates used in the grid search.

Evaluation step	Learning rate	Accuracy	Precision	Recall	F-score
2	0.002	0.9386364	0.964021	0.805051	0.877376
4	0.004	0.9417355	0.966232	0.814899	0.884094
6	0.006	0.9443526	0.963048	0.827778	0.890273
8	0.008	0.9460744	0.954971	0.842172	0.89494
11	0.02	0.9473829	0.958196	0.844192	0.897482
13	0.04	0.9480028	0.962131	0.842677	0.898383
15	0.06	0.946281	0.94965	0.848232	0.895957
17	0.08	0.944697	0.938748	0.853535	0.893912

3) After determining the size of the hidden layer and the momentum value in the previous iterations, different momentum values, ranging from 0.001 to 0.95, are tested. The momentum value varied within this range in the same system followed for the learning rate. The near optimal momentum value is found to be 0.9 and the F-score remained the same as previous iteration.
4) To optimize the error epsilon, different values ranging from 1.00E-10 to 0.1 are tested. Within this range, the values were increased by multiplying the previous value by 10. The value of 1.00E-10 resulted in the highest F-score.
5) A range of learning cycle values was tested, from 50 to 1000 in a linear pattern with a constant increment of 17 to yield 55 steps. The highest F-score value of 90.1% was achieved at 1000 training cycles.
6) After iteration 6, the F-score remained the same in the next iterations thus the optimization process was terminated.

The near optimal hyperparameters for the MLF model are determined to be 1 hidden layer with 35 neurons, a learning rate of 0.015, momentum of 0.9, an error epsilon of 1.00E-10, and 1000 training cycles. Table 18 summarizes the hyperparameter optimization process for the MLF model. After applying the optimization process for 6 iterations the final confusion matrices of Cross-Validation and testing phases of the optimized MLF models are shown in Table 19 and Table 20. The F-score results of the evaluation steps within each iteration of the grid search for the optimal hyperparameters and the resulting confusion matrices of the Cross-Validation for each iteration are provided in Appendix C.

Table 18 Summary of Hyperparameters Optimization for MLF Model.

Hyperparameter		Value	F-score
Iteration 0	Size of hidden layers	5	88.1
		25	89.59
		5 & 5	87.77
		25 & 25	88.98
		50 & 50	89.15
Iteration 1	Learning rate	0.001	86.03
		0.008	89.49
		0.04	89.84
		0.4	79.95
		0.95	81.07
Iteration 3	Momentum	0.001	88.45
		0.007	88.47
		0.05	88.45
		0.9	89.84
		0.95	88.88
Iteration 4	Error Epsilon	1.00E-10	89.84
		1.00E-06	89.84
		1.00E-04	89.84
		0.001	89.84
		0.1	82.93
Iteration 5	Training cycles	50	87.98
		344	89.54
		706	89.86
		845	89.91
		1000	90.10
Iteration 6	Hidden layer size	5	87.77
		15	89.73
		20	90.04
		35	90.18
		50	89.83
Iteration 7	Learning rate	0.001	88.81
		0.008	90.06

		0.015	90.30
		0.2	89.44
		0.95	81.81
Iteration 8	Momentum	0.01	89.60
		0.065	89.72
		0.1	89.89
		0.55	90.00
		0.9	90.30

Table 19 Cross-Validation confusion matrix of the optimized MLF model.

	true normal	true incident	class precision
pred. normal	10381	553	94.94%
pred. incident	179	3407	95.01%
class recall	98.30%	86.04%	
Accuracy	94.96%	F-score	90.3%

Table 20 Testing confusion matrix of the optimized MLF model.

	true normal	true incident	class precision
pred. normal	2238	149	93.76%
pred. incident	82	1011	92.50%
class recall	96.47%	87.16%	
Accuracy	93.36%	F-score	89.75%

4.2.2. Results of Cross-Validation of neural network.

In section 4.1.2, it was concluded that relying solely on the DR and FAR calculated from the confusion matrices can be misleading, as this approach overlooks the variations in the incident alarms and the assumption about false alarms. Such a simplistic approach can lead to an inaccurate estimation of the performance metrics of the incident detection system. Thus, the DR, MTTD, and FAR of the MLF mode during Cross-Validation and testing phases are calculated using the same method as in the previous sections.

The same Cross-Validation and testing dataset used for RF are used for the MLF. As mentioned before the Cross-Validation data set consists of 121 scenarios, comprising 22 normal conditions and 99 incident scenarios. The 22 normal scenarios in the dataset contain a total of 2640 intervals, while the 99 incident scenarios contain 11880 intervals. The number of false alarms for both sets of scenarios is presented in Table 21,, alongside the number of detected incidents and the MTTD required to identify these incidents.

Table 21 Performance comparison of normal and incident scenario sets for MLF model results from Cross-Validation dataset.

Scenario Set	Detected Incidents	Number of False Alarms	MTTD (minutes)
Normal	N/A	17	N/A
Incident	94	130	0.89

As shown in Table 21, out of the 99 incident cases included in the dataset, the model is unable to detect 5 incidents with a DR of 95.96%. These five incidents are one lane blockage incidents and occurred at 0.6 D/C ratio and the model failed to detect these incidents due to their minor impact during low traffic volume, as mentioned previously in the RF model. To better understand the performance of the MLF model in detecting different incident severities, Table 22 has been constructed to summarize its performance in terms of DR, FAR, and MTTD for each severity level. The Cross-Validation dataset consists of 121 hours of traffic data, with traffic state checked every 30 seconds, resulting in 14,520 model's applications. During these applications, the model generated a total of 147 false alarms. This resulted in a FAR of 1.01%. Therefore, the model's performance during Cross-Validation can be reported as DR = 94.95%, MTTD = 0.89 minutes, and FAR = 1.01%.

Table 22 Performance of MLF Model by Incident Severity Level.

Incident severity	DR(%)	FAR (%)	MTTD (minutes)
1 lane	80.77	1.06	1.98
2 lanes	100	2.08	1
3lanes	100	0.98	0.485
4 lanes	100	0.972	0.5
5 lanes	100	1.21	0.597

As discussed before, it is important to consider other factors, such as the severity of the incident, traffic conditions, and the location and spacing between the monitoring stations, to accurately evaluate the performance of the system. This comprehensive approach can help to identify the strengths and weaknesses of the incident detection system and enhance its overall effectiveness in real-world applications. Thus, a sensitivity analysis is conducted to assess the impact of each of these four factors on the MLF model's performance. It should be noted that this analysis excludes one lane blockage incidents with a 0.6 D/C ratio due to their minor impacts during low traffic volume. Also, the analysis considers the D/C ratios of 0.6, 0.8, 1.0, and 1.2 only. These cases are considered to be consistent with the sensitivity analysis of the RF-model.

4.2.2.1. Impacts of D/C ratio on performance metrics

The DR, FAR, and MTTD are evaluated across four different D/C ratios ranging from 0.6 to 1.2. Figure 36 illustrates the DR values as a function of D/C ratio, which varies from 0.6 to 1.2.

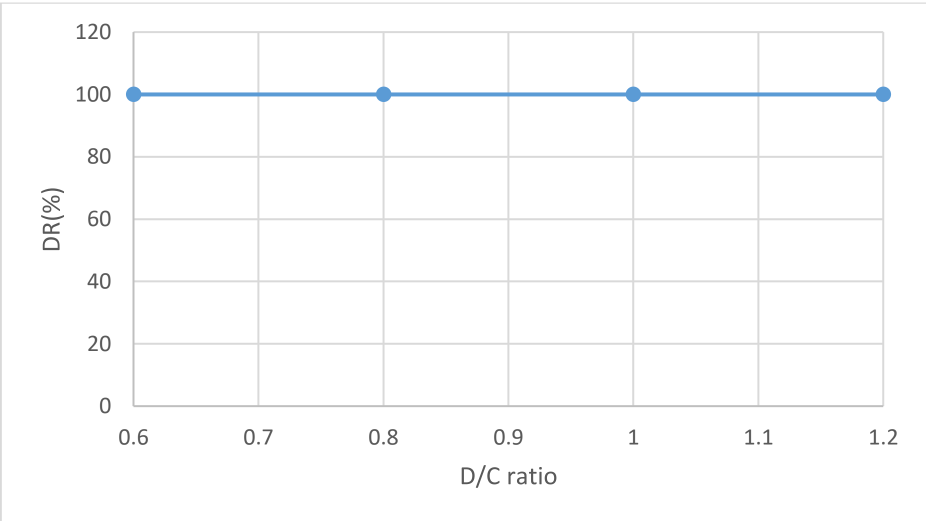

Figure 36 DR vs. D/C ratio for MLF model during Cross-Validation.

Figure 36, reveals that the DR is consistently 100% for all D/C ratios, except for one lane blockage incidents that occurred at 0.6 D/C ratio. Inclusion of these incidents results in a drop of DR to 76.2%. The reason for this drop in DR is attributed to the model's failure to detect five incident scenarios where one lane was blocked at 0.6 D/C ratio. The minor severity of these incidents, coupled with the low traffic volume at which they occurred, contributed to the failure of detection as discussed previously.

Based on the results presented in Figure 37, it can be observed that the FAR varies with D/C. As the D/C ratio increased from 0.6 to 0.8, the FAR is observed to increase, reaching its maximum value of 1.6% at a D/C ratio of 0.8. Subsequently, as the D/C ratio increased from 0.8 to 1.2, the FAR is observed to decrease.

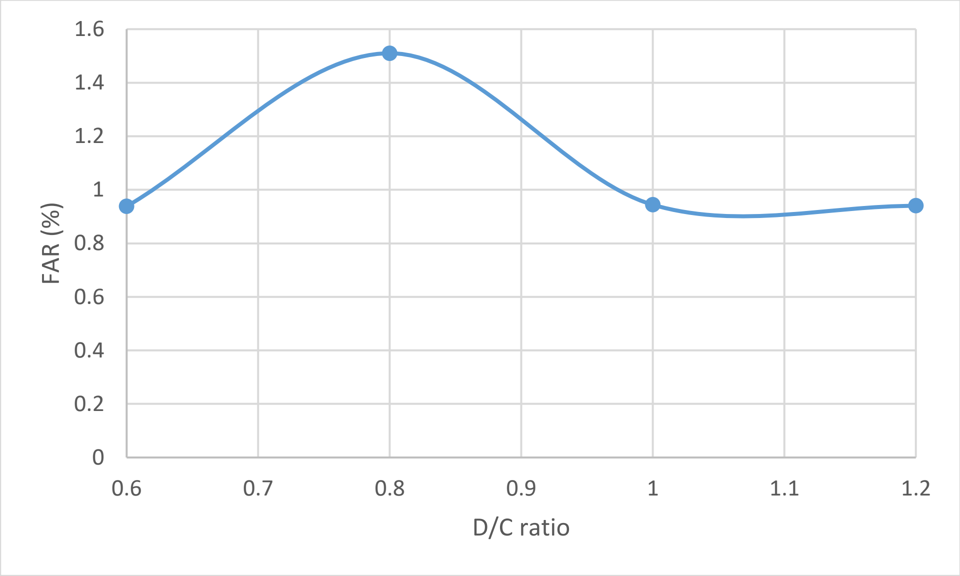

Figure 37 FAR vs. D/C ratio for MLF model during Cross-Validation.

The peak value of FAR at 0.8 D/C ratio can be reasonably justified by the fact that at this ratio, the road is operating close to its capacity with limited spaces and restricted ability to manoeuvre through lanes. Small interruption of traffic flow due to lane changes or manoeuvres may not be easily distinguished from incidents by the MLF model, leading to false alarms. On the other hand, at D/C ratios of 1.0 and 1.2, the road is oversaturated, and vehicles move in platoons, making the occurrence of incidents more distinguishable and easily detected by the model. This is consistent with the relatively stable FAR values observed at 1.0 and 1.2 D/C ratios, as shown in Figure 37, and is also consistent with the findings from Figure 26, which analysed the impact of D/C ratio on FAR for the RF model. Interestingly, when comparing Figure 37 to Figure 26, it appears that the RF model is more effective in distinguishing cases at 0.8 D/C ratio from incident cases compared to the MLF model, which may explain the differences in FAR values observed between the two models at this ratio. It should be noted that including incidents with one lane blockage at D/C of 0.6 resulted in a slightly lower FAR of 0.873% compared to the case without such incidents. However, when these scenarios were excluded, the FAR slightly increased as the number of false alarms was compared to a smaller number of application intervals.

Figure 38 displays the change in the MTTD with the variation of the D/C ratio.

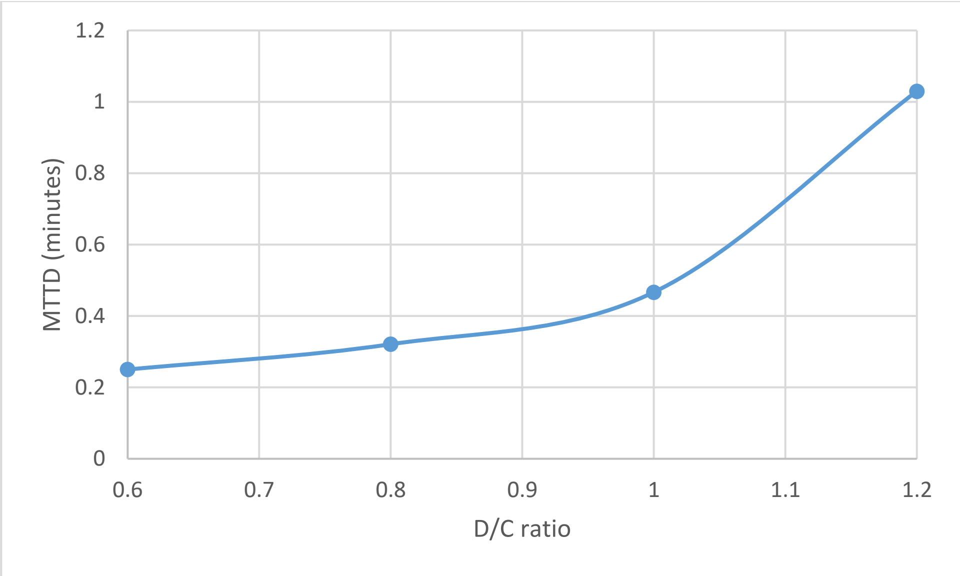

Figure 38 MTTD vs. D/C ratio for MLF model during Cross-Validation.

Figure 38 shows that the MTTD generally increased as the D/C ratio increased. The MTTD is the lowest at a D/C ratio of 0.6 with a value of 0.25 minutes and increased to a maximum value of 1.03 minutes at a D/C ratio of 1.2. The results indicate that the detection time of the incident detection system is influenced by the traffic flow, with higher D/C ratios leading to longer MTTD values. This can be attributed to the presence of platoons of vehicles at high traffic volumes, which may delay the detection of incident impacts, as discussed in the RF model analysis. The patterns exhibited by DR, FAR, and MTTD in response to changes in the D/C ratio align with the findings obtained for the RF model. This consistency in results underscores the influence of D/C ratio on the overall performance of the incident detection system.

4.2.2.2. Impacts of incident severity on performance metrics

The variations of DR, FAR, and MTTD values are examined when the lane blockage varies from 1 to 5 lanes. The results demonstrate that DR values are 100% for all incident severity levels, except for minor severity incidents occurring at a D/C ratio of

0.6, where it drops to 80% when these incidents are included in the evaluation. Figure 39 illustrates the relationship between the Incident's severity and FAR values.

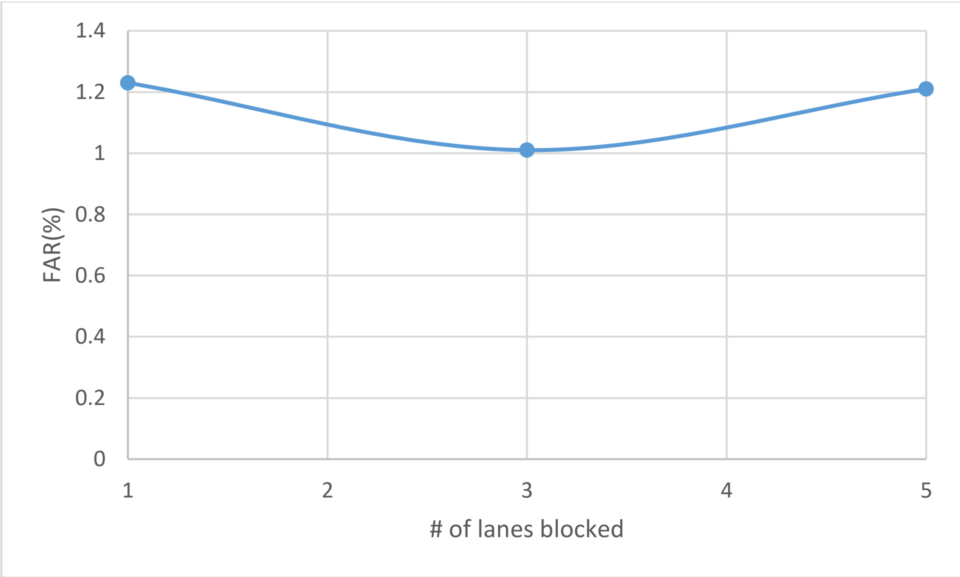

Figure 39 FAR vs. lane blockage for MLF model during Cross-Validation.

Figure 39, reveals that the FAR is marginally influenced by the number of lanes blocked, as observed in Figure 29. Specifically, the FAR for scenarios with one lane blocked was found to be 1.23%, while the FAR for scenarios with three and five lanes blocked were 1.01% and 1.21%, respectively. It is noteworthy to highlight, the MLF model shows slightly higher FAR values than the RF model at the same incident severity levels, which corroborates the earlier observation that the RF model slightly surpasses the MLF model in distinguishing between normal and incident conditions.

The results in Figure 40 show a decreasing trend for the MTTD as the number of lanes blocked increases. This trend is similar to what was observed in the analysis of the RF model's results. For example, the model is able to detect incidents with an average of 2.26 minutes of delay when only one lane is blocked. Incidents with an average delay of only 0.6 minutes are detected when five lanes are blocked. This trend could be due to the fact that incidents with more lanes blocked are more severe and more noticeable, leading to quicker detection. A notable finding is that the exclusion of minor incidents at 0.6 D/C ratio results in a slight increase in the FAR value for one lane blockage, from 1.1% to 1.23%, as well as a slight increase in the MTTD from 1.98 minutes to 2.26 minutes.

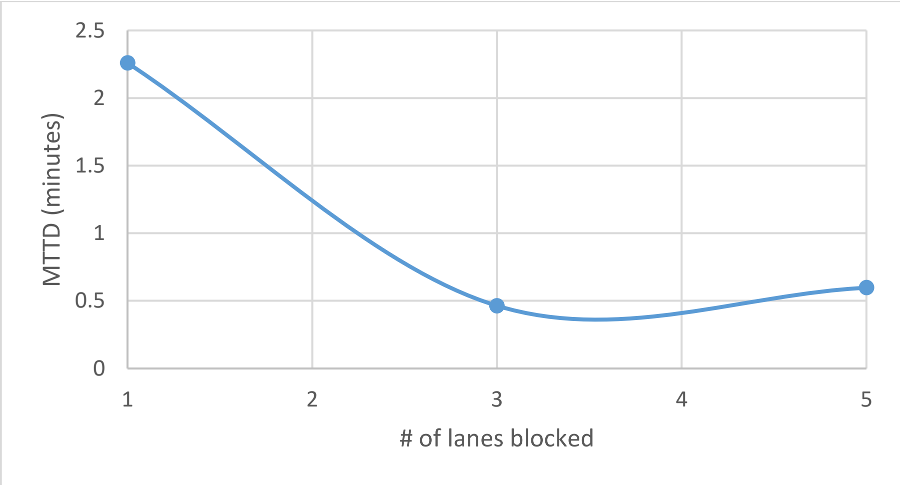

Figure 40 MTTD vs. lane blockage for MLF model during Cross-Validation.

The ANN model's DR, FAR, and MTTD trends with respect to the variation of incident severity are in agreement with the results obtained from a sensitivity analysis of the RF model.

4.2.2.3. Impacts of incident location on performance metrics

An analysis is conducted to assess the variation in DR, FAR, and MTTD as the incident location is positioned at different distances (0.25, 0.5, and 0.75) from the upstream detector. The DR values are 100% for all incidents' locations. This indicates that the MLF model is successful in detecting incidents across different locations. Figure 41 indicates that the FAR decreases from 1.33% to 0.74% as the incident moves farther from the upstream detector.

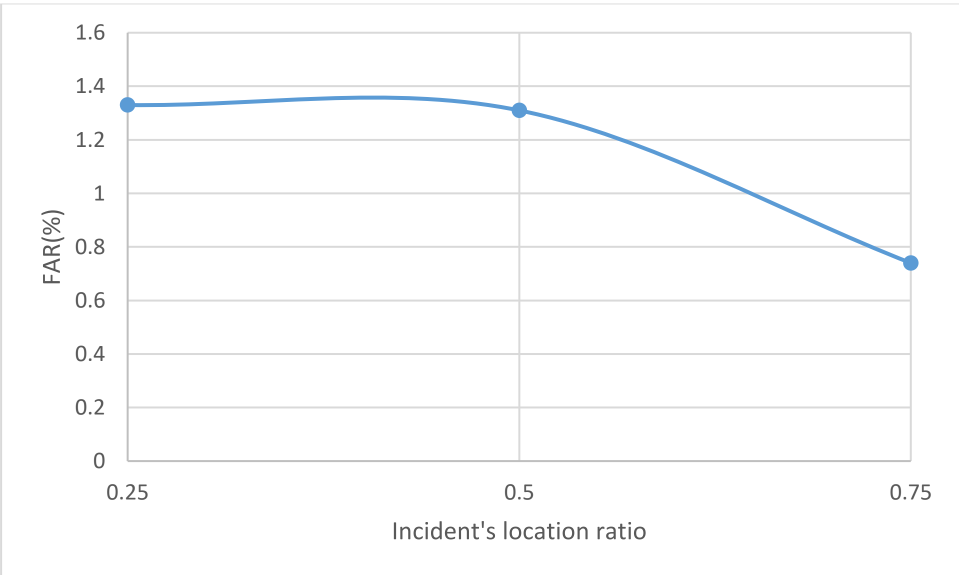

Figure 41 FAR vs. incident's location for MLF model during Cross-Validation.

The relationship between the MTTD and incident's location is showed in Figure 42.

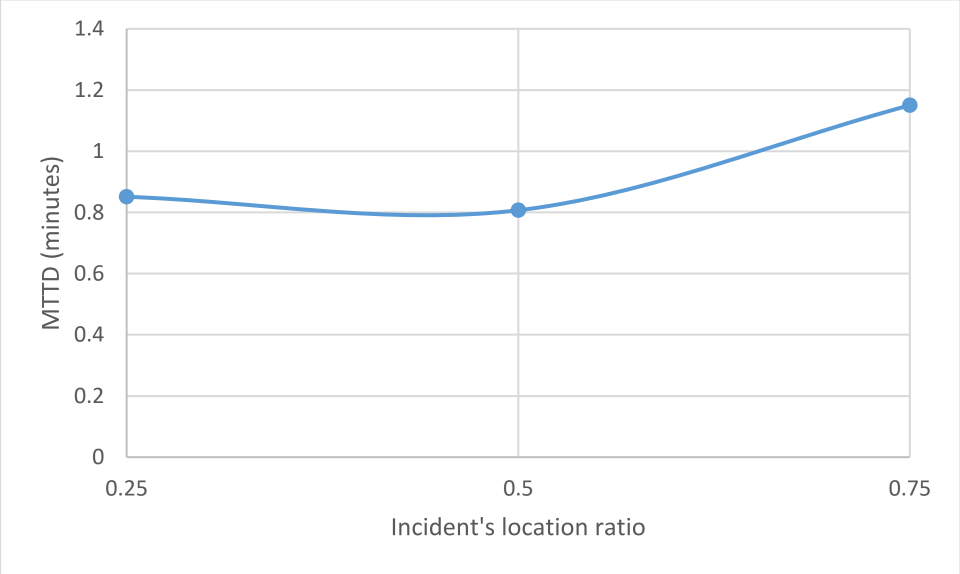

Figure 42 MTTD vs. incident's location for MLF model during Cross-Validation.

The MTTD increases as the distance between the location of the incident and the upstream detector increases, as depicted in Figure 42, similar to what was observed in Figure 32. This suggests that the delay in detecting the incidents is due to the longer time it takes for the incident impact to propagate upstream. This can also explain the negative association between the FAR and the incident's location in Figure 41. The extended time for the incident impact to propagate upstream could indeed be a reason why the FAR decreases as the incident occurs farther from the upstream detector station. When the impact of incidents is delayed, the disturbances to the stable traffic flow are also delayed, reducing the likelihood of creating disturbances that can be falsely detected as incidents by the model thus the FAR is reduced.

4.2.2.4. Impacts of distance between detectors on performance metrics

The DR, FAR and MTTD values are measured when the distance between the two detectors stations varies from 500 to 1500 meters. The DR for the incident detection system remained 100% for all three detector spacing values (500m, 1000m, and 1500m), except when considering minor one lane blockage incidents at a 0.6 D/C ratio. The inclusion of these incidents led to a reduction in DR values, with rates of 97.05%, 96.88%, and 89.67% observed for detector distances of 500m, 1000m, and 1500m, respectively. Notably, among the five minor incidents at 0.6 D/C ratio that were not detected, one occurred at 500m detector spacing, one at 1000m detector spacing, and the remaining three at 1500m spacing.

The association between FAR and detectors' spacing is illustrated in Figure 43.

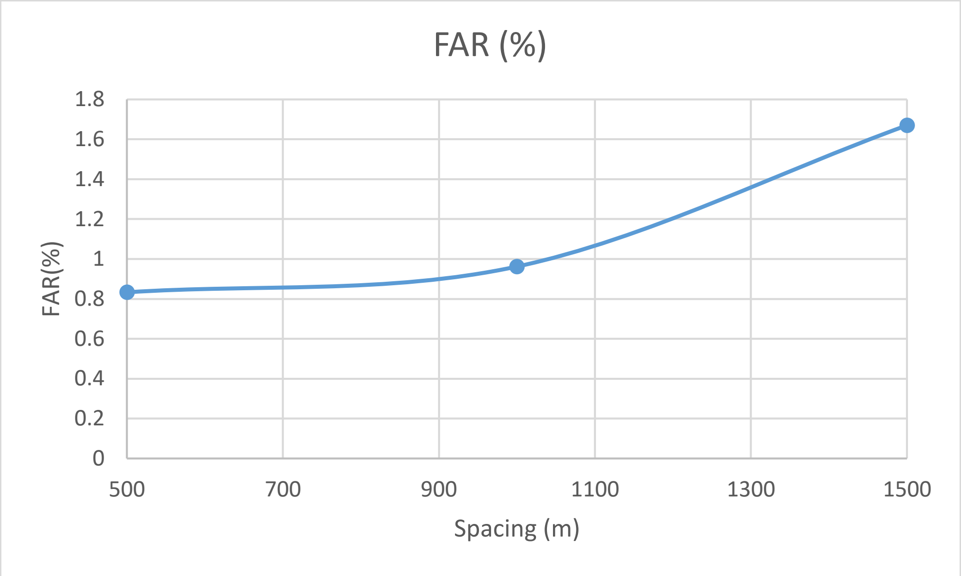

Figure 43 FAR vs. detectors' spacing for MLF model during Cross-Validation.

The FAR values in Figure 43 demonstrate that FAR increases as the spacing between the detectors increases from 500 to 1500 meters. As the distance between the detectors increases, there could be more traffic parameter differences between the upstream and downstream stations. These differences could be caused by factors such as weaving of ramp in and ramp out, lane changing, and the long travel time between the detectors. These differences can be mistakenly detected as incidents, leading to a higher number of false alarms.

The MTTD values illustrated in Figure 44 show a positive correlation between the spacing between upstream and downstream detectors and the MTTD values. This is consistent with the findings observed in in Figure 34, where the RF model also showed an increase in MTTD with larger detector spacing. There are several reasons why MTTD increases with an increase in the distance between detectors. One such factor is the longer travel time for vehicles to travel between detectors, resulting in a delay in detecting incidents as previously stated. In addition, the more complex traffic patterns that arise from longer distances between detectors, such as weaving between lanes and merging/diverging traffic, make it difficult to detect incidents in a timely manner.

Moreover, reduced sensor density due to gaps in coverage between detectors can also contribute to an increase in MTTD.

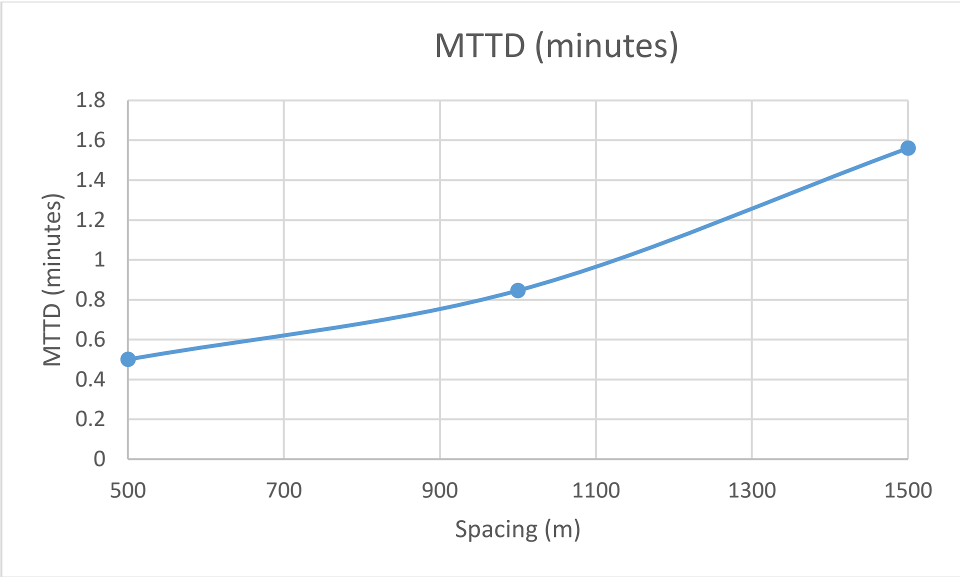

Figure 44 MTTD vs. detectors' spacing for MLF model during Cross-Validation.

The analysis shows that increasing the spacing between detectors resulted in an increase in the FAR and MTTD values, with the spacing of 15000 meters having the highest FAR and MTTD values. These observations align with the analysis of the impact of spacing between detectors on the DR, FAR, and MTTD for the RF model.

The performance metrics of DR, TTD, and false alarms for each scenario in the Cross-Validation dataset are presented in Appendix D-Table D 1. A sample of incidents in the Cross-Validation dataset is presented in Table 23 that shows their TTD and false alarms.

Table 23 Sample calculation of DR, MTTD, and false alarms for Cross-Validation dataset for MLF model.

Number of blocked lanes	Incident location	Spacing	D/C ratio	Detection Status	TTD (intervals)	TTD (minutes)	False alarms
3	250	500	0.4	Detected	0	0	1
4	250	1000	0.5	Detected	1	0.5	1
1	250	500	0.6	Not Detected	-	-	0
3	250	500	0.8	Detected	1	0.5	0
5	375	500	1	Detected	0	0	0
1	1125	1500	1.2	Detected	32	16	1
4	625	1250	0.9	Detected	1	0.5	1
2	375	750	1.1	Detected	3	1.5	1

Overall, the model succeeded in correctly detecting 95 incidents out of 99, yielding a DR of 95.96%, and a MTTD of 0.89 minutes. During 121 hours of traffic data, the model yielded 147 false alarms, resulting in a FAR of 1.01%.

4.2.3. Results of testing phase of neural network.

In this section, the DR, FAR and the MTTD of the MLF model are calculated and discussed. A sample of these calculations of an incident scenario in the testing dataset is shown in Table 24, which depicts the actual traffic status during a five -lanes blockage incident that occurred 1125 meters from the upstream detector station, with the distance between the upstream and downstream detector stations being 1500 meters. The incident occurred at 1.0 D/C ratio, and the table displays the model's predictions of some intervals along with a comparison between the actual status and the predictions.

Table 24 Sample calculation of DR, MTTD, and false alarms testing dataset for MLF model.

intervals	Actual Status	Model's Prediction	Comparison Result
1-30	normal	normal	True
31-33	incident	normal	Missed
34-70	incident	incident	True
71-73	normal	incident	False Alarm
74-107	normal	normal	True
108	normal	incident	False Alarm
109-120	normal	normal	True

As it can be noted from Table 24, the model is able to detect the incident 3 intervals after it occurred, with a TTD of 3 interval or 1.5 minutes, as the incident is detected at interval 34. Additionally, one false alarm is generated at interval 7, three false alarms are generated at intervals 71,72 and 73 and one false alarm is generated at interval 108. As the false alarms at intervals 71,72 and 73 occur at consecutive intervals, they are counted as only 1 false alarm. Therefore, the total false alarms generated in this incident scenario is 3. The DR, TTD, and number of false alarms for each scenario in the testing dataset is presented Table 25.

Table 25 TTD and false alarm of testing for MLF model.

Number of blocked lanes	Incident location	Spacing	D/C ratio	Detection Status	TTD (intervals)	TTD (minutes)	False alarms
3	250	500	0.5	Detected	0	0	1

3	250	1000	0.5	Detected	1	0.5	1
1	125	500	0.6	Detected	10	5	2
3	250	500	0.6	Detected	1	0.5	1
3	250	1000	0.6	Detected	1	0.5	1
1	500	1000	0.6	Detected	1	0.5	1
5	750	1000	0.6	Detected	0	0	1
5	750	1500	0.6	Detected	4	2	2
1	250	500	0.8	Detected	4	2	1
1	250	1000	0.8	Detected	3	1.5	1
3	375	500	0.8	Detected	9	4.5	2
3	375	1500	0.8	Detected	7	3.5	5
5	500	1000	0.8	Detected	1	0.5	1
3	1125	1500	0.8	Detected	1	0.5	1
1	125	500	1	Detected	1	0.5	0
1	500	1000	1	Detected	1	0.5	2
3	750	1000	1	Detected	1	0.5	2
3	750	1500	1	Detected	2	1	1
5	1125	1500	1	Detected	3	1.5	2
5	125	500	1.2	Detected	23	11.5	3
1	375	500	1.2	Detected	0	0	0
1	375	1500	1.2	Detected	1	0.5	4
1	750	1000	1.2	Detected	1	0.5	1
1	750	1500	1.2	Detected	1	0.5	1
3	1125	1500	1.2	Detected	1	0.5	2
4	375	750	0.9	Detected	1	0.5	3
2	625	1250	0.9	Detected	1	0.5	0
4	375	750	1.1	Detected	0	0	2
2	625	1250	1.1	Detected	11	5.5	1
					3.14	1.6	45

Table 25 shows that the model succeeded in correctly detecting all the 29 incidents in the testing dataset, yielding a DR of 100%, and a MTTD of 1.6 minutes. During 29 hours of traffic data, the model yielded 45 false alarms, resulting in a FAR of 1.29%. As mentioned before, the differences in the model's detection performance between the Cross-Validation and testing phases may be attributed to the detector location and spacing scenarios used during Cross-Validation, along with the learning and improvements made by the model.

4.2.4. Discussion of neural network model results

In this section, the results of the MLF model during Cross-Validation and testing phases are discussed. The performance of the MLF model is evaluated using three performance measures, DR, MTTD and FAR. In the Cross-Validation phase, the model achieved a DR of 95.96%, with an MTTD of 0.89 minutes and a FAR of 1.01%. On the other hand, in the testing phase, all 29 incidents were correctly detected, resulting in a DR of 100%, The MTTD in the testing phase was 1.6 minutes and the FAR was found to be 1.29%. Both results indicate a high level of accuracy and efficiency in detecting incidents. The size and complexity of the datasets in each phase may have contributed to the variation in the results. The sensitivity analysis was conducted to explore the potential impact of various factors on the performance measure of the incident detection system. The findings reveal that the efficiency and effectiveness of the system are influenced by multiple factors, such as traffic volume, incident severity, the distance between the detector station and the incident location. The results demonstrate that these factors can significantly affect the accuracy and speed of incident detection, which highlights the importance of considering these factors while developing and implementing the incident detection system.

The discussion focuses on the collective influence of the four factors on the detectability, TTD, and FAR of incidents during the Cross-Validation and testing stages of the MLF, following a thorough analysis of each factor's impact individually. Similar to the RF model, an evaluation is carried out to gain insight into the detection failures or incidents with extended TTD, with a focus on a range of incident scenarios. The aim is to comprehend the combined effect of all four factors on the detectability, TTD, and FAR of incidents in the Cross-Validation and testing phases. As mentioned before, the MLF model failed to detect five incidents that blocked only one lane and occurred at 0.6 D/C ratio. Several factors contributed to this failure in detection, including the minor severity of the incidents, the low traffic volume at which they occurred, the location of the incidents, and the distance between the upstream and downstream detector stations. The distance between the incidents and the upstream detector station was a crucial factor that affected detection performance. These five incidents in the Cross-Validation dataset were observed at a distance equal to or greater than 0.25 of the spacing between the upstream and downstream detectors. Notably, four of these incidents occurred when the detector spacing was set at 1000 meters or 1500 meters.

The increased distance between detectors can lead to lower detection accuracy, resulting in missed or undetected incidents. Furthermore, the minor severity of the incidents and the low traffic volume at the time of the incidents also contributed to the detection failure. These factors can make it difficult for the model to differentiate between incidents and normal fluctuations in traffic flow, resulting in a high FAR or missed incidents. Therefore, the combined influence of these four factors, including the severity of the incidents, the traffic volume, the location of the incidents, and the distance between the detectors, can result in the undetectability of incidents.

During the Cross-Validation phase, four incidents that blocked a single lane were detected with a TTD of over five minutes. The first incident occurred at a D/C ratio of 0.8, with a detector spacing of 1000 meters, and took six minutes to be detected. This incident was located 750 meters from the upstream station. Similarly, the second incident occurred at a D/C ratio of 1.0, with the same detector spacing and distance from the upstream station, and also took six minutes to be detected. The third incident occurred at a D/C ratio of 1.2, with a detector spacing of 500 meters and 125 meters from the upstream detector. It took 13 minutes to detect this incident. The longer TTD for the first two incidents can be attributed to their minor severity, relatively larger detector spacing, and distant location from the upstream station. In comparison, the third incident had a higher TTD than the first two incidents. This is due to the impact of vehicles platooning during high traffic volume, such as a D/C ratio of 1.2, as discussed in section 4.1.4 and visually depicted in Figure 35. Therefore, congestion level in combination with the high detector spacing and minor severity of the incident, it contributed to an extended TTD of 13 minutes. Similarly, the fourth incident had the longest TTD, as it occurred at a D/C ratio of 1.2, was located 1125 meters from the upstream station, and had a detector spacing of 1500 meters. The impact of congestion during high traffic volume further delayed the detection. Hence, the combination of the minor severity of the incident, the highest detector spacing of 1500 meters, distant location from the upstream detector, and the impact of congestion during high traffic volume collectively contributed to the longest TTD of 32 minutes for the fourth incident.

Another incident blocking five lanes occurred with a 1.2 D/C ratio. The incident is located 1125 meters from the upstream station, and the detector spacing is 1500 meters. The model took ten minutes to detect the incident. This delay in detection can be attributed to a combination of three factors. Firstly, a high D/C ratio of 1.2 is already present, indicating heavy traffic congestion. When a severe incident occurs, it can exacerbate the congestion, resulting in a further buildup of queued vehicles. This can cause vehicles to queue up, leading to congestion, and the incident may not immediately impact the traffic flow at the upstream station, causing a delay in detection. This factor was previously observed in the RF analysis as well. Additionally, the incident is located 1125 meters from the upstream station, and the detector spacing is 1500 meters, which can contribute to a longer TTD. This is because the incident is farther away from the detector, and it takes longer for the effects of the incident to propagate upstream and be detected by the model. Therefore, the combination of a high demand to capacity ratio and a large distance between the incident and the detectors are likely the main factors contributing to the longer TTD in this scenario.

In the testing phase of the MLF model, a one lane blockage incident occurred at a D/C ratio of 0.6, located 125 meters from the upstream detector station, with a spacing of 1000 meters between detectors. The model took 10 minutes to detect the incident. In this particular incident scenario, three factors (D/C ratio, incident severity, and detector spacing) contributed to the delay in detection by the model. Additionally, a five-lane blockage incident occurred at 1.2 D/C ratio and is located 125 meters from the upstream detector station. The spacing between the two detector stations is 500 meters, and the model took 23 minutes to detect this incident. This delay in detection can be attributed to the phenomenon of vehicles platooning during periods of high traffic volume, as illustrated earlier. Another incident occurred during the testing phase of the MLF model, where two lanes are blocked at a D/C ratio of 1.1. The incident is located 625 meters from the upstream detector station, and the spacing between the detectors is 1250 meters. The model took 11 minutes to detect the incident. The delay in detection can be attributed to the high D/C ratio, the location of the incident, and the spacing between the detectors.

In summary, it is evident that the incident detection system's accuracy is influenced by several critical factors, including incident severity, traffic conditions, and detector location and spacing. The findings from both the Cross-Validation and testing phases highlight the substantial impact of these factors on incident detectability and MTTD. It is crucial to consider these factors, either individually or collectively, when implementing an incident detection system. Overall, the results emphasize the significant potential of the system as a valuable tool for incident detection.

4.3. Comparison of RF and MLF Models with other AID Systems in the Literature.

This section aims to compare the performance of the two developed models, the RF and MLF models, with some of the AID models identified in the literature review that exhibit high performance. The effectiveness of these models is evaluated and compared using key performance metrics such as DR, FAR, and MTTD. The DR, FAR, and MTTD reported in the literature for the selected AID model are summarized in Table 26.

Table 26 A Comparative analysis of AID performance metrics: DR, FAR, and MTTD.

Authors	Models	DR(%)	FAR(%)	MTTD (minutes)
Texas Transportation Institute [18]	SND based algorithm	92	1.3	1.1
Motamed [24]	SVM_L , SVM_RB, SVM_P, MLF, PNN	SVM_L : 87 SVM_RB : 91.3 SVM_P : 91.3 MLF: 82.6 PNN : 95.6	SVM_L : 0.07 SVM_RB : 0.07 SVM_P : 0.01 MLF: 0.06 PNN : 0.3	SVM_L : 4.3 SVM_RB : 5.45 SVM_P : 2.25 MLF: 3.25 PNN : 3.84
Xie et al. [25]	hybrid AID method (SASYNO-RF-RSKNN)	94.7 – 97.3	0 - 0.061	Not provided
Cheu and Ritchie [54]	MLF model	78 - 80	0.17 - 1.5	3.23 – 4.95

Andrea and Marcelloni [76] [71]	system for detecting traffic congestion and incidents from real-time GPS traces	91.6	8.3	7
Ahuja [93]	IQD-based algorithms, Decision Tree and RF	IQD_speed : 91 – 94. IQD_ speed & occupancy: 88-92 Decision Tree: 94-97 RF : 94-97	IQD_speed : 3.8-5.4 IQD_ speed & occupancy: 2-4 Decision Tree: 1-3 RF: 1-3	Not provided
Chakraborty et al. [96], [97]	IQD based algorithm	97	4.8	12.4
Zyryanov [126]	MLF model	97.6 for the training dataset, 94.4 for the testing dataset, and 88.9 for the validation dataset.	Not provided	Not provided
Rossi et al. [140]	FL based model	93.09	0.445	2.95
Dogru and Subasi [149]	ANN, RF and SVM	ANN : 86.1 RF : 94 SVM : 88	ANN: 8 RF: 0.203 SVM : 4.2	Not provided

Ren el al. [164]	video-based detecting and positioning method	96.6	0.72	1.16

The performance of the developed models (highlighted in yellow) in terms of DR, FAR, and MTTD is illustrated in comparison to other existing models in the literature in Figure 45, Figure 46 and Figure 47.

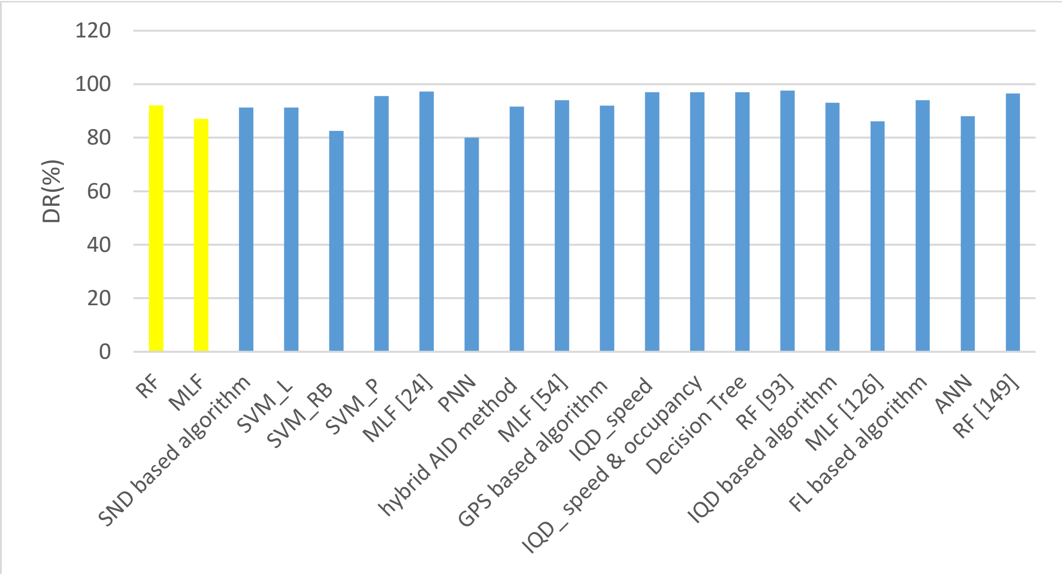

Figure 45 Comparison of DR between existing AID systems and developed models.

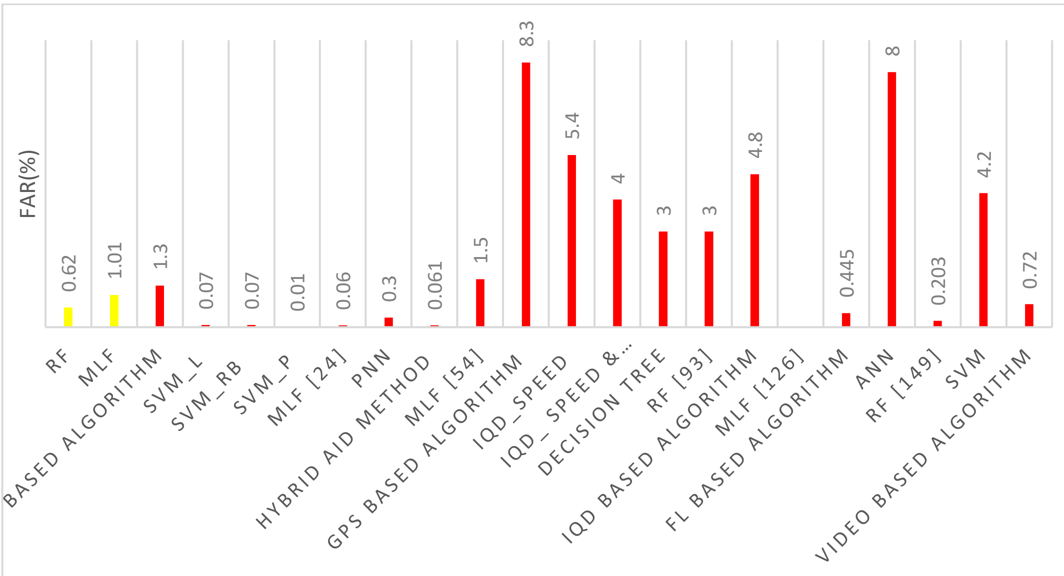

Figure 46 Comparison of FAR between existing AID systems and developed models.

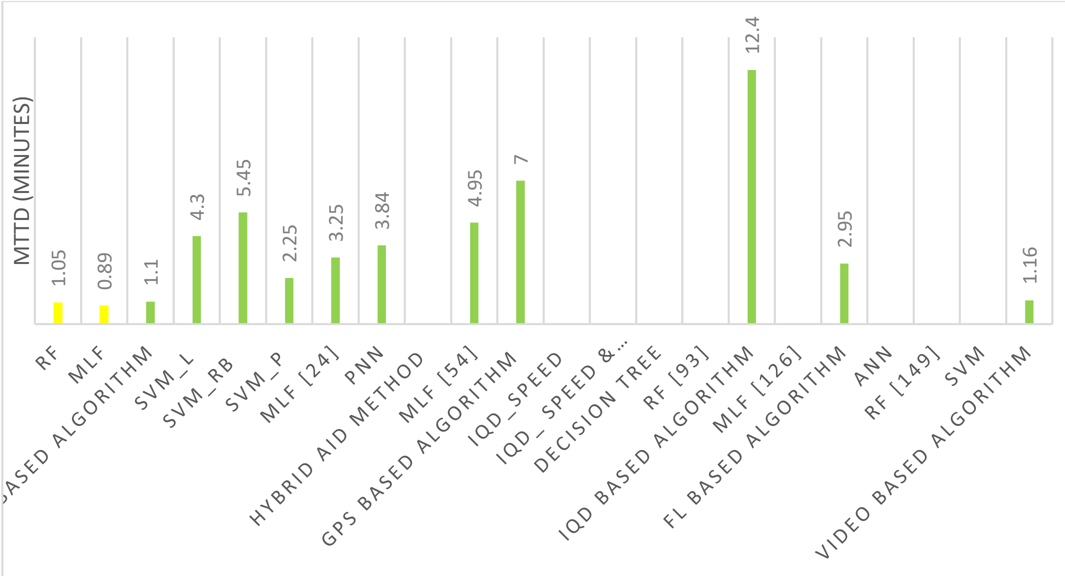

Figure 47 Comparison of MTTD between existing AID systems and developed models.

By comparing the performance of two models developed in this dissertation, RF and MLF, for incident detection. Both models achieved high accuracy rates, with the RF model exhibiting a slightly higher DR of 96.97% during Cross-Validation, compared to the MLF model's DR of 95.96%. The MLF model had a shorter MTTD of 0.89 minutes compared to the RF model's MTTD of 1.05 minutes. However, the MLF model had a slightly higher FAR of 1.01% compared to the RF model's FAR of 0.62%. During the testing phase, both models achieved perfect DR with a score of 100%, but the MLF model had a longer MTTD of 1.6 minutes compared to the RF model's MTTD of 1.17 minutes. Additionally, the MLF model had a higher FAR of 1.29% compared to the RF model's FAR of 0.862%. The slight differences in performance between the two models could be attributed to the different algorithms used in each model, as RF relies on decision trees and MLF relies on a neural network. Optimized hyperparameters in each model could also contribute to the slight differences. Based on the results of the two models, it can be inferred that RF has slightly better performance compared to MLF. Moreover, it is worth mentioning that all the models were run on the same PC with AMD Ryzen 7 5800H with Radeon Graphics, running at 3.20 GHz, 8 cores, 16.0 GB RAM and a 500 GB SSD. During the execution of the two models in RapidMiner, the RF model took 47 seconds to execute, while the MLF model took 2 minutes and 28 seconds, which is over three times longer, using the same computer. This provides an advantage to the RF model over the MLF model, which could be better suited for real-time applications. It should be noted that the execution time of an ML model can significantly impact its MTTD when deployed in real-world scenarios. Therefore, based on the results and execution time comparison, the RF model is more recommended for incident detection in time-critical situations. Yet, both models demonstrated their ability to detect traffic incidents with high accuracy rates, making them promising options for incident detection systems. However, it should be noted that these models were trained and tested on simulated traffic data and did not consider weather conditions. Future work could involve testing these models on traffic data and incorporating weather conditions into the models for more accurate incident detection. As illustrated in Table 26 , some AID models achieved better performance in terms of one or more of the performance metrics in comparison with the two developed models. However, these models have some limitations and weaknesses. The hybrid AID method developed by Xie et al. [25] achieved high DR and low FAR on the I-205 and I-880 datasets. Specifically, the framework achieved a DR of 97.3% and FAR of 0.061% on the I-205 dataset, and a DR of 94.7% and FAR of 0% on the I-880 dataset. However, a weakness of this framework is that the MTTD was not measured, which could limit the understanding of the model's effectiveness in real-world scenarios, thus indicating a potential weakness. Another potential limitation is that the framework used an oversampling technique to generate synthetic incident samples. The oversampling technique used in the framework to generate synthetic incident samples can potentially limit the real-world applicability of the developed model. The reason for this is that the synthetic samples generated do not represent actual traffic incidents, which can lead to false positives or overestimation of the incident detection performance when the model is applied to real-world scenarios. In other words, the model's performance on real-world data may differ from its performance on synthetic data generated through oversampling. Another potential limitation of the oversampling approach is overfitting. Overfitting can occur when a model is trained on a limited set of data, resulting in the model performing well on the training data but poorly on new and unseen data. By oversampling the incident samples, the model may become too focused on the incident cases in the training data, which could lead to overfitting and poor generalization to new data. Furthermore, the use of synthetic incident samples to balance the database between incident and non-incident cases may not be realistic. In real life, incidents are often a minority class and non-recurrent events. Therefore, balancing the incident and non-incident cases in the database may not be necessary, and may even introduce biases in the model's training data. The decision tree and RF developed by Ahuja [93] achieved high DR and reasonable FAR but they didn’t measure the MTTD of these models. While DR and FAR are important metrics to evaluate the performance of an AID system, they only provide a partial picture of how well the system performs in real-world scenarios. In addition to detecting incidents accurately and minimizing false alarms, it is also crucial for an AID system to detect incidents quickly, as the timely detection of incidents can help reduce their impact on traffic flow and prevent secondary accidents. Likewise, The MLF model developed by Zyryanov [126] accomplished very slightly higher DR than the two developed ML models in this dissertation during training phase. Nonetheless, the evaluation of the model's performance only considered DR and did not include FAR and MTTD, which are critical metrics in assessing the model's reliability and efficiency. As a result, the evaluation may not provide a comprehensive understanding of the model's effectiveness, limiting its practical applications. On the other hand, the FL system developed by Rossi et al. [143] has lower FAR than the two developed ML models but it has lower DR and higher MTTD. Additionally, the training and testing of the system were based on simulated traffic data, and the study only tested the system's performance for a limited range of flow rate values and distances among detector stations without considering other factors. The RF model created by Dogru and Subasi [149] achieved lower FAR of 0.2% but it has lower DR of 94% and the MTTD of this model was not reported. Besides, their models have limited practical applicability due to several assumptions. For example, the assumption that only probe vehicles are equipped with V2V communication devices and that they broadcast their position and speed every second. Additionally, the models assume that probe vehicles can calculate the position of transferring vehicles using signal processing and antenna techniques and that they can aggregate microscopic traffic values for each vehicle over the last 10 seconds. These assumptions may not hold in real-world scenarios and thus limit the models' practical use. The video-based detecting and positioning method developed by Ren el al. [164] attained higher DR and FAR than the MLF developed in this dissertation. In spite of that, the lighting conditions, extreme weather conditions and coverage range of the camera that is used to capture traffic video are the main limitations that can impact the performance of this method. In addition, the method requires a significant number of computational resources and may not be suitable for real-time applications or for use in areas with limited computing power. Moreover, they used oversampling technique to generate synthetic incident sample to balance between incident and non-incident samples in the datasets used to train and test the model. As mentioned earlier, using oversampling technique to generate artificial incidents is not realistic.

In summary, the comparison of the developed RF and MLF models in this dissertation with other AID systems showed that the models have better performance than most of them. The developed ML models achieved high DR, low FAR, and MTTD values, indicating their effectiveness in detecting incidents accurately and quickly. These models also considered a wider range of factors that have significant impacts on AID performance. Compared to previous models, the developed models integrated more factors, making them more comprehensive and generic. The aim of the developed models is not only to detect incidents accurately and rapidly, but also to establish effective and generic incident detection models for freeways. Taking into account a broader range of factors, could provide more accurate and reliable incident detection results, thereby reducing the impact of incidents on freeway operations and improving traffic safety. Additionally, the sensitivity analysis conducted for both the RF and MLF models revealed that the trends of DR, FAR, and MTTD with the variations of the four significant factors are consistent between the two models. This consistency in the trends of the performance metrics between the two models can confirm and validate the observations made in each model. Moreover, the use of ML techniques such as RF and ANN enabled the models to learn from data and adapt to changing traffic conditions, making them more robust and adaptable to new scenarios. This ensures the generalizability of the models to new data and the potential for future applications in real-time traffic management systems. Overall, the developed RF and MLF models offer a promising solution for accurate and effective incident detection on freeways.

Chapter 5. Summary and Conclusion

5.1. Summary

This dissertation aimed to develop efficient and reliable models for detecting traffic incidents, which are a major cause of delays, air pollution, and reduced gross domestic product in many countries around the world. The study area is a section of Sheikh Mohammad Bin Zayed Road (E311) in Dubai, UAE and it is simulated using VISSIM, which generates a diverse dataset by considering four factors namely, the congestion level, the distance between the upstream and downstream counter stations, the location of the incident relative to the counter stations, and the severity of the incident. The use of simulated data instead of real data provided benefits such as the ability to generate diverse and challenging scenarios for the ML models. The simulation also allowed for the collection of precise and accurate data on traffic parameters, including flow rate, speed, and occupancy at upstream and downstream stations, which are used as input variables for the models. The dataset is used to train and test two ML models, RF, and MLF using RapidMiner software. Notably, in the case of the RF model, the main hyperparameters that are optimized in this dissertation encompassed the number of trees, maximal depth, subset ratio, and the confidence parameter of pruning. Likewise, for the MLF model, the hyperparameters that were optimized in this dissertation consisted of the number of hidden layers, the size of hidden layers, the learning rate, momentum, error epsilon, and training cycles. Both models achieved high detection rates and low false alarm rates during testing. The RF model achieved a DR of 96.97%, MTTD =1.05 minutes and FAR of 0.62% during training, and a DR of 100%, MTTD = 1.17 minutes and FAR of 0.862% during testing. Similarly, the MLF achieved a DR of 95.96%, MTTD = 0.89 minutes FAR of 1.01% during training, and a DR of 100%, MTTD =1.6 minutes and FAR of 1.29% during testing. Sensitivity analysis is conducted to evaluate the impact of each of the four factors on the models' performance in terms of DR, FAR and MTTD. It should be noted that the analysis excluded a lane blockage incident that occurred at 0.6 D/C ratio. The sensitivity analysis revealed that these factors should be considered together when developing incident detection systems, as they have a significant impact on performance. The use of simulated data provided benefits, but the models should also be tested on real traffic data in future work. The study also did not consider weather conditions, which should be considered in future work as well.

5.2. Conclusions

This dissertation developed and tested RF and MLF models, which outperformed other models in the literature. The results demonstrate that both models achieved reliable and efficient detection of traffic incidents, as indicated by their DR, MTTD, and FAR values. The sensitivity analysis conducted in this dissertation has revealed several important conclusions:

- Incidents with minor severity that occur during low traffic volume may have a low impact on the traffic performance, making them difficult to detect or even go unnoticed by the incident detection system, this aligns with prior research findings [53], [56], [82], [94], [143]. This is because the model relies on changes in traffic patterns and flow characteristics to detect incidents, and during low traffic volume, the changes caused by minor incidents may not be significant enough to trigger incident alarm.
- The FAR decreases with an increase in D/C ratio, as higher congestion levels during high traffic volumes result in more consistent traffic behavior, making it easier for the incident detection system to recognize abnormal patterns. In contrast, lower traffic volumes may have widely spaced vehicles with varying speeds, making it harder to distinguish normal from abnormal behavior. This is in line with the findings of previous studies [70], [94], [143], [146].
- The MTTD values tend to increase as D/C ratio increases, possibly due to queues forming at blocked sections of the road during high traffic volumes. This can delay the effects of an incident on traffic performance, resulting in longer detection times. This result is in agreement with the findings of Rossi et al. [143] and Deniz and Celikoglu [91].
- The MTTD values tend to decrease with higher incident severity, as incidents with more severe lane blockages have a greater impact on traffic flow, making them easier to detect and resulting in shorter detection times. This results supports the findings of previous studies [54], [147].
- The MTTD increases as the distance between the incident location and upstream detector increases, as it takes longer for the effects of the incident to

propagate upstream. This delay in detection leads to higher MTTD values. This finding is in agreement with prior studies [72], [90], [143].

- The FAR decreases as the incident moves farther from the upstream detector, as delayed impact of incidents results in delayed disturbances to traffic flow. This reduces the likelihood of false detections by the model, leading to a decrease in FAR.
- The FAR increases with distance between detectors, possibly due to longer travel times of vehicles between detectors causing fluctuations in traffic measurements such as volume, speed, and occupancy. These fluctuations can trigger false incident alarms, resulting in an increase in FAR. This finding is consistent with earlier studies [143].
- Larger detector spacings are associated with longer detection times, as increased travel time between detectors can cause delays in detecting incidents, leading to longer MTTD values , consistent with the results reported in previous studies [72], [143].
- Finally, the importance of using persistence tests to reduce false alarms is also emphasized in this dissertation. The assumption of treating consecutive false alarms as a single false alarm if they persist for four or fewer intervals is made to reduce the false alarm rate. However, it is important to note that this assumption can affect the detection time of an incident, as an incident may occur during the two minutes that the alarms are ignored. Therefore, it is important to strike a balance between reducing the false alarm rate and not extending the MTTD of incidents that may occur after a false alarm.

In conclusion, this dissertation provides effective and generic incident detection models for freeways that consider various factors and traffic variables, which were not previously considered together in previous research studies. These findings provide a framework for incident detection systems that can be generalized for use in different regions with different traffic conditions. The results suggest that the RF and MLF models are reliable and efficient methods for detecting traffic incidents. Overall, the sensitivity analysis conducted in this dissertation provides valuable insights into the factors that affect the accuracy and performance of the AID system. The observations made in this study could help researchers and practitioners to better understand the limitations of the AID system and to develop strategies to improve its accuracy and performance. By improving the accuracy and efficiency of traffic incident detection systems, the negative impacts of traffic incidents on the environment and economy can be mitigated.

5.3. Future Work and Recommendations

It is important to note that the models presented in this dissertation were developed using simulated data due to the previously mentioned benefits and limited access to real data. Furthermore, some assumptions were made when modelling the study area, such as the speed distribution of vehicles and trip generation and distributions. Therefore, there is still a possibility for further research to be conducted to improve and extend the findings of this dissertation. Based on the results of this study, the following recommendations are suggested for future research and practical implementation:

1. Test the developed models on real data to verify their effectiveness and reliability.
2. Expand the study area to include other freeways or highway systems with varying traffic conditions and incidents to determine the generalizability and effectiveness of the models.
3. Explore the potential of utilizing different ML models or combinations of ML models to develop more accurate incident detection models.
4. Develop and test different persistence tests or algorithms to further reduce the FAR and improve the reliability of the models.

5. Investigate the potential of utilizing incident detection models in combination with other traffic management strategies such as ramp metering, variable speed limits, and lane control to optimize traffic flow and reduce congestion.
6. Collaborate with transportation agencies to evaluate the practicality and feasibility of implementing incident detection models in real-time traffic management systems.
7. Conduct a cost-benefit analysis to determine the economic feasibility of implementing the developed incident detection models in real-world scenarios.
8. Utilizing emerging technologies such as autonomous and connected vehicles to have live feedback of the traffic status to increase safety on the road and ensure smooth traffic flow. Autonomous and connected vehicles can provide real-time traffic data, which can be used to enhance incident detection and response

systems. In addition, ML models can be used to analyze data and identify patterns and anomalies that could indicate incidents or road hazards.

9. The results of this dissertation suggest that minor incidents during low traffic volume may not be detected or may take longer to detect if they don't have a significant impact on the traffic flow. Therefore, it is recommended that TMCs can depend on phone calls from road users or people involved in the incident when they call the police station or recovery service. TMCs should also consider implementing a public reporting system, where road users can report incidents or road hazards directly to the TMC. This can help in the detection of minor incidents, especially during low traffic volume periods.

The results of this dissertation revealed that as the distance between the upstream and downstream stations decreased, the detectability of the model increased, and the detection time of incidents decreased. At 500-meter distance between the stations, some minor incidents at low traffic volume were detected, and the detection time was shorter than for larger distances such as 1000 and 1500 meters. Therefore, it is recommended to keep the distance between the upstream and downstream stations to be as short as possible, with the constraint on the available budget.

References

[1] S. Kamran and O. Haas, “A Multilevel Traffic Incidents Detection Approach: Identifying Traffic Patterns and Vehicle Behaviours using real-time GPS data,” in *2007 IEEE Intelligent Vehicles Symposium*, IEEE, Jun. 2007, pp. 912–917. doi: 10.1109/IVS.2007.4290233.

[2] D. Srinivasan, X. Jin, and R. L. Cheu, “Evaluation of Adaptive Neural Network Models for Freeway Incident Detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 1–11, Mar. 2004, doi: 10.1109/TITS.2004.825084.

[3] M. Saini, “Survey On Vision Based On-Road Vehicle Detection,” *Int. J. u- e-Service, Sci. Technol.*, vol. 7, no. 4, pp. 139–146, Aug. 2014, doi: 10.14257/ijunesst.2014.7.4.14.

[4] P. B. Farradyne, “Traffic Incident Management,” in *Encyclopedia of Transportation: Social Science and Policy*, 2455 Teller Road, Thousand Oaks California 91320 United States: SAGE Publications, Inc., 2014. doi: 10.4135/9781483346526.n479.

[5] V. L. Knoop, S. P. Hoogendoorn, and H. J. van Zuylen, “Capacity Reduction at Incidents: Empirical Data Collected from a Helicopter,” *Transp. Res. Rec.*, vol. 2071, no. 1, pp. 19–25, Jan. 2008, doi: 10.3141/2071-03.

[6] "Road traffic injuries," World Health Organization, 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (accessed Dec. 21, 2021).

[7] M. J. Tantillo, E. Roberts, and U. Mangar, "Roles of Transportation Management Centers in Incident Management on Managed Lanes," FHWA-HOP-14-022, Dec. 2014.

[8] X. Jin, Z. Zhang, and A. Gan, “Traffic Management Centers: Challenges, Best Practices, and Future Plans,” National Center for Transportation Systems Productivity and Management (US), 2014.

[9] J. Xiao and Y. Liu, “Traffic Incident Detection Using Multiple-Kernel Support Vector Machine,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 2324, no. 1, pp. 44–52, Jan. 2012, doi: 10.3141/2324-06.

[10] Z. Iqbal and M. I. Khan, “Automatic incident detection in smart city using multiple traffic flow parameters via V2X communication,” *Int. J. Distrib. Sens. Networks*, vol. 14, no. 11, Nov. 2018, doi: 10.1177/1550147718815845.

[11] Z. Iqbal, M. I. Khan, S. Hussain, and A. Habib, “An Efficient Traffic Incident Detection and Classification Framework by Leveraging the Efficacy of Model Stacking,” *Complexity*, vol. 2021, pp. 1–17, 2021, doi: 10.1155/2021/5543698.

[12] C. Allen R and D. E. Cleveland, “The Detection of Freeway Capacity Reducing Incidents by Traffic Stream Measurements,” TrS-1, Dec. 1970. doi: 10.1001/archsurg.138.12.1282.

[13] C. L. Dudek, C. J. Messer, and N. B. Nuckles, "Incident detection on urban freeways," Transp. Res. Rec., no. 495, pp. 12–24, 1974. [Online]. Available: http://onlinepubs.trb.org/Onlinepubs/trr/1974/495/495-002.pdf (accessed Dec.

21, 2021).

[14] C. L. Dudek, G. D. Weaver, G. P. Ritch, and C. J. Messer, "Detecting freeway incidents under low-volume conditions," vol. 553, pp. 34-47, Texas A & M University, 1975.

[15] H. Payne, “Freeway incident detection based upon pattern classification,” in *1975 IEEE Conference on Decision and Control including the 14th Symposium on Adaptive Processes*, IEEE, Dec. 1975, pp. 688–692. doi: 10.1109/CDC.1975.270592.

[16] H. J. Payne and S. Tignor, "Freeway Incident-Detection Algorithms Based on Decision Trees with States," Transp. Res. Rec., no. 682, pp. 30–37, 1978. (accessed Dec. 21, 2021).

[17] M. Levin and G. M. Krause, “Incident detection: A Bayesian approach,” *Transp. Res. Rec.*, no. 682, pp. 52–58, 1978, [Online]. Available: http://onlinepubs.trb.org/Onlinepubs/trr/1978/682/682-008.pdf. (accessed Dec. 21, 2021).

[18] E. Parkany and C. Xie, “A Complete Review of Incident Detection Algorithms & Their Deployment : What Works and What Doesn’t,” 2005.

[19] T. Fangming and D. Han, “Simulation of traffic incident detection based on VISSIM and neural network,” in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, IEEE, May 2012, pp. 51–55. doi: 10.1109/CSAE.2012.6272726.

[20] Y.-K. Ki, J.-H. Kim, T.-K. Kim, N.-W. Heo, J.-W. Choi, and J.-H. Jeong, “Method for Automatic Detection of Traffic Incidents Using Neural Networks and Traffic Data,” in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, Canada: IEEE, Nov. 2018, pp. 184–188. doi: 10.1109/IEMCON.2018.8615081.

[21] R. P. Roess, E. S. Prassas, and W. R. McShane, “Traffic Engineering,” in *ATM: the broadband telecommunications solution*, Fifth edit.NY NY SE - x, 785 pages : illustrations ; 26 cm: Institution of Engineering and Technology, 1993, pp. 132–147. doi: 10.1049/PBTE029E_ch12.

[22] L. Elefteriadou, *An Introduction to Traffic Flow Theory*, 17th ed., vol. 84. in Springer Optimization and Its Applications, vol. 84. New York, NY: Springer New York, 2014. doi: 10.1007/978-1-4614-8435-6.

[23] M. Motamed and R. Machemehl, “Real time freeway incident detection,” 2014.

[24] M. Motamed, “Developing a real-time freeway incident detection model using machine learning techniques,” University of Texas at Austin, 2016. doi: 10.15781/t28g8fj1f.

[25] T. Xie, Q. Shang, and Y. Yu, “Automated Traffic Incident Detection: Coping With Imbalanced and Small Datasets,” *IEEE Access*, vol. 10, pp. 35521–35540, 2022, doi: 10.1109/ACCESS.2022.3161835.

[26] L. 1964-T. A.-T. T.- Elefteriadou, *An introduction to traffic flow theory*, 17th ed. in Springer optimization and its applications ; volume 84. New York: Springer,

2014. doi: 10.1007/978-1-4614-8435-6 LK - https://aus.on.worldcat.org/oclc/863640178.

[27] O. ElSahly and A. Abdelfatah, "A Systematic Review of Traffic Incident Detection Algorithms," *Sustainability*, vol. 14, no. 22, p. 14859, Nov. 2022, doi: 10.3390/su142214859.

[28] M. S. Sheikh, J. Liang, and W. Wang, "An Improved Automatic Traffic Incident Detection Technique Using a Vehicle to Infrastructure Communication," *J. Adv. Transp.*, vol. 2020, 2020, doi: 10.1155/2020/9139074.

[29] R. Li, F. C. Pereira, and M. E. Ben-Akiva, "Overview of traffic incident duration analysis and prediction," *Eur. Transp. Res. Rev.*, vol. 10, no. 2, p. 22, Jun. 2018, doi: 10.1186/s12544-018-0300-1.

[30] D. Chimba, B. Kutela, G. Ogletree, F. Horne, and M. Tugwell, "Impact of Abandoned and Disabled Vehicles on Freeway Incident Duration," *J. Transp. Eng.*, vol. 140, no. 3, pp. 1–8, Mar. 2014, doi: 10.1061/(ASCE)TE.1943-5436.0000635.

[31] G. Valenti, M. Lelli, and D. Cucina, "A comparative study of models for the incident duration prediction," *Eur. Transp. Res. Rev.*, vol. 2, no. 2, pp. 103–111, 2010, doi: 10.1007/s12544-010-0031-4.

[32] H. Zhang and A. Khattak, "What Is the Role of Multiple Secondary Incidents in Traffic Operations?," *J. Transp. Eng.*, vol. 136, no. 11, pp. 986–997, Nov. 2010, doi: 10.1061/(ASCE)TE.1943-5436.0000164.

[33] *Highway Capacity Manual 7th Edition*. Washington, D.C.: National Academies Press, 2022. doi: 10.17226/26432.

[34] V. Shah, G. Hatcher, E. Greer, J. Fraser, M. Franz, and K. Sadabadi, *Guidelines for Quantifying Benefits of Traffic Incident Management Strategies*. National Academies Press, 2022. doi: 10.17226/26486.

[35] H. J. Haule, T. Sando, R. Lentz, C. H. Chuan, and P. Alluri, "Evaluating the impact and clearance duration of freeway incidents," *Int. J. Transp. Sci. Technol.*, vol. 8, no. 1, pp. 13–24, 2019, doi: 10.1016/j.ijtst.2018.06.005.

[36] F. C. Pereira, O. Lederman, and M. Ben-akiva, "Approximating incident occurrence time with a change-point latent variable framework," in *Transportation Research Board 93rd Annual Meeting. January 12-16, Washington, D.C.*, 2014.

[37] F. Yuan and R. L. Cheu, "Incident detection using support vector machines," *Transp. Res. Part C Emerg. Technol.*, vol. 11, no. 3–4, pp. 309–328, Jun. 2003, doi: 10.1016/S0968-090X(03)00020-2.

[38] L. Calderoni, D. Maio, and S. Rovis, "Deploying a network of smart cameras for traffic monitoring on a 'city kernel,'" *Expert Syst. Appl.*, vol. 41, no. 2, pp. 502–507, 2014, doi: 10.1016/j.eswa.2013.07.076.

[39] H. Y. Cheng, V. Gau, C. W. Huang, and J. N. Hwang, "Advanced formation and delivery of traffic information in intelligent transportation systems," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8356–8368, 2012, doi: 10.1016/j.eswa.2012.01.184.

[40] W. Wen, “An intelligent traffic management expert system with RFID technology,” *Expert Syst. Appl.*, vol. 37, no. 4, pp. 3024–3035, 2010, doi: 10.1016/j.eswa.2009.09.030.

[41] H. S. Mahmassani, “Evaluation of Incident Detection Methodologies (FHWA/TX-00/1795-1),” Federal Highway Administration, Texas, FHWA-OP-99-032, 1999.

[42] P. T. Martin, J. Perrin, B. Hansen, R. Kump, and D. Moore, “Incident Detection Algorithm Evaluation,” Minnesota Department of Transportation, MPC 01-122, 2001. [Online]. Available: http://www.lrrb.org/PDF/200112.pdf

[43] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-98074-4.

[44] Y. Ma and H. He, "Imbalanced Learning," Wiley, 2013, pp. 13-59. doi: 10.1002/9781118646106.

[45] S. Chen, W. Wang, and H. van Zuylen, “Construct support vector machine ensemble to detect traffic incident,” *Expert Syst. Appl.*, vol. 36, no. 8, pp. 10976–10986, 2009, doi: 10.1016/j.eswa.2009.02.039.

[46] P. T. Martin, J. Perrin, B. Hansen, R. Kump, and D. Moore, “Incident Detection Algorithm Evaluation,” 2001.

[47] H. S. Mahmassani, C. Haas, S. Zhou, and J. Peterman, “Evaluation of Incident Detection Methodologies,” 1999.

[48] K. Hamad and C. Quiroga, “Geovisualization of Archived ITS Data-Case Studies,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 104–112, 2016, doi: 10.1109/TITS.2015.2460995.

[49] R. L. Cheu, “Neural Network Models for Automated Detection of Lane-Blocking Incidents on Freeways,” 1998.

[50] L. Ahuja, “Automatic Incident Detection (AID),” Iowa State University, 2018.

[51] P. Chakraborty *et al.*, “OUTLIER MINING BASED TRAFFIC INCIDENT DETECTION USING BIG DATA ANALYTICS,” in *96th Annual Meeting Transportation Research Board*, Washington DC, USA, Nov. 2016, pp. 8–12.

[52] K. Ozbay and P. Kachroo, "Incident management in intelligent transportation systems," Artech House Publishers, 1999. [Online]. Available: https://digitalscholarship.unlv.edu/ece_fac_articles/103/ (accessed Dec. 25, 2021).

[53] M. Karatsoli, M. Margreiter, and M. Spangler, “Bluetooth-based travel times for automatic incident detection – A systematic description of the characteristics for traffic management purposes,” *Transp. Res. Procedia*, vol. 24, pp. 204–211, 2017, doi: 10.1016/j.trpro.2017.05.109.

[54] R. L. Cheu and S. G. Ritchie, “Automated detection of lane-blocking freeway incidents using artificial neural networks,” *Transp. Res. Part C*, vol. 3, no. 6, pp. 371–388, 1995, doi: 10.1016/0968-090X(95)00016-C.

[55] J. F. Collins, C. M. Hopkins, and J. A. Martin, *Automatic incident detection : TRRL algorithms HIOCC and PATREG*. Crowthorne: Transport and Road Research Laboratory., 1979.

[56] P. H. Masters, J. K. Lam, and Kam Wong, “Incident detection algorithms for COMPASS - An advanced traffic management system,” in *Vehicle Navigation and Information Systems Conference, 1991*, IEEE, Oct. 1991, pp. 295–310. doi: 10.1109/VNIS.1991.205776.

[57] Q. Liu, E. Chung, and L. Zhai, “Fusing moving average model and stationary wavelet decomposition for automatic incident detection: case study of Tokyo Expressway,” *J. Traffic Transp. Eng. (English Ed.*, vol. 1, no. 6, pp. 404–414, 2014, doi: 10.1016/S2095-7564(15)30290-7.

[58] G. Bakioğlu *et al.*, “Incident Detection Algorithms : A Literature Review,” in *1st IRF Europe & Central Asia Regional Congress & Exhibition*, Istanbul, Turkey, 2015.

[59] B. B. Lyall, “Performance Evaluation of the McMaster Incident Detection Algorithm,” 1991.

[60] S. Cohen and Z. Ketselidou, “A Calibration Process for Automatic Incident Detection Algorithms,” in *the 4th International Conference on Microcomputers in Transportation*, Baltimore, MD, USA, 1993.

[61] B. Abdulhai and H. T. Abdelwahab, “Comparison of three incident detection algorithms using detailed simulation results,” *J. Transp. Eng.*, vol. 127, no. 3, pp. 251–259, 2001.

[62] Y. J. Stephanedes and J. Hourdakis, “Comparison of real-time traffic incident detection algorithms,” *Transp. Res. Rec.*, vol. 1554, pp. 44–51, 1996.

[63] M. Levin and G. M. Krause, “Incident-Detection Algorithms Part 1. Off-Line Evaluation,” *Transp. Res. Rec.*, vol. 722, pp. 49–58, 1979.

[64] L. M, K. GM, and B. JA, “Incident-detection algorithms. part 2. on-line evaluation,” *Transp. Res. Rec.*, vol. 722, pp. 58–64, 1979.

[65] P. H. Masters, J. K. Lam, and K. Wong, “Incident Detection Algorithms for COMPASS - An Advanced Traffic Management System,” IEEE, 1991.

[66] P. Athol, “Interdependence of Certain Operational Characteristics within a Moving Traffic Stream,” *Highw. Res. Rec.*, vol. 72, pp. 58–87, 1965, [Online]. Available: http://trid.trb.org/view.aspx?id=117031 (accessed Jan. 5, 2021).

[67] B. N. Persaud and F. L. Hall, “Catastrophe theory and patterns in 30-second freeway traffic data— Implications for incident detection,” *Transp. Res. Part A Gen.*, vol. 23, no. 2, pp. 103–113, 1989, doi: https://doi.org/10.1016/0191-2607(89)90071-X.

[68] B. N. Persaud, F. L. Hall, and L. M. Hall, “Congestion Identification Aspects of the Mcmaster Incident Detection Algorithm,” *Transp. Res. Rec.*, no. 1287, 1990, [Online]. Available: https://trid.trb.org/view.aspx?id=352877

[69] K. N. Balke, “An evaluation of existing incident detection algorithms,”

FHWA/TX-93/1232-20, 1993. [Online]. Available: http://trid.trb.org/view.aspx?id=1170418 (accessed Jan. 7, 2021).

[70] Y. J. Stephanedes and A. P. Chassiakos, "Application of Filtering Techniques for Incident Detection," *J. Transp. Eng.*, vol. 119, no. 1, pp. 13–26, Jan. 1993, doi: 10.1061/(ASCE)0733-947X(1993)119:1(13).

[71] Y. J. Stephanedes and A. P. Chassiakos, "Application of Filtering Techniques for Incident Detection," *J. Transp. Eng.*, vol. 119, no. 1, pp. 13–26, Jan. 1993, doi: 10.1061/(ASCE)0733-947X(1993)119:1(13).

[72] M. Margreiter, M. Spangler, T. Zeh, and C. Carstensen, "Bluetooth-Measured Travel Times for Dynamic Re-Routing," no. November, 2015, doi: 10.5176/2301-394x_ace15.04.

[73] M. Karatsoli, M. Margreiter, and M. Spangler, "Bluetooth-based travel times for automatic incident detection – A systematic description of the characteristics for traffic management purposes," *Transp. Res. Procedia*, vol. 24, pp. 204–211, 2017, doi: 10.1016/j.trpro.2017.05.109.

[74] U. Khalil, T. Javid, and A. Nasir, "Automatic road accident detection techniques: A brief survey," in *2017 International Symposium on Wireless Systems and Networks (ISWSN)*, Lahore, Pakistan: IEEE, Nov. 2017, pp. 1–6. doi: 10.1109/ISWSN.2017.8250025.

[75] "SUMO User Documentation," [Online]. Available: https://sumo.dlr.de/docs/index.html (accessed Dec. 15, 2022).

[76] E. D'Andrea and F. Marcelloni, "Detection of traffic congestion and incidents from GPS trace analysis," *Expert Syst. Appl.*, vol. 73, pp. 43–56, May 2017, doi: 10.1016/j.eswa.2016.12.018.

[77] X. Han, T. Grubenmann, R. Cheng, S. C. Wong, X. Li, and W. Sun, "Traffic incident detection: A trajectory-based approach," *Proc. - Int. Conf. Data Eng.*, vol. 2020-April, pp. 1866–1869, 2020, doi: 10.1109/ICDE48307.2020.00190.

[78] A. B. Nikolaev, Y. S. Sapego, A. M. Ivakhnenko, T. E. Mel′nikova, and V. Y. Stroganov, "Analysis of the incident detection technologies and algorithms in intelligent transport systems," *Int. J. Appl. Eng. Res.*, vol. 12, no. 15, pp. 4765–4774, 2017.

[79] Zehang Sun, G. Bebis, and R. Miller, "On-road vehicle detection: a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 5, pp. 694–711, May 2006, doi: 10.1109/TPAMI.2006.104.

[80] J. A. Khoury, C. T. Haas, H. Mahmassani, H. Logman, and T. Rioux, "Performance Comparison of Automatic Vehicle Identification and Inductive Loop Traffic Detectors for Incident Detection," *J. Transp. Eng.*, vol. 129, no. 6, pp. 600–607, Nov. 2003, doi: 10.1061/(ASCE)0733-947X(2003)129:6(600).

[81] M. Houbraken, S. Logghe, M. Schreuder, P. Audenaert, D. Colle, and M. Pickavet, "Automated Incident Detection Using Real-Time Floating Car Data," *J. Adv. Transp.*, vol. 2017, pp. 1–13, 2017, doi: 10.1155/2017/8241545.

[82] Z. Liang, H. Chen, Z. Song, Y. Zhou, and B. Zhang, "Traffic congestion incident

detection and dissipation algorithm for urban intersection based on FCD," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, IEEE, Dec. 2017, pp. 2578–2583. doi: 10.1109/CompComm.2017.8323001.

[83] Y. Asakura, T. Kusakabe, L. X. Nguyen, and T. Ushiki, "Incident detection methods using probe vehicles with on-board GPS equipment," *Transp. Res. Part C Emerg. Technol.*, vol. 81, pp. 330–341, Aug. 2017, doi: 10.1016/j.trc.2016.11.023.

[84] M. Fogue, P. Garrido, F. J. Martinez, J. C. Cano, C. T. Calafate, and P. Manzoni, "Automatic accident detection: Assistance through communication technologies and vehicles," *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 90–100, 2012. doi: 10.1109/MVT.2012.2203877.

[85] B. Fernandes, V. Gomes, J. Ferreira, and A. Oliveira, "Mobile Application for Automatic Accident Detection and Multimodal Alert," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, IEEE, May 2015, pp. 1–5. doi: 10.1109/VTCSpring.2015.7145935.

[86] R. Grzeszczyk, J. Merkisz, P. Bogus, and T. Kaminski, "Methods and procedures for testing the e-call in-vehicle unit for the purpose of its performance assessment and certification," in *21st International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, Stuttgart, Germany, 2009.

[87] R. Bossom *et al.*, "D31 European ITS Communication Architecture-Overall Framework- Proof of Concept Implementation," 2009.

[88] "Welcome to OnStar," [Online]. Available: https://www.onstar.com/us/en/home/ (accessed Apr. 15, 2021).

[89] M. Cesana, L. Fratta, M. Gerla, E. Giordano, and G. Pau, "C-VET the UCLA campus vehicular testbed: Integration of vanet and mesh networks," *2010 Eur. Wirel. Conf. EW 2010*, pp. 689–695, 2010, doi: 10.1109/EW.2010.5483535.

[90] V. Raosaheb Patil and S. Suresh Pardeshi, "Mechanism for accident detection, prevention and reporting system," *Mater. Today Proc.*, vol. 72, pp. 1975–1980, 2023, doi: 10.1016/j.matpr.2022.11.215.

[91] O. Deniz and H. B. Celikoglu, "Overview to some existing incident detection algorithms: a comparative evaluation," *Procedia Soc. Behav. Sci.*, vol. 00, pp. 1–13, 2011.

[92] J. Tsai and E. R. Case, "Development of freeway incident-detection algorithms by using pattern-recognition techniques," *Transp. Res. Rec.*, vol. 722, pp. 113–116, 1979.

[93] L. Ahuja, "Automatic incident detection," Iowa State University, 2018.

[94] F. Ahmed and Y. E. Hawas, "A Threshold-Based Real-Time Incident Detection System for Urban Traffic Networks," *Procedia - Soc. Behav. Sci.*, vol. 48, pp. 1713–1722, 2012, doi: 10.1016/j.sbspro.2012.06.1146.

[95] "Boson NetSim - Network Simulator for Cisco," [Online]. Available: https://www.boson.com/netsim-cisco-network-simulator (accessed Jan. 05,

2022).

[96] P. Chakraborty, A. Sharma, S. Knickerbocker, and J. Hess, “Outlier Mining Based Traffic Incident Detection Using Big Data Analytics,” in *96th Annual Meeting Transportation Research Board*, Washington DC, USA, Nov. 2016.

[97] P. Chakraborty, C. Hegde, and A. Sharma, “Data-driven parallelizable traffic incident detection using spatio-temporally denoised robust thresholds,” *Transp. Res. Part C Emerg. Technol.*, vol. 105, pp. 81–99, 2019, doi: https://doi.org/10.1016/j.trc.2019.05.034.

[98] Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation," Federal Highway Administration, Washington, D.C., 2005. [Online]. Available: https://purl.fdlp.gov/GPO/LPS69770 (accessed Jan. 10, 2022).

[99] A. R. Cook, “The Management of Urban Freeway Traffic,” *Can. J. Civ. Eng.*, vol. 1, no. 2, pp. 141–149, Dec. 1974, doi: 10.1139/l74-013.

[100] S. A. Ahmed and A. Cook, “Application of Time-Series Analysis Techniques to Freeway Incident Detection,” *Transp. Res. Rec.*, 1982.

[101] S. A. Ahmed and A. R. Cook, “Time Series Models for Freeway Incident Detection,” *Transp. Eng. J. ASCE*, vol. 106, no. 6, pp. 731–745, 1980.

[102] M. S. Ahmed and A. R. Cook, “Analysis of Freeway Traffic Time-Series Data by Using Box-Jenkins Techniques.,” *Transp. Res. Rec.*, no. 722, pp. 1–9, 1979.

[103] H. Li, S. Li, H. Zhu, X. Zhao, and X. Zhang, “Automated Detection Algorithm for Traffic Incident in Urban Expressway Based on Lengthways Time Series,” in *Green Intelligent Transportation Systems*, Springer Singapore, 2019, pp. 625–633. doi: 10.1007/978-981-13-0302-9_61.

[104] Xin Jin, D. Srinivasan, and Ruey Long Cheu, “Classification of freeway traffic patterns for incident detection using constructive probabilistic neural networks,” *IEEE Trans. Neural Networks*, vol. 12, no. 5, pp. 1173–1187, 2001, doi: 10.1109/72.950145.

[105] S. Suthaharan, *Machine Learning Models and Algorithms for Big Data Classification*, vol. 36. in Integrated Series in Information Systems, vol. 36. Boston, MA: Springer US, 2016. doi: 10.1007/978-1-4899-7641-3.

[106] M. Rusyaidi and Z. Ibrahim, “A Review: An Evaluation of Current Artificial Intelligent Methods in Traffic Flow Prediction,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 917, no. 1, p. 012063, Sep. 2020, doi: 10.1088/1757-899X/917/1/012063.

[107] O. I. Olayode, L. K. Tartibu, and M. O. Okwu, “Application of Artificial Intelligence in Traffic Control System of Non-autonomous Vehicles at Signalized Road Intersection.,” in *Procedia CIRP*, 2020, pp. 194–200. doi: 10.1016/j.procir.2020.02.167.

[108] S. A. Gamel, A. I. Saleh, and H. A. Ali, “Machine learning-based traffic management techniques for intelligent transportation system : Review,” *Nile J. Commun. Comput. Sci.*, vol. 1, no. 1, pp. 9–18, 2021.

[109] M. Nama *et al.*, “Machine learning-based traffic scheduling techniques for intelligent transportation system: Opportunities and challenges,” *Int. J. Commun. Syst.*, vol. 34, no. 9, Jun. 2021, doi: 10.1002/dac.4814.

[110] R. Šusteková, R. Knutelská, and M. Phd, “How is the artificial intelligence used in applications for traffic management,” in *Scientific Proceedings Xxiii International Scientific-Technical Conference "trans & MOTAUTO ’15"*, VARNA, BULGARIA, 2015, pp. 49–52. [Online]. Available: https://stumejournals.com/journals/mtm/2015/10/49.full.pdf

[111] T. Yuan, W. Rocha Neto, C. E. Rothenberg, K. Obraczka, C. Barakat, and T. Turletti, “Machine learning for next-generation intelligent transportation systems: A survey,” *Trans. Emerg. Telecommun. Technol.*, vol. 33, no. 4, pp. 1–35, Apr. 2022, doi: 10.1002/ett.4427.

[112] K. Hamad, M. A. Khalil, and A. R. Alozi, “Predicting Freeway Incident Duration Using Machine Learning,” *Int. J. Intell. Transp. Syst. Res.*, vol. 18, no. 2, pp. 367–380, May 2020, doi: 10.1007/s13177-019-00205-1.

[113] S. Olugbade, S. Ojo, A. L. Imoize, J. Isabona, and M. O. Alaba, “A Review of Artificial Intelligence and Machine Learning for Incident Detectors in Road Transport Systems,” *Math. Comput. Appl.*, vol. 27, no. 5, p. 77, Sep. 2022, doi: 10.3390/mca27050077.

[114] S. Sharma, S. Harit, and J. Kaur, “Traffic Accident Detection Using Machine Learning Algorithms,” in *Proceedings of Third International Conference on Sustainable Computing*, Bosnia and Herzegovina: Springer Singapore, 2022, pp. 501–507. [Online]. Available: http://dx.doi.org/10.1007/978-981-16-4538-9_49

[115] D. Mani, P. Amrith, E. Umamaheswari, D. M. Ajay, and R. U. Anitha, “Smart detection of vehicle accidents using object identification sensors with artificial intelligent systems,” *Int. J. Recent Technol. Eng.*, vol. 7, no. 5, pp. 375–379, 2019.

[116] T. Huang, S. Wang, and A. Sharma, “Highway crash detection and risk estimation using deep learning,” *Accid. Anal. Prev.*, vol. 135, no. April 2019, 2020, doi: 10.1016/j.aap.2019.105392.

[117] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 79, no. 8, pp. 2554–2558, 1982, doi: 10.1073/pnas.79.8.2554.

[118] L. Wang, J. Zhao, and R. Mortier, “Neural Network,” in *Investopedia*, 2022, pp. 219–242. doi: 10.1007/978-3-030-97645-3_11.

[119] "What are Neural Networks?," IBM, 2020. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks (accessed Dec. 27, 2021).

[120] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.

[121] L. Hardesty, "Explained: Neural networks," MIT News | Massachusetts Institute of Technology. [Online]. Available: https://news.mit.edu/2017/explained-

neural-networks-deep-learning-0414 (accessed Dec. 27, 2021).

[122] R. L. Cheu, S. G. Ritchie, W. W. Recker, and B. Bavarian, "Investigation of a neural network model for freeway incident detection," in *Artificial Intelligence and Civil Engineering*, 1991, pp. 267–274.

[123] X. Jin, R. L. Cheu, and D. Srinivasan, "Development and adaptation of constructive probabilistic neural network in freeway incident detection," *Transp. Res. Part C Emerg. Technol.*, vol. 10, no. 2, pp. 121–147, 2002, doi: 10.1016/S0968-090X(01)00007-9.

[124] H. Dia and G. Rose, "Development and evaluation of neural network freeway incident detection models using field data," *Transp. Res. Part C Emerg. Technol.*, vol. 5, no. 5, pp. 313–331, 1997, doi: 10.1016/S0968-090X(97)00016-8.

[125] B. Abdulhai and S. G. Ritchie, "Enhancing the universality and transferability of freeway incident detection using a Bayesian-based neural network," *Transp. Res. Part C Emerg. Technol.*, vol. 7, no. 5, pp. 261–280, 1999, doi: https://doi.org/10.1016/S0968-090X(99)00022-4.

[126] V. V. Zyryanov, "Incidents detection on city roads," in *IOP Conference Series: Materials Science and Engineering*, 2020. doi: 10.1088/1757-899X/913/4/042065.

[127] G. Gupta, R. Singh, A. Singh Patel, and M. Ojha, "Accident Detection Using Time-Distributed Model in Videos," in *Proceedings of Fifth International Congress on Information and Communication*, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., Singapore: Springer Singapore, 2021, pp. 214–223.

[128] L. Li, Y. Lin, B. Du, F. Yang, and B. Ran, "Real-time traffic incident detection based on a hybrid deep learning model," *Transp. A Transp. Sci.*, vol. 18, no. 1, pp. 78–98, Mar. 2022, doi: 10.1080/23249935.2020.1813214.

[129] Y. Lin, L. Li, H. Jing, B. Ran, and D. Sun, "Automated traffic incident detection with a smaller dataset based on generative adversarial networks," *Accid. Anal. Prev.*, vol. 144, pp. 1–9, 2020, doi: 10.1016/j.aap.2020.105628.

[130] A. O. Philip and R. K. Saravanaguru, "Multisource traffic incident reporting and evidence management in Internet of Vehicles using machine learning and blockchain," *Eng. Appl. Artif. Intell.*, vol. 117, no. November 2022, p. 105630, Jan. 2023, doi: 10.1016/j.engappai.2022.105630.

[131] I. Katsamenis *et al.*, "TraCon: A Novel Dataset for Real-Time Traffic Cones Detection Using Deep Learning," in *Novel & Intelligent Digital Systems: Proceedings of the 2nd International Conference (NiDS 2022)*, A. Krouska, C. Troussas, and J. Caro, Eds., in Lecture Notes in Networks and Systems, vol. 556. Athens, Greece: Springer International Publishing, 2023, pp. 382–391. doi: 10.1007/978-3-031-17601-2_37.

[132] L. A. Zadeh, "Fuzzy algorithms," *Inf. Control*, vol. 12, no. 2, pp. 94–102, 1968, doi: 10.1016/S0019-9958(68)90211-8.

[133] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, 1965, doi:

10.1016/S0019-9958(65)90241-X.

[134] L. A. Zadeh, “Fuzzy logic = computing with words,” *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 2, pp. 103–111, 1996, doi: 10.1109/91.493904.

[135] G. J. Klir, “Fuzzy logic,” *IEEE Potentials*, vol. 14, no. 4, pp. 10–15, 1995, doi: 10.1109/45.468220.

[136] J. M. Mendel, “Fuzzy Logic Systems for Engineering: A Tutorial,” *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, 1995, doi: 10.1109/5.364485.

[137] P. Albertos, A. Sala, and M. Olivares, “Fuzzy logic controllers. Methodology. Advantages and drawbacks,” in *Congreso Español sobre Tecnologías y Lógica Fuzzy*, 2000.

[138] E. Hüllermeier, “Does machine learning need fuzzy logic?,” *Fuzzy Sets Syst.*, vol. 281, pp. 292–299, 2015, doi: 10.1016/j.fss.2015.09.001.

[139] J. Jantzen, *Foundations of Fuzzy Control*. Wiley, 2013. doi: 10.1002/9781118535608.

[140] Y. Bai and D. Wang, “Fundamentals of Fuzzy Logic Control — Fuzzy Sets, Fuzzy Rules and Defuzzifications,” in *Advanced Fuzzy Logic Technologies in Industrial Applications*, Y. Bai, H. Zhuang, and D. Wang, Eds., London: Springer London, 2006, pp. 17–36. doi: 10.1007/978-1-84628-469-4_2.

[141] N. Nedjah and L. de Macedo Mourelle, *Fuzzy systems engineering: theory and practice*, vol. 181. Springer Science & Business Media, 2005.

[142] T. Floor and O. Greenwich, "WorldQuant Perspectives Reshaping the World with Fuzzy Logic Reshaping the World with Fuzzy Logic," 2018. [Online]. Available: www.weareworldquant.com (accessed Feb. 22, 2022).

[143] R. Rossi, M. Gastaldi, G. Gecchele, and V. Barbaro, “Fuzzy Logic-based Incident Detection System using Loop Detectors Data,” *Transp. Res. Procedia*, vol. 10, no. July, pp. 266–275, 2015, doi: 10.1016/j.trpro.2015.09.076.

[144] J. A. Hartigan and M. A. Wong, “Algorithm AS 136: A K-Means Clustering Algorithm,” *Appl. Stat.*, vol. 28, no. 1, p. 100, Mar. 1979, doi: 10.2307/2346830.

[145] F. Ahmed and Y. E. Hawas, “A fuzzy logic model for real-time incident detection in urban road network,” in *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence*, Vienna, Austria, 2013, pp. 465–472. doi: 10.5220/0004239904650472.

[146] F. W. F. Mustafa, “An Application of Fuzzy Logic In Urban Traffic Incident Detection,” United Arab Emirates University, 2015.

[147] S. Lee, R. A. Krammes, and J. Yen, “Fuzzy-logic-based incident detection for signalized diamond interchanges,” *Transp. Res. Part C Emerg. Technol.*, vol. 6, no. 5–6, pp. 359–377, 1998, doi: 10.1016/S0968-090X(99)00004-2.

[148] M. Bharath Kumar, A. Basit, M. B. Kiruba, R. Giridharan, and S. M. Keerthana, “Road Accident Detection Using Machine Learning,” in *2021 International Conference on System, Computation, Automation and Networking, ICSCAN 2021*, 2021. doi: 10.1109/ICSCAN53069.2021.9526546.

[149] N. Dogru and A. Subasi, “Traffic accident detection using random forest classifier,” in *2018 15th Learning and Technology Conference (L&T)*, Jeddah, Saudi Arabia: IEEE, Feb. 2018, pp. 40–45. doi: 10.1109/LT.2018.8368509.

[150] Yongchang Ma, M. Chowdhury, A. Sadek, and M. Jeihani, “Real-Time Highway Traffic Condition Assessment Framework Using Vehicle–Infrastructure Integration (VII) With Artificial Intelligence (AI),” *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 615–627, Dec. 2009, doi: 10.1109/TITS.2009.2026673.

[151] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “A practical guide to support vector classification,” University of National Taiwan, Taipei., Taipei, 2003.

[152] D. Ruppert, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," Second edi., vol. 99, no. 466, pp. 587-624, New York: Springer, 2004. doi: 10.1198/jasa.2004.s339.

[153] T. K. Ho, “Random decision forests,” in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 1995, pp. 278–282. doi: 10.1109/ICDAR.1995.598994.

[154] S. Suthaharan, “Support Vector Machine,” S. Suthaharan, Ed., Boston, MA: Springer US, 2016, pp. 207–235. doi: 10.1007/978-1-4899-7641-3_9.

[155] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[156] T. Yiu, "Understanding Random Forest," Towards Data Science, Jun. 12, 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 (accessed Aug. 03, 2022).

[157] "Random Forest Algorithm," 2022. [Online]. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/randomforest-algorithm?tag=random forest (accessed Aug. 03, 2022).

[158] Q. Liu, J. Lu, S. Chen, and K. Zhao, “Multiple Naïve Bayes Classifiers Ensemble for Traffic Incident Detection,” *Math. Probl. Eng.*, vol. 2014, pp. 1–16, 2014, doi: 10.1155/2014/383671.

[159] L. Bao, Q. Wang, W. Qu, and X. Mo, “Research on Highway Traffic Event Detection Method Based on Image Processing,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 791, no. 1, p. 012193, Jun. 2021, doi: 10.1088/1755-1315/791/1/012193.

[160] K. Saho, “Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design,” in *Kalman Filters - Theory for Advanced Applications*, InTech, 2018. doi: 10.5772/intechopen.71731.

[161] B. Ekstrand, “Some Aspects on Filter Design for Target Tracking,” *J. Control Sci. Eng.*, vol. 2012, pp. 1–15, 2012, doi: 10.1155/2012/870890.

[162] K. Saho and M. Masugi, “Automatic Parameter Setting Method for an Accurate Kalman Filter Tracker Using an Analytical Steady-State Performance Index,” *IEEE Access*, vol. 3, pp. 1919–1930, 2015, doi: 10.1109/ACCESS.2015.2486766.

[163] I. HASHLAMON and K. ERBATUR, “An improved real-time adaptive Kalman filter with recursive noise covariance updating rules,” *TURKISH J. Electr. Eng. Comput. Sci.*, vol. 24, no. 2, pp. 524–540, 2016, doi: 10.3906/elk-1309-60.

[164] J. Ren, Y. Chen, L. Xin, J. Shi, B. Li, and Y. Liu, “Detecting and positioning of traffic incidents via video-based analysis of traffic states in a road segment,” *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 428–437, 2016, doi: 10.1049/iet-its.2015.0022.

[165] J. P. Jesan, “The neural approach to pattern recognition,” *Ubiquity*, vol. 2004, no. April, pp. 2–2, 2004, doi: 10.1145/991108.985625.

[166] S. L. Min, “Evaluation of Adaptive Automatic Freeway Incident Detection Algorithms,” Malaysia University of Science and Technology, 2004.

[167] C. L. Mak and H. S. L. Fan, “Enhancement of Automatic Incident Detection Algorithms for Singapore’s Central Expressway,” *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1923, no. 1, pp. 144–152, Jan. 2005, doi: 10.1177/0361198105192300116.

[168] P. Group, “PTV Vissim 2022 User Manual.” PTV Group, 2022. [Online]. Available: https://vision-traffic.ptvgroup.com/en-us/products/ptv-vissim/

[169] T. Afrin and N. Yodo, “A Survey of Road Traffic Congestion Measures towards a Sustainable and Resilient Transportation System,” *Sustainability*, vol. 12, no. 11, p. 4660, Jun. 2020, doi: 10.3390/su12114660.

[170] Z. Li *et al.*, “Temporal-spatial dimension extension-based intersection control formulation for connected and autonomous vehicle systems,” *Transp. Res. Part C Emerg. Technol.*, vol. 104, no. May, pp. 234–248, Jul. 2019, doi: 10.1016/j.trc.2019.05.003.

[171] O. Elsahly and A. Abdelfatah, “Effects of Automated Vehicles on Freeway Traffic Performance,” *Wulfenia J.*, vol. 27, no. 2, pp. 83–107, 2020.

[172] V. Ngan, T. Sayed, and A. Abdelfatah, “Impacts of Various Parameters on Transit Signal Priority Effectiveness,” *J. Public Transp.*, vol. 7, no. 3, pp. 71–93, Sep. 2004, doi: 10.5038/2375-0901.7.3.4.

[173] E. Aria, J. Olstam, and C. Schwietering, “Investigation of Automated Vehicle Effects on Driver’s Behavior and Traffic Performance,” *Transp. Res. Procedia*, vol. 15, pp. 761–770, 2016, doi: 10.1016/j.trpro.2016.06.063.

[174] FDOT, "Traffic Analysis Handbook," no. March, 2014. [Online]. Available: https://fdotwww.blob.core.windows.net/sitefinity/docs/default-source/planning/systems/systems-management/document-repository/traffic-analysis/traffic-analysis-handbook_05-2021.pdf?sfvrsn=cecdd23b_2 (accessed Aug. 03, 2022).

[175] G. Perraki, C. Roncoli, I. Papamichail, and M. Papageorgiou, “Evaluation of a model predictive control framework for motorway traffic involving conventional and automated vehicles,” *Transp. Res. Part C Emerg. Technol.*, vol. 92, no. April, pp. 456–471, Jul. 2018, doi: 10.1016/j.trc.2018.05.002.

[176] T. Toledo and H. N. Koutsopoulos, “Statistical Validation of Traffic Simulation

Models," *Transp. Res. Rec. J. Transp. Res. Board*, vol. 1876, no. 1, pp. 142–150, Jan. 2004, doi: 10.3141/1876-15.

[177] M. M. Morando, Q. Tian, L. T. Truong, and H. L. Vu, "Studying the Safety Impact of Autonomous Vehicles Using Simulation-Based Surrogate Safety Measures," *J. Adv. Transp.*, vol. 2018, pp. 1–11, 2018, doi: 10.1155/2018/6135183.

[178] C. H. Spiegelman, E. S. Park, and L. R. Rilett, "Transportation Statistics and Microsimulation," Boca Raton [Florida], CRC Press, 2011. [Online]. Available: http://catalogue.bnf.fr/ark:/12148/cb42287553d (accessed Aug. 28, 2022).

[179] A. B. Nikolaev, Y. S. Sapego, A. N. Jakubovich, L. I. Berner, and A. M. Ivakhnenko, "Simulation of automatic incidents detection algorithm on the transport network," *Int. J. Environ. Sci. Educ.*, vol. 11, no. 16, pp. 9060–9078, 2016, doi: 11(16): 9060-9078.

[180] B. D. 1952- Ripley, *Pattern recognition and neural networks*. Cambridge ; SE - xi, 403 pages : illustrations ; 26 cm: Cambridge University Press, 1996. [Online]. Available: http://catdir.loc.gov/catdir/toc/cam027/95025223.html

[181] G. (Gareth M. James, D. Witten, T. Hastie, and R. Tibshirani), "An Introduction to Statistical Learning: With Applications in R," vol. 103, New York, Springer, 2013, pp. 175-197. doi: 10.1007/978-1-4614-7138-7.

[182] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, NV-1 onl. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-6849-3.

[183] I. Baturynska and K. Martinsen, "Prediction of geometry deviations in additive manufactured parts: comparison of linear regression with machine learning algorithms," *J. Intell. Manuf.*, vol. 32, no. 1, pp. 179–200, Jan. 2021, doi: 10.1007/s10845-020-01567-0.

[184] T. Agrawal, *Hyperparameter Optimization in Machine Learning*, 1st ed. 20. Berkeley, CA: Apress, 2021. doi: 10.1007/978-1-4842-6579-6.

[185] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012.

[186] H. Ishwaran, "The effect of splitting on random forests," *Mach. Learn.*, vol. 99, no. 1, pp. 75–118, Apr. 2015, doi: 10.1007/s10994-014-5451-2.

[187] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.

[188] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering*. Auerbach Publications, 2006. doi: 10.1201/9781420013061.

[189] S. Shanmuganathan, *Artificial Neural Network Modelling*, vol. 628. in Studies in Computational Intelligence, vol. 628. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-28495-8.

[190] S. S. Haykin, *Neural networks : a comprehensive foundation*, 2nd ed. Upper Saddle River, N.J. SE - xxi, 842 pages : illustrations ; 25 cm: Prentice Hall, 1994.

[191] M. H. SAZLI, “A brief review of feed-forward neural networks,” *Commun. Fac. Sci. Univ. Ankara*, vol. 50, no. May, pp. 11–17, 2006, doi: 10.1501/0003168.

[192] W. A. Gardner, “Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique,” *Signal Processing*, vol. 6, no. 2, pp. 113–133, Apr. 1984, doi: 10.1016/0165-1684(84)90013-6.

[193] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning : with Applications in R*, vol. 2nd ed. 20. New York, NY: Springer US, 2013. doi: 10.1007/978-1-0716-1418-1.

[194] K. E. Koech, "Cross-Entropy Loss Function," Towards Data Science, 2020. [Online]. Available: https://towardsdatascience.com/cross-entropy-loss-functionf38c4ec8643e#:~:text=than%20in%202.-,Cross-Entropy%20Loss%20Function,from%20the%20actual%20expected%20value. (accessed Jan. 09, 2021).

[195] J. Shore and R. Johnson, “Properties of cross-entropy minimization,” *IEEE Trans. Inf. Theory*, vol. 27, no. 4, pp. 472–482, Jul. 1981, doi: 10.1109/TIT.1981.1056373.

[196] D. E. RUMELHART, G. E. HINTON, and R. J. WILLIAMS, “Learning Internal Representations by Error Propagation,” in *Readings in Cognitive Science*, Elsevier, 1988, pp. 399–421. doi: 10.1016/B978-1-4832-1446-7.50035-2.

[197] A. Sarangam, "Epoch in Machine Learning: A Simple Introduction (2021)," Jigsaw Academy, 2021. [Online]. Available: https://www.jigsawacademy.com/blogs/ai-ml/epochin-machine-learning# (accessed Jan. 07, 2022).

[198] C. M. T. A.-T. T.- Bishop, "Pattern Recognition and Machine Learning," NV-1 online, New York, Springer, 2006. [Online]. Available: http://resolver.scholarsportal.info/isbn/9780387455280 (accessed Jan. 07, 2022).

[199] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning (Adaptive Computation and Machine Learning series)," Cambridge, MIT Press, 2016, pp. 164-167.

[200] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems*, Second edi. Sebastopol, CA SE - xxv, 819 pages : illustrations (some color) ; 24 cm: O’Reilly Media, Inc., 2019.

[201] I. H. (Ian H. . Witten, E. Frank, M. A. (Mark A. Hall, and C. J. T. A.-T. T.- Pal, *Data Mining*, Fourth edi. in Morgan Kaufmann Series in Data Management Systems. Cambridge, MA, United States: Elsevier, 2017. doi: 10.1016/C2015-0-02071-8.

[202] E. Alpaydd, *Introduction to machine learning*, 2nd editio. in Adaptive computation and machine learning series. Cambridge, Massachusetts: MIT Press, 2009. [Online]. Available: https://ieeexplore.ieee.org/xpl/bkabstractplus.jsp?bkn=6267367 (accessed Jan. 07, 2022).

[203] T. Hastie, R. Tibshirani, and J. H. (Jerome H. . Friedman, *The elements of statistical learning : data mining, inference, and prediction*, Second edi. New York SE - xxii, 745 pages : illustrations (some color), charts ; 24 cm.: Springer, 2009. [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/ (accessed Jan. 07, 2022).

[204] Y. S. 1957- Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data : a short course*. [United States] SE - xii, 201 pages : illustrations (chiefly color) ; 26 cm: AMLBook.com, 2012.

[205] K. P. 1970-T. A.-T. T.- Murphy, *Machine learning : a probabilistic perspective*, NV-1 onl. in Adaptive computation and machine learning series. Cambridge, Mass.: MIT Press, 2012. [Online]. Available: http://site.ebrary.com/id/10597102 (accessed Jan. 11, 2022).

[206] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, Massachusetts SE - xxii, 775 pages : illustrations (some color) ; 24 cm.: The MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org/(accessed Jan. 07, 2022).

Appendix A: Results of Hyperparameters Optimization for Random Forest Model

Iteration 0:

Number of trees: 100, Criterion: information gain, Maximal depth: 30, guess subset ratio: true, pruning: False, pre-pruning: False.

Given these parameters the confusion matrix of the Cross-Validation of the initial iteration is shown in the table below.

Table A 1 Confusion matrix of iteration 0

	true normal	true incident	class precision
pred. normal	10455	592	94.64%
pred. incident	105	3368	96.98%
class recall	99.01%	85.05%	
accuracy	95.2%	F-score	90.62

Iteration 1:

Number of trees: 250, Criterion: information gain, Maximal depth: 30, guess subset ratio: true, pruning: False, pre-pruning: False.

Table A 2 Evaluation steps of iteration 1

Evaluation step	Number of trees	Accuracy	Precision	Recall	F-score
1	100	0.9519284	0.96977425	0.850253	0.906059
2	110	0.9525482	0.97038637	0.85202	0.907331
3	120	0.9525482	0.97012561	0.852273	0.90735
4	130	0.9528237	0.97042655	0.85303	0.907912
5	140	0.9524105	0.96984095	0.85202	0.907081
6	150	0.9524105	0.97010913	0.851768	0.907057
7	160	0.9525482	0.97066892	0.851768	0.907299
8	170	0.952686	0.97068089	0.852273	0.907598
9	180	0.9529614	0.97123523	0.852778	0.908132
10	190	0.9526171	0.9706647	0.85202	0.907457
11	200	0.9524793	0.97064776	0.851515	0.907161
12	210	0.9528237	0.97122716	0.852273	0.907841
13	220	0.9528237	0.97094991	0.852525	0.907865
14	230	0.9527548	0.97121522	0.85202	0.907694
15	240	0.9528926	0.97150159	0.852273	0.907961

16	250	0.9530303	0.97179118	0.852525	0.908232
17	260	0.9528926	0.97177763	0.85202	0.907937
18	270	0.9528926	0.9717781	0.85202	0.907941
19	280	0.952686	0.97148229	0.851515	0.907524
20	290	0.952686	0.97067347	0.852273	0.907599
21	300	0.9526171	0.97066313	0.85202	0.907451
22	310	0.9526171	0.97120259	0.851515	0.9074
23	320	0.9526171	0.97093886	0.851768	0.907425
24	330	0.9527548	0.97150396	0.851768	0.907663
25	340	0.9526171	0.97093661	0.851768	0.907419
26	350	0.9527548	0.97149964	0.851768	0.907664
27	360	0.9527548	0.97150027	0.851768	0.907666
28	370	0.9528237	0.97150728	0.85202	0.907812
29	380	0.9527548	0.9712237	0.85202	0.907689
30	390	0.952686	0.97148548	0.851515	0.907518
31	400	0.9527548	0.97149365	0.851768	0.907664
32	410	0.9525482	0.97120127	0.851263	0.907249
33	420	0.9528926	0.97178059	0.85202	0.907932
34	430	0.9524793	0.97119383	0.85101	0.907102
35	440	0.9524793	0.97119383	0.85101	0.907102
36	450	0.9525482	0.97120562	0.851263	0.907252
37	460	0.9526171	0.97093785	0.851768	0.907423
38	470	0.9528237	0.97123535	0.852273	0.907838
39	480	0.9526171	0.97121291	0.851515	0.907397
40	490	0.9525482	0.97147455	0.85101	0.907225
41	500	0.9525482	0.97147455	0.85101	0.907225

Table A 3 Confusion matrix of iteration 1

	true normal	true incident	class precision
pred. normal	10462	584	94.71%
pred. incident	98	3376	97.18%
class recall	99.07%	85.25%	
accuracy	95.3%	F-score	90.82

<u>Iteration 2:</u>

Number of trees: 250, Criterion: information gain, Maximal depth: 29, guess subset ratio: true, pruning: False, pre-pruning: False.

Table A 4 Evaluation steps of iteration 2

Evaluation step	Depth	Accuracy	Precision	Recall	F-score
1	25	0.9530303	0.9720654	0.852273	0.908206
2	27	0.9529614	0.97178274	0.852273	0.908084
3	29	0.9530303	0.97179118	0.852525	0.908232
4	31	0.9530303	0.97179118	0.852525	0.908232
5	33	0.9530303	0.97179118	0.852525	0.908232
6	34	0.9530303	0.97179118	0.852525	0.908232
7	36	0.9530303	0.97179118	0.852525	0.908232
8	38	0.9530303	0.97179118	0.852525	0.908232
9	40	0.9530303	0.97179118	0.852525	0.908232
10	42	0.9530303	0.97179118	0.852525	0.908232
11	44	0.9530303	0.97179118	0.852525	0.908232
12	46	0.9530303	0.97179118	0.852525	0.908232
13	48	0.9530303	0.97179118	0.852525	0.908232
14	49	0.9530303	0.97179118	0.852525	0.908232
15	51	0.9530303	0.97179118	0.852525	0.908232
16	53	0.9530303	0.97179118	0.852525	0.908232
17	55	0.9530303	0.97179118	0.852525	0.908232
18	57	0.9530303	0.97179118	0.852525	0.908232
19	59	0.9530303	0.97179118	0.852525	0.908232
20	61	0.9530303	0.97179118	0.852525	0.908232
21	63	0.9530303	0.97179118	0.852525	0.908232
22	64	0.9530303	0.97179118	0.852525	0.908232
23	66	0.9530303	0.97179118	0.852525	0.908232
24	68	0.9530303	0.97179118	0.852525	0.908232
25	70	0.9530303	0.97179118	0.852525	0.908232
26	72	0.9530303	0.97179118	0.852525	0.908232
27	74	0.9530303	0.97179118	0.852525	0.908232
28	76	0.9530303	0.97179118	0.852525	0.908232
29	78	0.9530303	0.97179118	0.852525	0.908232
30	79	0.9530303	0.97179118	0.852525	0.908232
31	81	0.9530303	0.97179118	0.852525	0.908232
32	83	0.9530303	0.97179118	0.852525	0.908232
33	85	0.9530303	0.97179118	0.852525	0.908232
34	87	0.9530303	0.97179118	0.852525	0.908232
35	89	0.9530303	0.97179118	0.852525	0.908232
36	91	0.9530303	0.97179118	0.852525	0.908232
37	93	0.9530303	0.97179118	0.852525	0.908232
38	94	0.9530303	0.97179118	0.852525	0.908232
39	96	0.9530303	0.97179118	0.852525	0.908232
40	98	0.9530303	0.97179118	0.852525	0.908232
41	100	0.9530303	0.97179118	0.852525	0.908232

Table A 5 Confusion matrix of iteration 2

	true normal	true incident	class precision
pred. normal	10462	584	94.71%
pred. incident	98	3376	97.18%
class recall	99.07%	85.25%	
accuracy	95.3%	F-score	90.82

Iteration 3:

Number of trees: 250, Criterion: information gain, Maximal depth: 29, guess subset ratio: False, subset ratio: 0.95, pruning: False, pre-pruning: False.

Table A 6 Evaluation steps of iteration 3

Evaluation step	**Subset ratio**	**Accuracy**	**Precision**	**Recall**	**F-score**
1	0.05	0.9513085	0.9751799	0.842929	0.904212
2	0.1	0.9519284	0.9735831	0.846717	0.9057
3	0.15	0.9526171	0.9736559	0.849242	0.907185
4	0.2	0.9526171	0.9731062	0.849747	0.907233
5	0.25	0.952686	0.9720383	0.85101	0.907479
6	0.3	0.9529614	0.9718014	0.852273	0.908094
8	0.4	0.9523416	0.9690211	0.852525	0.907014
9	0.45	0.9528237	0.9693475	0.85404	0.908018
11	0.55	0.9525482	0.9685093	0.853788	0.907501
12	0.6	0.9526171	0.9679683	0.854545	0.907701
10	0.5	0.9524793	0.9684966	0.853535	0.907353
14	0.7	0.9524793	0.9668835	0.855051	0.907499
13	0.65	0.9523416	0.9663455	0.855051	0.907274
17	0.85	0.9525482	0.9663783	0.855808	0.907709
16	0.8	0.9527548	0.9666685	0.856313	0.908126
15	0.75	0.9527548	0.9671915	0.855808	0.90807
7	0.35	0.9522039	0.9692741	0.851768	0.9067
18	0.9	0.9527548	0.966145	0.856818	0.908174
20	1	0.9528237	0.9658898	0.857323	0.908345
19	0.95	0.9528926	0.966959	0.856566	0.908389

Table A 7 Confusion matrix of iteration 3

	true normal	true incident	class precision
pred. normal	10444	568	94.84%
pred. incident	116	3392	96.69%
class recall	98.90%	85.66%	
accuracy	**95.29%**	**F-score**	**90.84**

Iteration 4:

Number of trees: 250, Criterion: information gain, Maximal depth: 29, guess subset ratio: False, subset ratio: 0.95, pruning: true, Confidence: 8.00E-07 pre-pruning: False.

Table A 8 Evaluation steps of iteration 4

Evaluation step	Confidence	Accuracy	Precision	Recall	F-score
1	1.00E-07	0.9528237	0.9666818	0.856566	0.908267
2	2.00E-07	0.9528926	0.9666908	0.856818	0.90841
3	4.00E-07	0.9528926	0.9666911	0.856818	0.908413
4	6.00E-07	0.9528926	0.9666911	0.856818	0.908413
5	8.00E-07	0.9529614	0.9667001	0.857071	0.908556
6	1.00E-06	0.9528926	0.9666911	0.856818	0.908413
7	2.00E-06	0.9528926	0.9666908	0.856818	0.90841
8	4.00E-06	0.9528926	0.9666908	0.856818	0.90841
9	6.00E-06	0.9528926	0.9666908	0.856818	0.90841
10	8.00E-06	0.9528926	0.9666911	0.856818	0.908413
11	1.00E-05	0.9528926	0.9666911	0.856818	0.908413
12	2.00E-05	0.9528237	0.9666818	0.856566	0.908267
13	4.00E-05	0.9528926	0.9666911	0.856818	0.908413
14	6.00E-05	0.9529614	0.9667001	0.857071	0.908556
15	8.00E-05	0.9528926	0.9666908	0.856818	0.90841
16	1.00E-04	0.9528237	0.9666818	0.856566	0.908267
17	2.00E-04	0.9528237	0.9666818	0.856566	0.908267
18	4.00E-04	0.9528237	0.9666818	0.856566	0.908267
19	6.00E-04	0.9528237	0.9666818	0.856566	0.908267
20	8.00E-04	0.9528926	0.9666911	0.856818	0.908413
21	0.001	0.9528926	0.9666911	0.856818	0.908413
22	0.002	0.9529614	0.9667001	0.857071	0.908556
23	0.004	0.9528237	0.9666818	0.856566	0.908267
24	0.006	0.9528926	0.9666908	0.856818	0.90841
25	0.008	0.9528926	0.9666908	0.856818	0.90841

26	0.01	0.9528237	0.9666818	0.856566	0.908267
27	0.02	0.9528926	0.9666911	0.856818	0.908413
28	0.04	0.9528926	0.9666908	0.856818	0.90841
29	0.06	0.9527548	0.9663896	0.856566	0.908146
30	0.08	0.9528237	0.9663973	0.856818	0.908285
31	0.1	0.9524105	0.9660746	0.855556	0.90743
32	0.15	0.9528237	0.9666818	0.856566	0.908267
33	0.2	0.9529614	0.9667001	0.857071	0.908556
34	0.25	0.9528926	0.9666908	0.856818	0.90841
35	0.3	0.9528237	0.9663973	0.856818	0.908285
36	0.35	0.951584	0.9656884	0.852778	0.905701
37	0.4	0.9523416	0.9658048	0.855556	0.907312
38	0.45	0.9525482	0.9663589	0.855808	0.9077
39	0.5	0.9525482	0.9663589	0.855808	0.9077

Table A 9 Confusion matrix of iteration 4

	true normal	true incident	class precision
pred. normal	10443	566	94.86%
pred. incident	117	3394	96.67%
class recall	98.89%	85.71%	
accuracy	95.3%	F-score	90.86

Iteration 5:

Number of trees: 380, Criterion: information gain, Maximal depth: 29, guess subset ratio: False, subset ratio: 0.95, pruning: true, Confidence: 8.00E-07 pre-pruning: False.

Table A 10 Evaluation steps of iteration 5

Evaluation step	# of trees	Accuracy	Precision	Recall	F-score
1	100	0.9523416	0.965005	0.856313	0.907386
2	110	0.9524105	0.9650167	0.856566	0.907536
3	120	0.9528237	0.9656164	0.857576	0.908361
4	130	0.9527548	0.9656071	0.857323	0.908218
5	140	0.9528237	0.9658736	0.857323	0.908332
6	150	0.952686	0.9655939	0.857071	0.908065
7	160	0.952686	0.9655929	0.857071	0.908067
8	170	0.9525482	0.9653105	0.856818	0.907798
9	180	0.952686	0.9658647	0.856818	0.90804
10	190	0.9527548	0.9661371	0.856818	0.908165

11	200	0.952686	0.9661276	0.856566	0.908016
12	210	0.9528237	0.9664158	0.856818	0.908289
13	220	0.9527548	0.9661462	0.856818	0.908167
14	230	0.952686	0.9658756	0.856818	0.908045
15	240	0.9529614	0.9667001	0.857071	0.908556
16	250	0.9528237	0.9666818	0.856566	0.908267
17	260	0.9529614	0.9669679	0.856818	0.908533
18	270	0.9527548	0.9664122	0.856566	0.908143
19	280	0.9527548	0.9664171	0.856566	0.908148
20	290	0.9528926	0.9661649	0.857323	0.908464
21	300	0.9528926	0.9661594	0.857323	0.908464
22	310	0.9528926	0.9661527	0.857323	0.908461
23	320	0.9527548	0.9661313	0.856818	0.908166
24	330	0.9529614	0.9664368	0.857323	0.908579
25	340	0.9528237	0.9661456	0.857071	0.908311
26	350	0.9527548	0.9661317	0.856818	0.908163
27	360	0.9528237	0.9658852	0.857323	0.908335
28	370	0.9527548	0.9658714	0.857071	0.908185
29	380	0.9529614	0.9661568	0.857576	0.9086
30	390	0.9527548	0.9658697	0.857071	0.908189
31	400	0.9528237	0.9658799	0.857323	0.908333
32	410	0.9528237	0.9661474	0.857071	0.908309
33	420	0.9527548	0.9661393	0.856818	0.908164
34	430	0.952686	0.9663918	0.856313	0.907996
35	440	0.9527548	0.9664056	0.856566	0.908144
36	450	0.9528237	0.9664145	0.856818	0.908288
37	460	0.952686	0.9663966	0.856313	0.908001
38	470	0.9528237	0.9664131	0.856818	0.908292
39	480	0.952686	0.9663966	0.856313	0.908001
40	490	0.9526171	0.966135	0.856313	0.90788
41	500	0.9526171	0.966135	0.856313	0.90788

Table A 11 Confusion matrix of iteration 5

	true normal	true incident	class precision
pred. normal	10441	564	94.88%
pred. incident	119	3396	96.61%
class recall	98.87%	85.76%	
accuracy	95.3%	F-score	90.86

Iteration 6:

Number of trees: 380, Criterion: information gain, Maximal depth: 27, guess subset ratio: False, subset ratio: 0.95, pruning: true, Confidence: 8.00E-07 pre-pruning: False.

Table A 12 Evaluation steps of iteration 6

Evaluation step	Depth	Accuracy	Precision	Recall	F-score
1	25	0.9529614	0.9666897	0.857071	0.908548
2	27	0.9530303	0.9666978	0.857323	0.908694
3	29	0.9528926	0.9661478	0.857323	0.908457
4	31	0.9529614	0.9661568	0.857576	0.9086
5	33	0.9529614	0.9658832	0.857828	0.908625
6	34	0.9528237	0.9658663	0.857323	0.908334
7	36	0.9528926	0.9658753	0.857576	0.908477
8	38	0.9529614	0.9658832	0.857828	0.908625
9	40	0.9528926	0.9658753	0.857576	0.908477
10	42	0.9528926	0.9658743	0.857576	0.908481
11	44	0.9528237	0.9658663	0.857323	0.908334
12	46	0.9528926	0.9658743	0.857576	0.908481
13	48	0.9528237	0.9658663	0.857323	0.908334
14	49	0.9529614	0.9658832	0.857828	0.908625
15	51	0.9528926	0.9658753	0.857576	0.908477
16	53	0.9528926	0.9658743	0.857576	0.908481
17	55	0.9528926	0.9658743	0.857576	0.908481
18	57	0.9528926	0.9658743	0.857576	0.908481
19	59	0.9528237	0.9658663	0.857323	0.908334
20	61	0.9528237	0.9658663	0.857323	0.908334
21	63	0.9528237	0.9658663	0.857323	0.908334
22	64	0.9528926	0.9658753	0.857576	0.908477
23	66	0.9528926	0.9658743	0.857576	0.908481
24	68	0.9528926	0.9658743	0.857576	0.908481
25	70	0.9529614	0.9658832	0.857828	0.908625
26	72	0.9528926	0.9658743	0.857576	0.908481
27	74	0.9528926	0.9658753	0.857576	0.908477
28	76	0.9528237	0.9656046	0.857576	0.90836
29	78	0.9527548	0.9655967	0.857323	0.908212
30	79	0.9529614	0.9658832	0.857828	0.908625
31	81	0.9528237	0.965606	0.857576	0.908355
32	83	0.9528926	0.9658753	0.857576	0.908477
33	85	0.9528237	0.965606	0.857576	0.908355
34	87	0.9527548	0.9655967	0.857323	0.908212
35	89	0.9529614	0.9658832	0.857828	0.908625
36	91	0.9528926	0.9656139	0.857828	0.908503
37	93	0.9528926	0.9658753	0.857576	0.908477
38	94	0.9528926	0.9658753	0.857576	0.908477

39	96	0.9529614	0.9658832	0.857828	0.908625
40	98	0.9528237	0.9656046	0.857576	0.90836
41	100	0.9528926	0.9658743	0.857576	0.908481

Table A 13 Confusion matrix of iteration 6

	true normal	true incident	class precision
pred. normal	10443	565	94.87%
pred. incident	117	3395	96.67%
class recall	98.89%	85.73%	
accuracy	95.3%	F-score	90.87

Iteration 7:

Number of trees: 380, Criterion: information gain, Maximal depth: 27, guess subset ratio: False, subset ratio: 0.8, pruning: true, Confidence: 8.00E-07 pre-pruning: False.

Table A 14 Evaluation steps of iteration 7

Evaluation step	Subset ratio	Accuracy	Precision	Recall	F-score
1	0.05	0.9513085	0.9751834	0.842929	0.904214
2	0.1	0.9524105	0.9752841	0.84697	0.906581
3	0.15	0.9523416	0.9739025	0.84798	0.906566
4	0.2	0.952686	0.9736685	0.849495	0.90733
5	0.25	0.9527548	0.9728614	0.850505	0.907557
6	0.3	0.9526171	0.9712098	0.851515	0.90741
7	0.35	0.9524105	0.9701123	0.851768	0.907067
8	0.4	0.952686	0.9701383	0.852778	0.907655
9	0.45	0.9529614	0.9707126	0.853283	0.908187
10	0.5	0.952686	0.9685228	0.854293	0.907802
11	0.55	0.9524105	0.9679486	0.853788	0.907249
12	0.6	0.9523416	0.9673928	0.85404	0.907157
13	0.65	0.9527548	0.9679929	0.855051	0.908
14	0.7	0.9524105	0.967407	0.854293	0.907303
15	0.75	0.9524105	0.9666172	0.855051	0.907395
16	0.8	0.9533058	0.9678182	0.857323	0.9092
17	0.85	0.9530992	0.9675151	0.856818	0.908781
18	0.9	0.9528926	0.9666863	0.856818	0.908417
19	0.95	0.9530303	0.9666978	0.857323	0.908694
20	1	0.9529614	0.9659145	0.857828	0.908633

Table A 15 Confusion matrix of iteration 7

	true normal	true incident	class precision
pred. normal	10447	565	94.87%
pred. incident	113	3395	96.78%
class recall	98.93%	85.73%	
accuracy	95.33%	F-score	90.92

Iteration 8:

Number of trees: 380, Criterion: information gain, Maximal depth: 27, guess subset ratio: False, subset ratio: 0.8, pruning: true, Confidence: 2.00E-07 pre-pruning: False.

Table A 16 Evaluation steps of iteration 8

Evaluation step	Confidence	Accuracy	Precision	Recall	F-score
1	1.00E-07	0.953168	0.9677967	0.856818	0.908906
2	2.00E-07	0.9533058	0.9678182	0.857323	0.9094
3	4.00E-07	0.9533058	0.9678182	0.857323	0.9092
4	6.00E-07	0.9533058	0.9678182	0.857323	0.9092
5	8.00E-07	0.9532369	0.9678041	0.857071	0.909052
6	1.00E-06	0.9532369	0.9678041	0.857071	0.909052
7	2.00E-06	0.953168	0.9677967	0.856818	0.908906
8	4.00E-06	0.9533058	0.9678182	0.857323	0.9092
9	6.00E-06	0.953168	0.9677967	0.856818	0.908906
10	8.00E-06	0.9532369	0.9678108	0.857071	0.909054
11	1.00E-05	0.953168	0.9677967	0.856818	0.908906
12	2.00E-05	0.9532369	0.9678041	0.857071	0.909052
13	4.00E-05	0.9533058	0.9678182	0.857323	0.9092
14	6.00E-05	0.953168	0.9677967	0.856818	0.908906
15	8.00E-05	0.9532369	0.9678041	0.857071	0.909052
16	1.00E-04	0.9533058	0.9678182	0.857323	0.9092
17	2.00E-04	0.9532369	0.9678041	0.857071	0.909052
18	4.00E-04	0.9532369	0.9678108	0.857071	0.909054
19	6.00E-04	0.9532369	0.9678108	0.857071	0.909054
20	8.00E-04	0.9532369	0.9678108	0.857071	0.909054
21	0.001	0.9533058	0.9678182	0.857323	0.9092
22	0.002	0.9532369	0.9678108	0.857071	0.909054
23	0.004	0.9532369	0.9678108	0.857071	0.909054
24	0.006	0.9533058	0.9678182	0.857323	0.9092
25	0.008	0.9532369	0.9678108	0.857071	0.909054

26	0.01	0.953168	0.9677967	0.856818	0.908906
27	0.02	0.9532369	0.9678108	0.857071	0.909054
28	0.04	0.9532369	0.9678041	0.857071	0.909052
29	0.06	0.952686	0.9666586	0.856061	0.907982
30	0.08	0.952686	0.9671949	0.855556	0.907936
31	0.1	0.9527548	0.9672058	0.855808	0.908083
32	0.15	0.953168	0.9677967	0.856818	0.908906
33	0.2	0.9532369	0.9678041	0.857071	0.909052
34	0.25	0.9533058	0.9678182	0.857323	0.9092
35	0.3	0.952686	0.9671949	0.855556	0.907936
36	0.35	0.9517906	0.9675858	0.851768	0.905965
37	0.4	0.9524105	0.9666069	0.855051	0.907391
38	0.45	0.9528926	0.9674937	0.856061	0.908349
39	0.5	0.9528926	0.9674937	0.856061	0.908349

Table A 17 Confusion matrix of iteration 8

	true normal	true incident	class precision
pred. normal	10453	569	94.84%
pred. incident	107	3391	96.94%
class recall	98.99%	85.63%	
accuracy	95.34%	F-score	90.94

Appendix B: Cross-Validation Results for Random Forest Model

Table B 1 TTD and false alarms of Cross-Validation for RF model

Number of blocked lanes	Incident location	Spacing	D/C ratio	Detection Status	TTD (intervals)	TTD (minutes)	False alarms
0	0	500	0.4	Detected	N/A	N/A	1
0	0	1000	0.4	Detected	N/A	N/A	1
0	0	1500	0.4	Detected	N/A	N/A	3
0	0	500	0.5	Detected	N/A	N/A	0
0	0	1000	0.5	Detected	N/A	N/A	0
0	0	1500	0.5	Detected	N/A	N/A	1
0	0	500	0.6	Detected	N/A	N/A	0
0	0	1000	0.6	Detected	N/A	N/A	0
0	0	1500	0.6	Detected	N/A	N/A	0
0	0	500	0.8	Detected	N/A	N/A	0
0	0	1000	0.8	Detected	N/A	N/A	1
0	0	1500	0.8	Detected	N/A	N/A	1
0	0	750	0.9	Detected	N/A	N/A	0
0	0	1250	0.9	Detected	N/A	N/A	1
0	0	500	1	Detected	N/A	N/A	0
0	0	1000	1	Detected	N/A	N/A	0
0	0	1500	1	Detected	N/A	N/A	0
0	0	750	1.1	Detected	N/A	N/A	0
0	0	1250	1.1	Detected	N/A	N/A	0
0	0	500	1.2	Detected	N/A	N/A	0
0	0	1000	1.2	Detected	N/A	N/A	0
0	0	1500	1.2	Detected	N/A	N/A	1
3	250	500	0.4	Detected	0	0	0

3	250	1000	0.4	Detected	0	0	0
4	250	500	0.4	Detected	0	0	0
4	250	1000	0.4	Detected	0	0	0
4	250	500	0.5	Detected	0	0	1
4	250	1000	0.5	Detected	1	0.5	1
3	125	500	0.6	Detected	1	0.5	1
5	125	500	0.6	Detected	0	0	1
1	250	500	0.6	Detected	5	2.5	0
1	250	1000	0.6	Not Detected	-	-	0
5	250	500	0.6	Detected	0	0	0
5	250	1000	0.6	Detected	1	0.5	2
1	375	500	0.6	Detected	1	0.5	1
1	375	1500	0.6	Not Detected	-	-	0
3	375	500	0.6	Detected	0	0	1
3	375	1500	0.6	Detected	1	0.5	1
5	375	500	0.6	Detected	0	0	1
5	375	1500	0.6	Detected	0	0	2
3	500	1000	0.6	Detected	1	0.5	1
5	500	1000	0.6	Detected	1	0.5	1
1	750	1000	0.6	Detected	19	9.5	0
1	750	1500	0.6	Not Detected	-	-	0
3	750	1000	0.6	Detected	1	0.5	0
3	750	1500	0.6	Detected	1	0.5	2
1	1125	1500	0.6	Detected	0	0	1
3	1125	1500	0.6	Detected	0	0	0

5	1125	1500	0.6	Detected	1	0.5	1
1	125	500	0.8	Detected	2	1	1
3	125	500	0.8	Detected	1	0.5	1
5	125	500	0.8	Detected	0	0	1
3	250	500	0.8	Detected	1	0.5	0
3	250	1000	0.8	Detected	1	0.5	1
5	250	500	0.8	Detected	0	0	0
5	250	1000	0.8	Detected	1	0.5	1
1	375	500	0.8	Detected	2	1	0
1	375	1500	0.8	Detected	2	1	1
5	375	500	0.8	Detected	0	0	0
5	375	1500	0.8	Detected	2	1	2
1	500	1000	0.8	Detected	1	0.5	3
3	500	1000	0.8	Detected	1	0.5	2
1	750	1000	0.8	Detected	2	1	1
1	750	1500	0.8	Detected	2	1	2
3	750	1000	0.8	Detected	1	0.5	0
3	750	1500	0.8	Detected	2	1	2
5	750	1000	0.8	Detected	0	0	0
5	750	1500	0.8	Detected	1	0.5	1
1	1125	1500	0.8	Detected	2	1	0
5	1125	1500	0.8	Detected	1	0.5	1
3	125	500	1	Detected	1	0.5	1
5	125	500	1	Detected	0	0	1
1	250	500	1	Detected	2	1	1
1	250	1000	1	Detected	2	1	2

3	250	500	1	Detec ted	1	0.5	0
3	250	1000	1	Detec ted	1	0.5	1
5	250	500	1	Detec ted	0	0	0
5	250	1000	1	Detec ted	1	0.5	1
1	375	500	1	Detec ted	4	2	0
1	375	1500	1	Detec ted	6	3	0
3	375	500	1	Detec ted	0	0	1
3	375	1500	1	Detec ted	2	1	2
5	375	500	1	Detec ted	1	0.5	0
5	375	1500	1	Detec ted	2	1	1
3	500	1000	1	Detec ted	1	0.5	0
5	500	1000	1	Detec ted	1	0.5	1
1	750	1000	1	Detec ted	4	2	1
1	750	1500	1	Detec ted	5	2.5	0
5	750	1000	1	Detec ted	0	0	0
5	750	1500	1	Detec ted	1	0.5	2
1	1125	1500	1	Detec ted	3	1.5	1
3	1125	1500	1	Detec ted	3	1.5	0
1	125	500	1.2	Detec ted	12	6	1
3	125	500	1.2	Detec ted	1	0.5	1
1	250	500	1.2	Detec ted	0	0	1
1	250	1000	1.2	Detec ted	2	1	1
3	250	500	1.2	Detec ted	2	1	0
3	250	1000	1.2	Detec ted	2	1	1
5	250	500	1.2	Detec ted	1	0.5	0
5	250	1000	1.2	Detec ted	2	1	1

3	375	500	1.2	Detec ted	0	0	0
3	375	1500	1.2	Detec ted	2	1	1
5	375	500	1.2	Detec ted	0	0	0
5	375	1500	1.2	Detec ted	2	1	2
1	500	1000	1.2	Detec ted	1	0.5	0
3	500	1000	1.2	Detec ted	1	0.5	0
5	500	1000	1.2	Detec ted	10	5	0
3	750	1000	1.2	Detec ted	1	0.5	0
3	750	1500	1.2	Detec ted	3	1.5	2
5	750	1000	1.2	Detec ted	1	0.5	0
5	750	1500	1.2	Detec ted	3	1.5	3
1	1125	1500	1.2	Detec ted	36	18	2
5	1125	1500	1.2	Detec ted	2	1	0
2	375	750	0.9	Detec ted	2	1	1
4	625	1250	0.9	Detec ted	1	0.5	4
3	250	500	0.9	Detec ted	1	0.5	0
3	250	1000	0.9	Detec ted	2	1	2
2	375	750	1.1	Detec ted	3	1.5	1
4	625	1250	1.1	Detec ted	4	2	1
3	250	500	1.1	Detec ted	1	0.5	0
3	250	1000	1.1	Detec ted	3	1.5	1
					2.09375	1.046875	90

Appendix C: Results of Hyperparameters Optimization for Neural Network Model

Iteration 0:

Number of hidden layers =1, size of hidden layers = 25 neurons, learning rate = 0.01, momentum = 0.9, training cycles = 200 and error epsilon = 0.0001.

Table C 1 Evaluation steps of iteration 0.

Evaluation step	Size of layer 1	Size of layer 2	Recall	Precision	F-score
1	5	0	80.86	96.77	88.1
2	10	0	93.18	96.37	89.29
3	15	0	83.23	96.4	89.34
4	20	0	83.41	95.93	89.23
5	25	0	84.19	95.72	89.59
6	30	0	84.52	94.84	89.39
7	35	0	83.08	96.34	89.22
8	40	0	83.13	96.26	89.21
9	45	0	82.9	96.39	89.14
10	50	0	82.2	97.08	89.02
11	5	5	80.61	96.35	87.77
12	5	10	80.51	95.39	87.31
13	5	15	81.04	94.08	87.1
14	5	20	80.56	95.88	87.55
15	5	25	80.56	95.97	87.58
16	5	30	80.76	95.66	87.58
17	5	35	80.56	95.42	87.36
18	5	40	81.69	94.04	87.44
19	5	45	80.56	95.68	87.46
20	5	50	80.4	96.31	87.64
21	10	5	81.67	96.91	88.64
22	10	10	82.58	93.8	87.89
23	10	15	82.32	96.19	88.72
24	10	20	80.86	96.07	87.81
25	10	25	82.83	95.49	88.71
26	10	30	81.52	96.27	88.28
27	10	35	81.64	95.93	88.21
28	10	40	82.07	96.27	88.6
29	10	45	81.29	96.49	88.24
30	10	50	81.04	96.08	87.9
31	15	5	82.8	95.51	88.71
32	15	10	82.8	95.54	88.72
33	15	15	82.1	95.67	88.37
34	15	20	82.4	96.71	88.98
35	20	20	82.73	96.38	89.03
36	25	25	83.56	95.14	88.98

37	30	30	82.3	96.91	89.01
38	35	35	83.76	96.03	89.48
39	40	40	82.75	96.38	89.05
40	45	45	82.37	96.68	88.95
41	50	50	82.47	97	89.15

Given these parameters the confusion matrix of the Cross-Validation of the initial iteration is shown in the table below.

Table C 2 Confusion matrix of iteration 0

	true normal	true incident	class precision
pred. normal	10411	626	94.33%
pred. incident	149	3334	95.72%
class recall	98.59%	84.19%	
accuracy	94.66%	F-score	89.59%

Iteration 1:

Number of hidden layers =1, size of hidden layers = 25 neurons, learning rate = 0.04, momentum = 0.9, training cycles = 200 and error epsilon = 0.0001.

Table C 3 Evaluation steps of iteration 1

Evaluation step	Learning rate	Accuracy	Precision	Recall	F-score
1	0.001	0.9306474	0.954227	0.783333	0.860313
2	0.002	0.9386364	0.964021	0.805051	0.877376
3	0.003	0.9410468	0.967189	0.811364	0.882446
4	0.004	0.9417355	0.966232	0.814899	0.884094
5	0.005	0.9429752	0.964448	0.821212	0.887041
6	0.006	0.9443526	0.963048	0.827778	0.890273
7	0.007	0.9469008	0.960858	0.839646	0.896107
8	0.008	0.9460744	0.954971	0.842172	0.89494
9	0.009	0.9462121	0.954105	0.843434	0.895332
10	0.01	0.9466253	0.957445	0.841919	0.895889
11	0.02	0.9473829	0.958196	0.844192	0.897482
12	0.03	0.9459366	0.951766	0.844697	0.894992
13	0.04	0.9480028	0.962131	0.842677	0.898383
14	0.05	0.9461433	0.95535	0.842172	0.895108
15	0.06	0.946281	0.94965	0.848232	0.895957
16	0.07	0.9451791	0.949021	0.845202	0.893845

17	0.08	0.944697	0.938748	0.853535	0.893912
18	0.09	0.9432507	0.935547	0.851768	0.891239
19	0.1	0.9420799	0.930083	0.852525	0.889327
20	0.2	0.9431818	0.946033	0.839899	0.889667
21	0.3	0.9277548	0.964243	0.763384	0.851893
22	0.4	0.9086088	0.956841	0.699495	0.799483
23	0.5	0.9190083	0.94558	0.747222	0.833359
24	0.6	0.9225895	0.954316	0.752525	0.841376
25	0.7	0.9123967	0.95983	0.708586	0.814417
26	0.8	0.9123967	0.961559	0.707828	0.813546
27	0.9	0.8860193	0.966487	0.604545	0.723357
28	0.95	0.9033058	0.891921	0.75101	0.810712

Table C 4 Confusion matrix of iteration 1

	true normal	true incident	class precision
pred. normal	10428	623	94.36%
pred. incident	132	3337	96.19%
class recall	98.75%	84.27%	
accuracy	94.8%	F-score	89.84%

Iteration 2:

Number of hidden layers =1, size of hidden layers = 25 neurons, learning rate = 0.04, momentum = 0.9, training cycles = 200 and error epsilon = 0.0001.

Table C 5 Evaluation steps of iteration 2

Evaluation step	Momentum	Accuracy	Precision	Recall	F-score
1	0.001	0.9418733	0.9656576	0.815909	0.884462
2	0.002	0.9418733	0.9653726	0.816162	0.884491
3	0.003	0.9418733	0.9653726	0.816162	0.884491
4	0.004	0.9418733	0.965378	0.816162	0.884492
5	0.005	0.942011	0.9656779	0.816414	0.884766
6	0.006	0.9420799	0.965971	0.816414	0.884888
7	0.007	0.942011	0.9662393	0.815909	0.8847
8	0.008	0.9418044	0.9653693	0.815909	0.884336
9	0.009	0.9418733	0.9653802	0.816162	0.88449
10	0.01	0.9417355	0.9653648	0.815657	0.884185
11	0.02	0.9418733	0.9653972	0.816162	0.884492
12	0.03	0.9419421	0.9648424	0.816919	0.884699
13	0.04	0.9422176	0.9659854	0.816919	0.885184

14	0.05	0.9418733	0.9651061	0.816414	0.884521
15	0.06	0.942011	0.9648545	0.817172	0.884856
16	0.07	0.9419421	0.9648421	0.816919	0.884703
17	0.08	0.9419421	0.9648482	0.816919	0.884705
18	0.09	0.9420799	0.9648719	0.817424	0.885007
19	0.1	0.9424931	0.9643941	0.819444	0.885971
20	0.2	0.942562	0.9633253	0.820707	0.886247
21	0.3	0.944146	0.9622562	0.827778	0.889902
22	0.4	0.9461433	0.9590361	0.838384	0.894635
23	0.5	0.9465565	0.9568088	0.842172	0.895785
24	0.6	0.9466942	0.9597008	0.839899	0.895771
25	0.7	0.9460055	0.953696	0.843182	0.89494
26	0.8	0.944697	0.9430637	0.848737	0.893328
27	0.85	0.9462121	0.9488575	0.84899	0.896035
28	0.9	0.9480028	0.9621311	0.842677	0.898383
29	0.95	0.942562	0.9431885	0.840909	0.888757

Table C 6 Confusion matrix of iteration 2

	true normal	true incident	class precision
pred. normal	10428	623	94.36%
pred. incident	132	3337	96.19%
class recall	98.75%	84.27%	
accuracy	94.8%	F-score	89.84%

Iteration 3

Number of hidden layers =1, size of hidden layers = 25 neurons, learning rate = 0.04, momentum = 0.9, training cycles = 200 and error epsilon = 0.0000000001.

Table C 7 Evaluation steps of iteration 3

Evaluation step	Error epsilon	Accuracy	Precision	Recall	F-score
1	1.00E-10	0.9480028	0.9621311	0.842677	0.898383
2	1.00E-09	0.9480028	0.9621311	0.842677	0.898383
3	1.00E-08	0.9480028	0.9621311	0.842677	0.898383
4	1.00E-07	0.9480028	0.9621311	0.842677	0.898383
5	1.00E-06	0.9480028	0.9621311	0.842677	0.898383
6	1.00E-05	0.9480028	0.9621311	0.842677	0.898383
7	1.00E-04	0.9480028	0.9621311	0.842677	0.898383
8	0.001	0.9480028	0.9621311	0.842677	0.898383
9	0.01	0.9480028	0.9621311	0.842677	0.898383

10	0.1	0.9168733	0.9423289	0.740909	0.829255

Table C 8 Confusion matrix of iteration 3

	true normal		true incident	class precision
pred. normal	10428		623	94.36%
pred. incident	132		3337	96.19%
class recall	98.75%		84.27%	
accuracy	94.8%		F-score	89.84%

Iteration 4

Number of hidden layers =1, size of hidden layers = 25 neurons, learning rate = 0.04, momentum = 0.9, training cycles = 1000 and error epsilon = 0.0000000001.

Table C 9 Evaluation steps of iteration 4

Evaluation step	training cycles	accuracy	precision	recall	f-score
1	50	0.9391185	0.9534893	0.816919	0.879752
2	67	0.9424242	0.9494779	0.833333	0.887595
3	85	0.9426309	0.9495595	0.834091	0.88805
4	102	0.944146	0.9556257	0.834091	0.890677
5	119	0.9456612	0.9595996	0.836111	0.893555
6	136	0.9464876	0.9592995	0.839646	0.895406
7	154	0.9466253	0.9590339	0.840404	0.89572
8	171	0.9466942	0.9565471	0.843182	0.896156
9	188	0.9480028	0.9600453	0.844697	0.898595
10	205	0.947865	0.9611174	0.843182	0.898191
11	223	0.9469697	0.9584265	0.842424	0.896564
12	240	0.946281	0.9566052	0.841667	0.895293
13	257	0.9463499	0.9558513	0.842677	0.895515
14	275	0.9459366	0.9531786	0.843687	0.894926
15	292	0.9460055	0.9526868	0.844444	0.895145
16	309	0.9464876	0.9543254	0.844697	0.896008
17	326	0.9466942	0.9558188	0.843939	0.896264
18	344	0.9462121	0.954016	0.843939	0.895431
19	361	0.9462121	0.9514999	0.846465	0.895732
20	378	0.946832	0.952264	0.84798	0.896946
21	395	0.946832	0.9530535	0.847222	0.89686
22	413	0.9466942	0.9510104	0.848485	0.896735
23	430	0.9459366	0.949589	0.847222	0.895315
24	447	0.9461433	0.9498005	0.847727	0.895725

25	465	0.9471074	0.9542157	0.84697	0.897303
26	482	0.9473829	0.953733	0.848485	0.897947
27	499	0.9465565	0.9523632	0.846717	0.896327
28	516	0.9469008	0.9540045	0.846465	0.896898
29	534	0.9476584	0.9541423	0.849242	0.898522
30	551	0.9475207	0.9523111	0.850505	0.898413
31	568	0.9480028	0.9551897	0.849495	0.899142
32	585	0.9479339	0.9544452	0.85	0.899081
33	603	0.9475207	0.9530537	0.849747	0.898332
34	620	0.9479339	0.9539141	0.850505	0.899134
35	637	0.9471074	0.9540085	0.84697	0.897282
36	655	0.9475895	0.9538757	0.84899	0.898327
37	672	0.9472452	0.9545594	0.84697	0.89751
38	689	0.9473829	0.9536072	0.848485	0.89792
39	706	0.9477273	0.9539798	0.849495	0.898632
40	724	0.947314	0.9520728	0.849747	0.897949
41	741	0.9475207	0.9516021	0.85101	0.898439
42	758	0.9466942	0.950728	0.848737	0.896749
43	775	0.947314	0.9510463	0.850758	0.898061
44	793	0.947314	0.9510752	0.850758	0.898067
45	810	0.9471763	0.9505744	0.850758	0.897809
46	827	0.9477961	0.9506674	0.85303	0.899149
47	845	0.9477961	0.9509296	0.852778	0.899124
48	862	0.9475207	0.9515799	0.85101	0.898452
49	879	0.947865	0.9486482	0.855303	0.899498
50	896	0.9480716	0.9501855	0.854798	0.89985
51	914	0.948416	0.9531587	0.85303	0.900247
52	931	0.9481405	0.9526449	0.852525	0.899711
53	948	0.9481405	0.9535885	0.851515	0.899595
54	965	0.9487603	0.9541534	0.853283	0.900846
55	983	0.9486915	0.9536084	0.853535	0.900749
56	1000	0.9489669	0.9570344	0.851263	0.900989

Table C 10 Confusion matrix of iteration 4

	true normal	true incident	class precision
pred. normal	10408	589	94.64%
pred. incident	152	3371	95.69%
class recall	98.56%	85.13%	
accuracy	94.9%	F-score	90.1%

Iteration 5

Number of hidden layers =1, size of hidden layers = 35 neurons, learning rate = 0.04, momentum = 0.9, training cycles = 1000 and error epsilon = 0.0000000001.

Table C 11 Evaluation steps of iteration 5

Evaluation step	hidden layer size	recall	precision	f-score
1	5	82.1	94.26	87.77
2	10	83.46	95.94	89.26
3	15	83.76	96.62	89.73
4	20	85.25	95.39	90.04
5	25	85.13	95.69	90.1
6	30	85.18	94.91	89.79
7	35	85.76	95.07	90.18
8	40	84.44	96.04	89.87
9	45	85.13	95.31	89.93
10	50	84.14	96.36	89.83

Table C 12 Confusion matrix of iteration 5

	true normal	true incident	class precision
pred. normal	10408	589	94.64%
pred. incident	152	3371	95.69%
class recall	98.56%	85.13%	
accuracy	**94.9%**	**F-score**	**90.1%**

Iteration 6

Number of hidden layers =1, size of hidden layers = 35 neurons, learning rate = 0.015, momentum = 0.9, training cycles = 1000 and error epsilon = 0.0000000001.

Table C 13 Evaluation steps of iteration 6

Evaluation step	learning rate	accuracy	precision	recall	f-score
1	0.001	0.94332	0.961477	0.8252525	0.888140839
2	0.002	0.949036	0.962218	0.8464646	0.900580964
3	0.003	0.949311	0.960973	0.8487374	0.901300015
4	0.004	0.948967	0.956271	0.8520202	0.901067518
5	0.005	0.949311	0.95631	0.8532828	0.901775913
6	0.006	0.949105	0.950866	0.8578283	0.901888507
7	0.007	0.948967	0.953327	0.854798	0.901326453
8	0.008	0.94938	0.955297	0.8545455	0.902055278

9	0.009	0.949105	0.95451	0.8542929	0.901496899
10	0.01	0.94814	0.948724	0.8563131	0.90005481
11	0.015	0.949587	0.950388	0.8603535	0.903016458
12	0.02	0.949174	0.951826	0.8570707	0.901944892
13	0.025	0.947727	0.944642	0.8588384	0.899627937
14	0.03	0.94814	0.951908	0.8530303	0.899724678
15	0.035	0.947727	0.949967	0.8535354	0.899099493
16	0.04	0.949036	0.95104	0.8575758	0.901764918
17	0.045	0.949105	0.95011	0.8585859	0.901987633
18	0.05	0.94759	0.947331	0.8555556	0.899077751
19	0.055	0.949725	0.957445	0.8537879	0.902593436
20	0.06	0.948623	0.955213	0.8517677	0.900449224
21	0.065	0.947383	0.951765	0.8502525	0.898128417
22	0.07	0.94876	0.951654	0.8558081	0.901112774
23	0.075	0.947039	0.944753	0.8560606	0.898133394
24	0.08	0.94876	0.950071	0.8573232	0.901237821
25	0.085	0.948209	0.948386	0.8570707	0.900164021
26	0.09	0.949518	0.959851	0.8507576	0.901875901
27	0.095	0.947727	0.950622	0.8527778	0.89900117
28	0.1	0.946281	0.943011	0.8550505	0.89676614
29	0.15	0.948072	0.954188	0.8507576	0.899381763
30	0.2	0.946006	0.958743	0.8383838	0.894361512
31	0.25	0.935193	0.962477	0.7934343	0.869688369
32	0.3	0.933402	0.963415	0.7856061	0.865406818
33	0.35	0.927066	0.954317	0.7694444	0.851608128
34	0.4	0.925413	0.962074	0.7563131	0.846826918
35	0.45	0.922521	0.949536	0.7563131	0.841743989
36	0.5	0.923347	0.948474	0.7603535	0.84397856
37	0.55	0.921074	0.940898	0.7598485	0.839910613
38	0.6	0.921625	0.957829	0.7454545	0.838279048
39	0.65	0.915083	0.94834	0.7285354	0.823595903
40	0.7	0.91219	0.966994	0.7022727	0.812811092
41	0.75	0.873554	0.955013	0.5628788	0.708293613
42	0.8	0.907645	0.958611	0.6914141	0.802454934
43	0.85	0.911295	0.944948	0.7169192	0.814608305
44	0.9	0.877204	0.90697	0.5780303	0.658203453
45	0.95	0.91281	0.947151	0.7204545	0.818060681

Table C 14 Confusion matrix of iteration 6

	true normal	true incident	class precision
pred. normal	10381	553	94.94%
pred. incident	179	3407	95.01%

class recall	98.30%	86.04%	
accuracy	94.96%	F-score	90.3%

Iteration 7

Number of hidden layers =1, size of hidden layers = 35 neurons, learning rate = 0.015, momentum = 0.9, training cycles = 1000 and error epsilon = 0.0000000001.

Table C 15 Evaluation steps of iteration 7

Evaluation step	momentum	accuracy	precision	recall	f-score
1	0.01	0.94697	0.963206	0.837626	0.895993
2	0.015	0.94697	0.963206	0.837626	0.895993
3	0.02	0.947039	0.963214	0.837879	0.896145
4	0.025	0.947039	0.96321	0.837879	0.896147
5	0.03	0.947107	0.963224	0.838131	0.896296
6	0.035	0.947245	0.963512	0.838384	0.896564
7	0.04	0.947245	0.963247	0.838636	0.896592
8	0.045	0.947521	0.963552	0.839394	0.897154
9	0.05	0.947521	0.963836	0.839141	0.897126
10	0.055	0.94759	0.963844	0.839394	0.897273
11	0.06	0.947658	0.964124	0.839394	0.897395
12	0.065	0.947521	0.963557	0.839394	0.897152
13	0.07	0.94759	0.963565	0.839646	0.8973
14	0.075	0.947796	0.96387	0.840152	0.89772
15	0.08	0.948209	0.964738	0.840909	0.898531
16	0.085	0.948209	0.963933	0.841667	0.898614
17	0.09	0.948278	0.964759	0.841162	0.898688
18	0.095	0.948278	0.965042	0.840909	0.898663
19	0.1	0.948416	0.965335	0.841162	0.898934
20	0.15	0.948485	0.963704	0.842929	0.89924
21	0.2	0.948623	0.961362	0.845707	0.899776
22	0.25	0.948829	0.962164	0.845707	0.900138
23	0.3	0.949174	0.962223	0.84697	0.900888
24	0.35	0.949449	0.962063	0.848232	0.901496
25	0.4	0.949242	0.961448	0.84798	0.901104
26	0.45	0.948003	0.956125	0.848485	0.899013
27	0.5	0.94938	0.961888	0.848232	0.901388
28	0.55	0.948623	0.959976	0.847222	0.899975
29	0.6	0.948347	0.959967	0.846212	0.899333
30	0.65	0.947865	0.95421	0.85	0.898957
31	0.7	0.947934	0.953151	0.851263	0.899166
32	0.75	0.949793	0.956782	0.854798	0.902805
33	0.8	0.948623	0.951204	0.855808	0.900817
34	0.85	0.948967	0.948449	0.859848	0.90187

35	0.9	0.949587	0.950388	0.860354	0.903016

Table C 16 Confusion matrix of iteration 7

	true normal	true incident	class precision
pred. normal	10381	553	94.94%
pred. incident	179	3407	95.01%
class recall	98.30%	86.04%	
accuracy	94.96%	F-score	90.3%

Appendix D: Cross-Validation Results for Neural Network Model

Table D 1 TTD and false alarms of Cross-Validation for MLF model

Number of blocked lanes	Incident location	Spacing	D/C ratio	Detection Status	TTD (intervals)	TTD (minutes)	False alarms
0	0	500	0.4	Detected	N/A	N/A	2
0	0	1000	0.4	Detected	N/A	N/A	4
0	0	1500	0.4	Detected	N/A	N/A	0
0	0	500	0.5	Detected	N/A	N/A	0
0	0	1000	0.5	Detected	N/A	N/A	1
0	0	1500	0.5	Detected	N/A	N/A	0
0	0	500	0.6	Detected	N/A	N/A	0
0	0	1000	0.6	Detected	N/A	N/A	0
0	0	1500	0.6	Detected	N/A	N/A	0
0	0	500	0.8	Detected	N/A	N/A	0
0	0	1000	0.8	Detected	N/A	N/A	2
0	0	1500	0.8	Detected	N/A	N/A	2
0	0	750	0.9	Detected	N/A	N/A	0
0	0	1250	0.9	Detected	N/A	N/A	2
0	0	500	1	Detected	N/A	N/A	0
0	0	1000	1	Detected	N/A	N/A	1
0	0	1500	1	Detected	N/A	N/A	1
0	0	750	1.1	Detected	N/A	N/A	0
0	0	1250	1.1	Detected	N/A	N/A	0
0	0	500	1.2	Detected	N/A	N/A	1
0	0	1000	1.2	Detected	N/A	N/A	0
0	0	1500	1.2	Detected	N/A	N/A	1
3	250	500	0.4	Detected	0	0	1
3	250	1000	0.4	Detected	2	1	3
4	250	500	0.4	Detected	0	0	0
4	250	1000	0.4	Detected	1	0.5	2
4	250	500	0.5	Detected	0	0	0
4	250	1000	0.5	Detected	1	0.5	1
3	125	500	0.6	Detected	1	0.5	1
5	125	500	0.6	Detected	0	0	1
1	250	500	0.6	Not Detected	-	-	0
1	250	1000	0.6	Not Detected	-	-	1
5	250	500	0.6	Detected	0	0	0
5	250	1000	0.6	Detected	1	0.5	3
1	375	500	0.6	Detected	0	0	0
1	375	1500	0.6	Not Detected		0	0
3	375	500	0.6	Detected	0	0	1
3	375	1500	0.6	Detected	1	0.5	1

5	375	500	0.6	Detected	0	0	2
5	375	1500	0.6	Detected	0	0	2
3	500	1000	0.6	Detected	1	0.5	1
5	500	1000	0.6	Detected	1	0.5	3
1	750	1000	0.6	Not Detected	-	-	2
1	750	1500	0.6	Not Detected	-	-	1
3	750	1000	0.6	Detected	0	0	0
3	750	1500	0.6	Detected	1	0.5	1
1	1125	1500	0.6	Detected	1	0.5	1
3	1125	1500	0.6	Detected	0	0	0
5	1125	1500	0.6	Detected	1	0.5	1
1	125	500	0.8	Detected	2	1	1
3	125	500	0.8	Detected	0	0	2
5	125	500	0.8	Detected	0	0	2
3	250	500	0.8	Detected	1	0.5	0
3	250	1000	0.8	Detected	1	0.5	1
5	250	500	0.8	Detected	0	0	2
5	250	1000	0.8	Detected	1	0.5	2
1	375	500	0.8	Detected	1	0.5	0
1	375	1500	0.8	Detected	2	1	1
5	375	500	0.8	Detected	0	0	0
5	375	1500	0.8	Detected	2	1	3
1	500	1000	0.8	Detected	1	0.5	2
3	500	1000	0.8	Detected	1	0.5	4
1	750	1000	0.8	Detected	6	3	1
1	750	1500	0.8	Detected	2	1	4
3	750	1000	0.8	Detected	1	0.5	0
3	750	1500	0.8	Detected	0	0	4
5	750	1000	0.8	Detected	0	0	0
5	750	1500	0.8	Detected	1	0.5	2
1	1125	1500	0.8	Detected	2	1	2
5	1125	1500	0.8	Detected	1	0.5	5
3	125	500	1	Detected	1	0.5	1
5	125	500	1	Detected	0	0	1
1	250	500	1	Detected	3	1.5	1
1	250	1000	1	Detected	1	0.5	1
3	250	500	1	Detected	0	0	0
3	250	1000	1	Detected	1	0.5	1
5	250	500	1	Detected	0	0	0
5	250	1000	1	Detected	1	0.5	3
1	375	500	1	Detected	2	1	3
1	375	1500	1	Detected	3	1.5	3
3	375	500	1	Detected	0	0	1

3	375	1500	1	Detected	2	1	2
5	375	500	1	Detected	0	0	0
5	375	1500	1	Detected	2	1	2
3	500	1000	1	Detected	1	0.5	2
5	500	1000	1	Detected	1	0.5	0
1	750	1000	1	Detected	5	2.5	1
1	750	1500	1	Detected	6	3	3
5	750	1000	1	Detected	1	0.5	1
5	750	1500	1	Detected	1	0.5	3
1	1125	1500	1	Detected	2	1	2
3	1125	1500	1	Detected	3	1.5	0
1	125	500	1.2	Detected	13	6.5	1
3	125	500	1.2	Detected	1	0.5	1
1	250	500	1.2	Detected	1	0.5	1
1	250	1000	1.2	Detected	1	0.5	0
3	250	500	1.2	Detected	0	0	4
3	250	1000	1.2	Detected	0	0	2
5	250	500	1.2	Detected	1	0.5	1
5	250	1000	1.2	Detected	4	2	1
3	375	500	1.2	Detected	0	0	0
3	375	1500	1.2	Detected	3	1.5	2
5	375	500	1.2	Detected	0	0	0
5	375	1500	1.2	Detected	2	1	2
1	500	1000	1.2	Detected	1	0.5	0
3	500	1000	1.2	Detected	1	0.5	0
5	500	1000	1.2	Detected	10	5	0
3	750	1000	1.2	Detected	1	0.5	1
3	750	1500	1.2	Detected	4	2	1
5	750	1000	1.2	Detected	1	0.5	0
5	750	1500	1.2	Detected	3	1.5	2
1	1125	1500	1.2	Detected	32	16	1
5	1125	1500	1.2	Detected	2	1	1
2	375	750	0.9	Detected	1	0.5	4
4	625	1250	0.9	Detected	1	0.5	1
3	250	500	0.9	Detected	0	0	0
3	250	1000	0.9	Detected	2	1	1
2	375	750	1.1	Detected	3	1.5	1
4	625	1250	1.1	Detected	3	1.5	3
3	250	500	1.1	Detected	1	0.5	0
3	250	1000	1.1	Detected	2	1	1
					1.776595745	0.888298	147

Vita

Osama ElSahly was born in 1993 in Dubai, United Arab Emirates. After completing his primary and secondary education at local schools, he graduated with honours from Al Ahliya Private School in Sharjah, UAE in 2011. With a keen interest in civil engineering, Osama pursued his undergraduate studies at the University of Sharjah, where he earned a Bachelor of Science in Civil Engineering in 2016 with magna cum laude. Following his undergraduate studies, Osama joined the American University of Sharjah in 2017 as a graduate teaching assistant and was awarded a graduate assistantship. In 2018, Osama graduated with a Master of Science in Civil Engineering from the American University of Sharjah. Currently, Osama is pursuing his PhD in Engineering Systems Management at the American University of Sharjah. His research focuses on intelligent transportation systems, traffic operations and control, transportation planning, and traffic modelling and simulation.