12-2015

An Application of Fuzzy Logic in Urban Traffic Incident Detection

Feda Wasfi Fayez Mustafa

United Arab Emirates University

College of Engineering

Department of Electrical Engineering

AN APPLICATION OF FUZZY LOGIC IN URBAN TRAFFIC INCIDENT DETECTION

Feda Wasfi Fayez Mustafa

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Under the Supervision of Dr. Nabil Bastaki

December 2015

Declaration of Original Work

I, Feda Wasfi Fayez Mustafa, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this thesis entitled "*An Application of Fuzzy Logic in Urban Traffic Incident Detection*", hereby, solemnly declare that this thesis is my own original research work that has been done and prepared by me under the supervision of Dr. Nabil Bastaki, in the College of Engineering at UAEU. This work has not previously been presented or published, or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my thesis have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this thesis.

Student's Signature: ________________________ Date: _______________

Advisory Committee

1) Advisor: Dr. Nabil Bastaki

Title: Assistant Professor

Department of Electrical Engineering

College of Engineering

2) Co-advisor: Dr. Yasser Hawas

Title: Professor

Director of Roadway, Transportation and Traffic Safety Research Center

Department of Civil Engineering

College of Engineering

Approval of the Master Thesis

This Master Thesis is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Dr. Nabil Bastaki

   Title: Assistant Professor

   Department of Electrical Engineering

   College of Engineering

   Signature ______________________________ Date _______________

2) Member: Dr. Addy Wahyudie

   Title: Assistant Professor

   Department of Electrical Engineering

   College of Engineering

   Signature ______________________________ Date _______________

3) Member (External Examiner): Prof. Bob John

   Title: Professor of Operational Research and Computer Science

   School of Computer Science

   Institution: University of Nottingham, U.K.

   Signature ______________________________ Date _______________

This Master Thesis is accepted by:

Dean of the College of Engineering: Professor Mohsen M. Sherif

Signature ______________________________ Date ____________________

Dean of the College of the Graduate Studies: Professor Nagi T. Wakim

Signature ______________________________ Date ____________________

Copy ____ of ____

Abstract

Traffic congestion in urban areas is an increasing problem around the world. Traffic incidents (such as accidents) are considered as the major source of traffic congestion. Traffic incidents have negative impacts on traffic flow, air pollution and fuel consumption. As a result, increasing interest in finding new techniques to deal with this issue has been shown. Traffic incident-management systems can decrease the effect of such events and keep roads capacity as close as possible to normal levels. Incident detection system is an important part of any incident management system.

This thesis proposes a new approach to incident detection in urban traffic networks using fuzzy logic theory with the objective of reducing traffic delays and increasing road safety. The proposed detection system can be then integrated with a traffic incident management system to reduce traffic congestion related to non-recurrent incident situations. A methodology has been established based on fuzzy logic for detecting incident status in urban areas using detector accumulative count differences. Three fuzzy models were developed and evaluated using simulated data (generated using the commercial software: PTV VISSIM by PTV Group). The fuzzy models can detect incident status on a regular basis (every minute). Performance measures were introduced to capture the capabilities of the suggested models in detecting incidents. The dissertation concludes with a summary of the major findings, recommendations and future research.

Keywords: urban traffic networks, incident detection, traffic incident-management systems, incident detection system, fuzzy logic, performance measures.

Title and Abstract (in Arabic)

تطبيق المنطق الضبابي في اكتشاف الحوادث المرورية في المدن

الملخص

يعد الازدحام المروري في المدن مشكلة متزايدة في جميع أنحاء العالم. وتعتبر الحوادث المرورية مصدراً رئيسياً لهذا الازدحام. لا تقتصر الآثار السلبية للحوادث المرورية على حركة المرور فحسب، بل تمتد لتشمل تلوث الهواء، واستهلاك الوقود، وغيرها. كنتيجة لذلك، أصبح هناك اهتمام متزايد لإيجاد تقنيات جديدة للتعامل مع هذه المشكلة.

يمكن لنظام إدارة الحوادث المرورية التقليل من تأثير مثل هذه الحوادث والحفاظ على استيعاب الطرق أقرب ما يمكن للمستويات العادية. ويعد نظام الكشف عن الحوادث جزءاً هاماً من أي نظام لإدارة الحوادث المرورية.

يهدف هذا البحث للحد من تأخير حركة المرور وزيادة السلامة على الطرق باستخدام طريقة جديدة لكشف الحوادث المرورية في المدن عن طريق نظرية المنطق الضبابي (Fuzzy Logic). ويمكن للنظام المقترح بعد ذلك أن يعمل بطريقة متكاملة مع نظام إدارة الحوادث المرورية للحد من الاختناقات المرورية.

يقدم هذا البحث وصفاً لطريقة بناء نظام لكشف الحوادث المرورية في المدن إذ يقوم هذا النظام على المنطق الضبابي ويستخدم الفروقات في قراءات العدادات. وقد وضعت ثلاثة نماذج للنظام المقترح وتم تقييمها باستخدام سيناريوهات نفذت بواسطة برنامج المحاكاة (PTV VISSIM). ويمكن للنماذج المقترحة الكشف عن حالة الشارع على فترات منتظمة (كل دقيقة). و لقد تم تقييم أداء النماذج باستخدام قياسات التحقق. وختم هذا البحث بملخص للنتائج والتوصيات واقتراحات للأبحاث المستقبلية.

مفاهيم البحث الرئيسية: المنطق الضبابي، الحوادث المرورية، نظام إدارة الحوادث المرورية، قياسات التحقق.

Acknowledgements

First of all, I would like to thank Allah for giving me the faith and strength to successfully complete my work. My deepest thanks go to my family who has supported me to complete this work. I would like to express my deepest thanks for every single person who helped me during this unforgettable period of my life.

At the first place, I would like to deliver my endless respect and appreciation to my thesis supervisor Dr. Nabil Bastaki for his unlimited support, being friendly with enormous guidance, being on time, providing me with precious knowledge and advices during the whole project stages.

I would like to thank Dr. Yasser Hawas for his guidance, support, and assistance throughout my preparation of this thesis.

I would like to thank Mr. Sonal Ahuja and PTV Group for their support and for providing me with their simulator (PTV VISSIM).

My acknowledgments are extended to all faculty members of the Department of Electrical Engineering at the United Arab Emirates University for their continuous support and encouragement.

Dedication

To my beloved parents and family

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AID	Automatic Incident Detection
RTA	Roads and Transport Authority
TIM	Traffic Incident Management
GPS	Global Positioning System
CCTV	Closed-Circuit Television
DR	Detection Rate
FAR	False Alarm Rate
MTTD	Mean Time to Detect
FL	Fuzzy Logic
FLS	Fuzzy Logic System
MF	Membership Functions
ANFIS	Adaptive Neuro-Fuzzy Inference System
FCM	Fuzzy C-Means
FIS	Fuzzy Inference System
ITS	Intelligent Transportation Systems
GUI	Graphical User Interface
US1	First lane upstream detector
MS1	First lane midstream detector
DS1	First lane downstream detector
US2	Second lane upstream detector
MS2	Second lane midstream detector

DS2	Second lane downstream detector
US3	Third lane upstream detector
MS3	Third lane midstream detector
DS3	Third lane downstream detector
FM1	Fuzzy Model 1
FM2	Fuzzy Model 2
FM3	Fuzzy Model 3
Z	Zero
P	Positive
VP	Very-Positive
FPGA	Field-Programmable Gate Array
A/D	Analog-to-Digital
D/A	Digital-to-Analog

Chapter 1: Introduction

This chapter introduces the research problem. It presents the objectives of this study. It also discusses the challenge and the expected impact of this work. Finally, it outlines the organization of this thesis.

1.1 Research Problem

Traffic incident in urban networks is one of the most critical issues in transportation systems, especially with the continuous growth in the number of vehicles on the roads. Traffic incidents can be defined as the non-recurring events that result in traffic congestion such as accidents, stalled vehicles, load spills and other unusual events affecting roadways. Urban streets capability is highly affected by the traffic incidents because they lead to reduction of roadway capacity and increase the queues, which consequently result in more delays. Moreover, traffic incidents have an important impact on safety, pollution, and the cost of travel. Previous studies show that incidents are one of the main causes of time and money waste in transportation networks. “According to the Roads and Transport Authority (RTA), Dubai’s economy was set back by Dh 2.9 billion in terms of loss in working hours, time and fuel in 2013. A simple division of the annual loss with the total length of roads in Dubai, which is 3,760.63 centerline kilometers, shows a loss of Dh 771,147.388 per kilometer” (Shahbandari, 2015).

Incident detection is considered an essential part of traffic incident-management systems. The better the incident detection technique, the better the overall performance of the traffic incident-management system in all. Therefore, it can be said that a successful traffic incident-management system is highly dependent on the identification of urban incidents. Nowadays, there is an increasing interest in developing effective incident detection techniques. Quick and reliable incident detection helps to reduce congestion, delay and secondary incidents. This research proposes a new approach for incident detection in urban traffic networks using fuzzy logic theory.

1.2 Research Objectives

The main objective of this research is to present a methodology for developing an automated system for incident detection in urban areas based on fuzzy logic. The suggested system will utilize the detectors' accumulative counts differences to identify the incident patterns. It aims at increasing the safety of the urban streets, avoiding the secondary accidents and congestions, reducing the amount of delay experienced by the road users, and restoring the normal status of the controlled intersection as fast as possible.

1.3 Challenges

The first challenge in this work was in the process of collecting data. There are two main sources for traffic data, real data source and simulated data source. For this research, beside the information about the intersection under study, it was necessary to get detailed information about the incidents such as their locations and durations. This is because the models developed depend heavily on traffic within individual lanes and the location of incidents compared to the various upstream, midstream and downstream counters. Real traffic data is hard to collect. It is also a time and money consuming process. Moreover, the available real traffic data does not have the details needed for each incident. To overcome these difficulties, simulated data source was selected in this study.

The second challenge was in generating the fuzzy membership functions because the fuzzy set theory does not confine the shape and the width of a fuzzy membership function to any specific form. Finally, another challenge was in setting the number of inputs because it can lead to an exponential increase in the number of generated rules. For example: if the number of inputs is nine with three membership functions per input and there is one output with two membership functions, then the total number of rules will be ($3^9*2 = 39366$ rules). If the number of input variables is reduced to six with three membership functions per input and there is one output with two membership functions, then the total number of rules will be ($3^6*2 = 1458$ rules).

1.4 Expected Impact of this Work

The expected impact of this research can be summarized in the following:

1. Reduction of the number of secondary incidents.
2. Reduction of the travel time delays.
3. Quick restoration of the normal traffic situation.
4. Possibility to warn approaching drivers.
5. Possibility to recommend alternative routes.

1.5 Thesis Organization

This dissertation has been organized as follows:

Chapter 1 (Introduction): The problem of urban traffic incident detection is discussed here. Research objectives, challenges and expected impact are also presented.

Chapter 2 (Literature review): Traffic incident related definitions are presented along with an overview of traffic incident management systems. Incident detection process and the traffic incident pattern are also presented, followed by, an overview of the fuzzy logic theory and fuzzy c-means clustering technique. Towards the end of the chapter, an overview of some of the previous incident detection algorithms/ methodologies is presented.

Chapter 3 (Data collection): Presents the data source used and gives an overview of the simulation setup and parameters. It also discusses the generated traffic data used to develop the fuzzy models.

Chapter 4 (Fuzzy models): The chapter starts by describing the development of three fuzzy models for incident detection in urban streets, including, the inputs and outputs of the proposed models.

Chapter 5 (Results): Fuzzy logic process is discussed in detail and the results obtained from testing the developed models are presented.

Chapter 6 (Conclusion): Includes a summary of this thesis and gives recommendations for future work.

Chapter 2: Literature Review

This chapter discusses the areas of interest in literature. It starts with the definition of traffic incident and then presents an overview of the traffic incident-management system. Next, it discusses the incident detection process and presents an example of a traffic incident pattern. Furthermore, fuzzy logic theory and fuzzy c-means clustering technique are explained. Finally, it presents an overview of a few of the previous incident detection algorithms and models.

2.1 Definition of Traffic Incident

Different definitions have been introduced in literature for traffic incident. One of these definitions is "traffic incident is any non-recurring event that causes a reduction of roadway capacity or an abnormal increase in demand" (Farradyne, 2000). This definition includes traffic accidents, spilled cargo, disabled vehicles, reconstruction projects, maintenance works and special events such as concerts, ball games, or any other event that significantly affects roadway capacity. Another definition is "traffic incident is an unplanned randomly occurring traffic event that adversely effects normal traffic operations" (FHWA, 2013). This definition excludes planned activities such as roadwork or maintenance activities and special events. The second definition is considered in this thesis.

2.2 Traffic Incident Management (TIM) System

A TIM system consists of five steps: incident detection, verification, information dissemination, and response and incident clearance. Incident detection is the first step in a TIM system. Its purpose is to get information about the existence of an incident. This can be achieved using either manual or automatic methods. The second step is verification where the occurrence of the incident is confirmed. This step also includes important details about the incident such as its severity and location. The next step of a TIM system is information dissemination. In this step, the incident information is passed to the concerned authority that will be responsible for the next two steps of a TIM system, which are response and clearance. Distributing and sharing the incident information can be achieved in several ways such as radio broadcasts, mobile applications, Internet services, etc. Once the incident information is shared, the responsible agency is expected to respond by dispatching the team or person in charge to head to the incident location. The last step is clearance, which concerns removing obstacles that may disrupt the traffic flow such as vehicle parts or any other material left by the incident. A flow chart of a TIM system is shown in Figure 1. Different technologies are used in a TIM system such as roadside camera recognition, inductive loop detectors, radio wave or infrared beacons, Global Positioning System (GPS), wireless networks, mobile phone technology, etc. All these technologies play a vital role in improving the performance of a TIM system and accordingly they have a great impact on the incident recovery process. (Deniz, O.; Celikoglu, H. B., 2011)

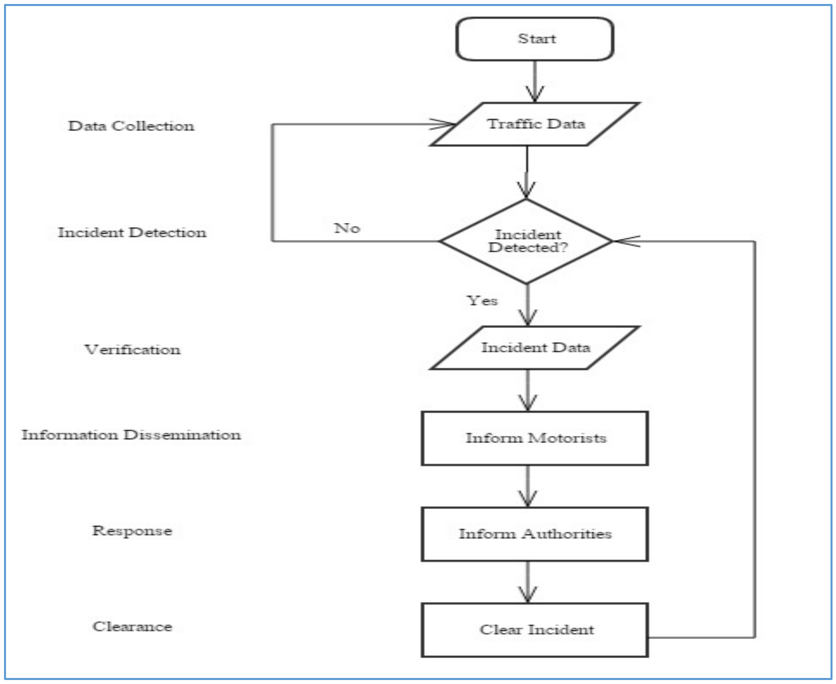


Figure 1: A flow chart of the TIM system

2.3 Incident Detection

Incident detection includes collecting and analyzing traffic data in order to determine the existence of an incident. Incident detection methods can be divided into two types: manual and automatic. Closed-circuit television (CCTV) cameras, phone calls from road users present at the incident location and routine police patrol belong to the manual methods. On the other hand, automated methods are mainly based on traffic flow data collected using inductive loops or roadside detectors and include conventional algorithms and artificial neural network-based models.

CCTV cameras can provide information about exact incident location and severity. It can also provide information about volume, speed, occupancy and headway data. CCTV cameras provide video images as a data, which can be processed manually by an operator or automatically by an image-processing algorithm to detect incidents.

Loop detectors may provide traffic information such as volume, speed (with two detectors), occupancy, vehicle class and headway data. Mathematical models and algorithms can be used to detect incidents with data obtained using loop detectors. (Deniz, O.; Celikoglu, H. B., 2011)

Both manual and automatic incident detection methods can be used in a TIM system concurrently where the automatic part is used to detect the presence of an incident and the manual part such as CCTV cameras is used to confirm the existence of an incident. Incident detection is a critical part of the incident management procedure. It is important to detect an incident as quickly as possible to guarantee the success of the entire procedure. Traffic congestion and long queues are inevitable if the incident detection was delayed which could lead to additional accidents. Quick and timely incident detection help in saving lives and money.

Incident detection is facing three challenges: (i) surveillance problems, (ii) verification problems and (iii) algorithmic problems.

2.3.1 Surveillance Issues

Traffic surveillance can be defined as the process of collecting traffic flow data and sending it to the traffic operation center. There are different traffic sensors and technologies used to gather traffic data. Loop detectors are the most broadly used sensors. Table 1 summarizes the technologies that are used in traffic sensors (Ozbay, K.; Kachroo, P., 1999). Active sensors can transmit and receive signals. They transmit a signal toward a targeted object and measure its reflection, while passive sensors can only receive signals emitted from an object.

Table 1: Detector technologies

Sensor Technology	Active/ Passive	How it works
Inductive loop technology	Active	Detect vehicles by responding to ferrous mass (vehicles)
Magnetometer technology	Passive	Responds to ferrous mass (vehicles)
Infrared technology	Active or passive	• Active infrared technology uses reflected signals • Passive infrared technology uses the contract in thermal radiation
Acoustic detection technology	Passive	Detect vehicles using sound
Ultrasonic detection technology	Active	Detect vehicles using reflected sound
Charged coupled devices (CCD) camera	Passive	Detect vehicles using contrast in visible light
Doppler radar detection	Active	Detect vehicles using frequency shift of reflected signal
Pulsed radar technology	Active	Detect vehicles using reflected signals

There are few factors that are commonly considered when evaluating traffic sensors; they are reliability, data accuracy, performance under different environmental conditions, real-time performance and cost. It can be said that reliability is the most important factor since recurrent failures of sensors can seriously affect the performance of the entire TIM system. Data accuracy is another factor that depends on the calibration of sensors. Both very high and very low sensor accuracies may lead to errors in incident detection. The third factor is the performance under different environmental conditions. Traffic sensors may perform differently under various environmental situations, and hence, it is essential to consider the environment when selecting specific types of sensors. The next factor is the real-time performance which has a great influence on detection time and accordingly on the decision making duration. The last factor is the cost where high cost is not desirable even if the sensor is rated for high performance in terms of all the aforementioned factors. Given all that, the selection of the traffic sensor should be performed very carefully due to its high impact on the incident detection process. (Ozbay, K.; Kachroo, P., 1999)

2.3.2 Verification Issues

Verification is about evaluating incident detection systems. Three main measures are used to evaluate incident detection algorithms, detection rate (DR), false alarm rate (FAR) and mean time to detect (MTTD). DR is evaluated by finding the ratio of the number of detected incidents to the actual number of incidents in the data set. FAR is calculated by dividing the number of incorrect detections by the total number of algorithm applications. MTTD is the average time needed for the algorithm to detect an incident.

One issue with these measures is the lack of a standard methodology to obtain them, which makes it difficult to compare the evaluation results among the different algorithms. Another issue is the dependency of these measures on each other. For example, longer detection time signifies fewer false alarms; however, it indicates that there is a tradeoff between loosing precision in time and lowering false alarm rates. (Ozbay, K.; Kachroo, P., 1999)

2.3.3 Algorithmic Issues

There are many algorithms presented in the literature for automatic incident detection. These algorithms can be divided into two categories: point based algorithms and spatial measurement based algorithms. Point based algorithms are based on traffic flow measurements made at one point. They use the following approaches to detect incidents on the freeways: comparative or pattern recognition, statistics, traffic model and theoretical algorithms and artificial intelligence based algorithms. Spatial measurement based algorithms make use of video cameras and image processing techniques. There is no one algorithm that can be applied anywhere or anytime. Each algorithm has its constraints and limitations which makes it challenging to select the finest algorithm. (Ozbay, K.; Kachroo, P., 1999)

2.4 Traffic Incident Patterns

Incident patterns can be affected by factors such as link volume, link length, incident duration, incident location and the green cycle of the traffic signal (Hawas, 2007).

Figure 2 shows an example of a traffic incident pattern presented using detector accumulative counts.

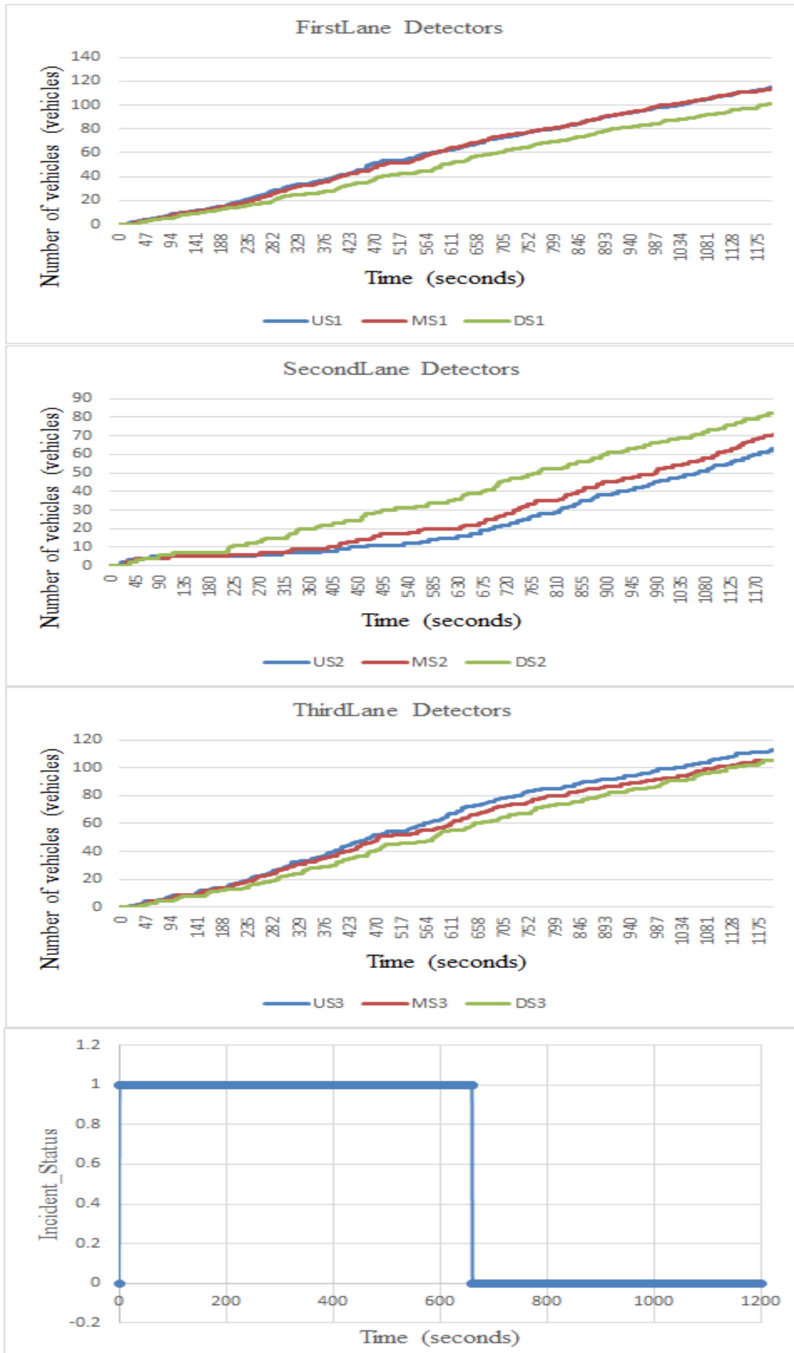


Figure 2: Traffic incident detector counts pattern

This section presents a brief overview of the influence of traffic incident on the accumulative detector counts pattern. Figure 2 displays an example of the detector counts pattern for an incident of 10 minute duration. It started at minute 1 (60 seconds) and finished at minute 11 (660 seconds) where the total duration of the simulation was 20 minutes. The link had three lanes of length 300 m. The green cycle of the traffic signal at the downstream was 30 seconds and the link volume was 1000 vph. Each lane was equipped with three detectors at 10 m, 150 m and 290 m (from the upstream starting line). The incident was introduced on the second lane at a distance of 75 m from the upstream starting line.

The incident was positioned in the middle of the distance between the second lane upstream (US2) and midstream (MS2) detectors, in order to have a clear effect on both detectors (see second lane detectors in Figure 2). The patterns of the first and second lanes are almost the same which is close to the no-incident pattern except that there are more vehicles served than in normal situations. The second lane detector readings show that it served less number of vehicles (during the incident duration) than the other two lanes. After the incident has been cleared, the second lane started serving more vehicles and returned to normal status (Hawas, 2007).

2.5 Fuzzy Logic (FL)

In 1965, Lotfi A. Zadeh of the University of California at Berkeley introduced and published the concept of "Fuzzy Sets" which was extended later to fuzzy logic (FL) theory. Zadeh had observed that computers could not handle data with vague or subjective concepts using conventional logic, so he came up with the idea of fuzzy logic to give computers the ability to operate such data in a way similar to the process of human reasoning (Klir, 1995) (Zadeh, 2008).

FL depends on the "degrees of truth" instead of the usual "true or false" Boolean logic. It includes 0 and 1 as extreme cases where 0 represents absolute falseness and 1 represents absolute truth. The rest of the truth values are indicated by a value in the range [0, 1]. FL works in a way similar to how our brain works. Fuzzy Logic System (FLS) is defined as the nonlinear mapping of an input data set to a scalar output data set. It consists of three main parts: fuzzifier, inference engine, and defuzzifier. FLS components are depicted in Figure 3.

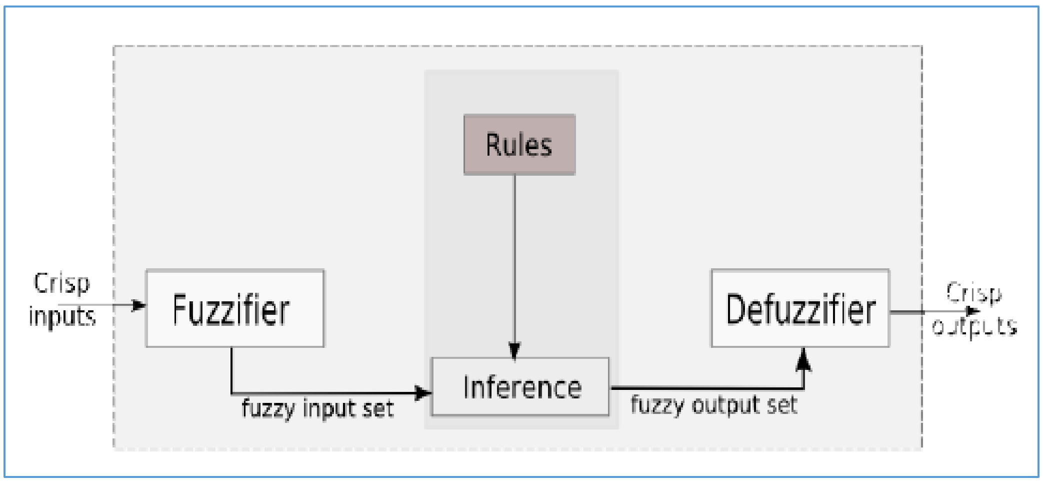


Figure 3: FLS components

The initialization of the FLS involves defining the linguistic variables and terms, constructing the membership functions and the rule base. Subsequently, the process of FL starts with converting a crisp set of input data to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions. This step is called fuzzification. Next, the fuzzy rules in the rule base are applied and the results of all rules are combined into one fuzzy output. In other words, the fuzzy input is converted to a fuzzy output. This step is called fuzzy inference. In the last step, the resulting fuzzy output is mapped to a crisp output value using the membership functions, which is known as the defuzzification step (Bay, Y.; Wang, D., 2006) (Kaehler, 2003).

2.5.1 Linguistic Variables

Linguistic variables are the inputs and outputs of the FLS that are represented by words instead of numerical values. The range of possible values of a linguistic variable represents the universe of discourse of that variable. Each linguistic variable in the system is split into a set of linguistic terms. For example, if the temperature (t) is considered as a linguistic variable that represents the room temperature, then T(t) = {too_cold, cold, normal, hot, too_hot} can be the set of linguistic terms where each member of the set is called a linguistic term. Each linguistic term can cover a portion of the temperature universe.

2.5.2 Membership Functions (MFs)

A membership function (MF) is a curve that defines how each point in the input or the output space is mapped to a degree of membership (μ) between 0 and 1. MFs are used in the fuzzification step to convert the crisp inputs to fuzzy ones. They are also used in the defuzzification step to convert the fuzzy output to a crisp output. Each linguistic variable is represented by a set of membership functions where each membership function represents a linguistic term. Figure 4 shows the membership functions for the linguistic terms of temperature linguistic variable. Any input value can be fuzzified to one or more membership functions. It means that the input value may belong to multiple membership functions (linguistic terms) at the same time.

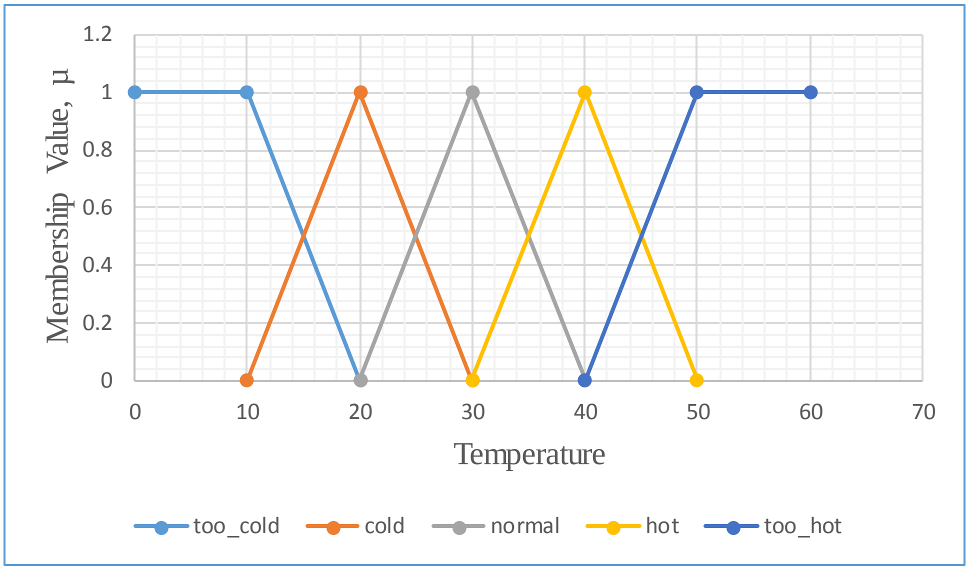


Figure 4: Temperature MFs or linguistic terms

MFs can have different shapes; such as triangular, trapezoidal, Gaussian, generalized bell and sigmoidal. The selection of the best shape MFs for a specific application depends on the application itself and the user experience (MENDEL, 1995). Below is the mathematical representation of the triangular and the trapezoidal MFs which are used in this research.

- Triangular MF:

A triangular MF is defined using three parameters {a, b, c} as follows:

$$Triangle(x;a,b,c) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ \dfrac{c-x}{c-b}, b \le x \le c \\ 0, c \le x \end{cases} \quad \ldots\ldots\ldots\ \text{Equation 1}$$

Another expression for the previous equation is represented below:

$$Triangle(x;a,b,c) = \max\left(\min\left(\frac{x-a}{b-a},\frac{c-x}{c-b}\right),0\right) \quad \text{………. Equation 2}$$

The parameters {a, b, c} (with a < b < c) define the x coordinates of the three corners of the underlying triangular MF.

- Trapezoidal MF:

A trapezoidal MF is defined using four parameters {a, b, c} as follows:

$$Trapezoid(x;a,b,c,d) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ 1, b \le x \le c \\ \dfrac{d-x}{d-c}, c \le x \le d \\ 0, d \le x \end{cases} \quad \text{………. Equation 3}$$

Another expression for the previous equation is represented below:

$$Trapezoid(x;a,b,c,d) = \max\left(\min\left(\frac{x-a}{b-a},1,\frac{d-x}{d-c}\right),0\right) \quad \text{………. Equation 4}$$

The parameters {a, b, c, d} (with a < b < c < d) define the x coordinates of the four corners of the underlying trapezoidal MF.

There are two main points to be considered when designing membership functions:

- An overlap should exist only between the nearest neighboring membership functions.
- The summation of the membership values for any input data should be 1 (or nearly).

2.5.3 Fuzzy Rules

The fuzzy rules are the most important part of the FLS and generating these rules is usually the most difficult step in the design process. The fuzzy rules map the fuzzy (or linguistic) inputs to fuzzy (or linguistic) outputs and they are usually in the form of if-then statements. A fuzzy rule consists of two parts: an IF part (antecedent) that consists of a few combinations of inputs and a THEN part (consequent) that consists of output variables. Both parts (antecedent and consequent) may contain several linguistic variables or only one.

Fuzzy rules give FLS the advantage of adding human-like subjective reasoning capabilities to machine intelligence, which are usually based on Boolean logic. Two methods can be used to construct fuzzy rules. The first one is based on expert knowledge. The second one is based on data input-output pairs. Fuzzy rules based on expert knowledge depend on observations by human operators. A trade-off between the complexity of the model and its clarity should be taken into consideration. Fuzzy rules based on data are also indirectly based on expert knowledge. This is because the selection of the data points and the variables is (preferably) performed by an expert. The step following the formation of if-then statements, is usually data clustering using a suitable clustering technique. In this research, fuzzy rules are constructed based on data obtained using a well-known traffic simulator called “PTV VISSIM”.

2.5.4 Fuzzy Inference Systems (FIS)

There are two main types of fuzzy logic inference systems, Mamdani and Sugeno. Of these two types, Mamdani fuzzy inference system is selected in this research.
Mamdani type is the most commonly used FIS. Ebrahim Mamdani proposed Mamdani fuzzy inference system in 1975 to control a steam engine and boiler combination.
Sugeno type was first introduced in 1985. It works in a similar way to Mamdani type in fuzzifying the inputs and applying the fuzzy rules. The main difference between the two types is how to derive crisp output from fuzzy inputs. In Mamdani FIS, the resultant fuzzy output is converted to a crisp output using a defuzzification technique while in Sugeno FIS there is no need for the defuzzification process and the crisp output is computed using weighted average. Moreover, Mamdani FIS has output membership functions where Sugeno FIS has no output membership functions. (Kamboj, V.; Kaur, A., 2013)

2.5.4.1 Comparison between Mamdani and Sugeno FIS

Sugeno FIS is more flexible than Mamdani FIS as the former can be integrated with an Adaptive Neuro-Fuzzy Inference System (ANFIS) tool to optimize the outputs. Additionally, Sugeno type is more compact and computationally efficient when compared to Mamdani type. It is chosen over Mamdani type in control problems (especially dynamic nonlinear systems). This is because it works well with optimization and adaptive techniques that are useful in adjusting the membership functions to derive fuzzy system with best data models. On the other hand, Mamdani type permits to describe the expertise in more intuitive or more human-like manner and therefore, it is commonly used for capturing expert knowledge. (Kamboj, V.; Kaur, A., 2013)

The advantages of both Mamdani and Sugeno FISs are summarized below (MathWorks, Comparison of Sugeno and Mamdani Systems, 2015):

Advantages of the Mamdani FIS:

- It is intuitive.
- It has widespread acceptance.
- It is well suited to human input.

Advantages of the Sugeno FIS:

- It is computationally efficient.
- It works well with linear techniques such as PID control.
- It works well with optimization and adaptive techniques.
- It has guaranteed continuity of the output surface.
- It is well suited to mathematical analysis.

Since Mamdani FIS allows describing the expertise in a human-like manner, and because the problem discussed in this study is a pattern recognition problem, Mamdani FIS is selected to be applied in this research.

2.5.5 Fuzzy Set Operations

Fuzzy set operations are used to evaluate the fuzzy rules and to combine the results of individual rules. Three fuzzy set operations commonly used: fuzzy intersection (using AND operator), fuzzy union (using OR operator) and fuzzy complement (using NOT operator). There is a clear difference between conventional and fuzzy operations although they use the same operators (AND, OR and NOT). Both fuzzy intersection and fuzzy union can be performed in various ways. The most used operation for AND operator is

MIN and the most used operation for OR operator is MAX. Table 2 shows few fuzzy set operations using AND and OR operators. Note that μ_A and μ_B are the membership functions for fuzzy sets A and B. (Nedjah, N.; Mourelle, L., 2005) (Engin, 2010).

Table 2: Fuzzy set operations

AND (intersection)	OR (Union)
MIN $Min\{\mu_A(x), \mu_B(x)\}$	MAX $Max\{\mu_A(x), \mu_B(x)\}$
PROD $\mu_A(x)\mu_B(x)$	ASUM $\mu_A(x)+\mu_B(x)-\mu_A(x)\mu_B(x)$
BDIF $Max\{0, \mu_A(x), \mu_B(x)-1\}$	BSUM $Min\{1, \mu_A(x), \mu_B(x)\}$

Equation 5 is used for evaluating the complement (NOT).

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \text{ ………. Equation 5}$$

Evaluating all the rules is followed by combining the obtained results into one final answer. This procedure is called inference. The results of individual rules can be combined using different aggregation methods (see Table 3). The maximum method is the one most used and it is the default method in MATLAB fuzzy logic toolbox.

Table 3: Aggregation methods

Operation	Formula
Maximum	$Max\{\mu_A(x), \mu_B(x)\}$
Bounded sum	$Min\{1, \mu_A(x)+\mu_B(x)\}$
Normalized sum	$\overline{Max\{1, Max\{\mu_A(x'), \mu_B(x')\}\}}$

2.5.6 Defuzzifcation

The result obtained from the inference process is a vague or fuzzy value. This value should be converted to a crisp value to be used in real applications. The process of converting the fuzzy output to a crisp output is called defuzzification. There are different defuzzification techniques such as centroid method, mean of maximum (or middle of maxima) method, the height (or max-membership) method, etc. The selection of the defuzzification method depends on the application. The technique commonly used is centroid (also known as center of area or center of gravity) method and it is the one used in this research.

2.5.6.1 Center of Gravity

Center of gravity method was developed by Sugeno in 1985. It is the most frequently used technique. In this method, the center of gravity of the area under the output membership function is calculated. In other words, it finds the point where a vertical line would slice the aggregate output set into two equal masses. The only difficulty with this technique is that it is computationally difficult for complex membership functions (Sameena Naaz, Afshar Alam, Ranjit Biswas, 2011). The mathematical representation of the centroid defuzzification technique is presented in Equation 6.

$$z_{COG} = \frac{\int_z \mu_A(z) z dz}{\int_z \mu_A(z) dz} \text{ ………. Equation 6}$$

z_{COG}: The crisp output

$\mu_A(z)$: The aggregated membership function

z: The output variable

2.5.7 Fuzzy Logic Advantages and Drawbacks

Fuzzy logic is used to deal with enigmatic concepts but it is not the only method. It is usually preferred due to its great function approximation in control engineering. Fuzzy logic has many advantages that especially people who work in control engineering can make use of it. These advantages can be summarized in the following points (Albertos, P.; Sala, A., 1998):

1. Simple base design with the same language used in control and supervision.
2. Intuitive interface that can be used by experts and non-experts alike.
3. Easy computations and easy access toolboxes.
4. Fast self-learning and improved user interpretation.
5. Fuzzy logic is perfect for interpreting ambiguous information and dealing with incomplete data that result in different actions based on probability.
6. Fuzzy logic has implemented control schemas that combine regulation algorithms and logic reasoning.
7. The model of soft computations has proved itself in other prototypes.
8. Fuzzy logic control has universal approximation abilities that can be used to calibrate certain plant nonlinearities in conventional designs.

Fuzzy logic's specific capabilities do not make it a perfect solution for all cases. There are certain drawbacks when using fuzzy logic. These drawbacks are as follows (Albertos, P.; Sala, A., 1998):

1. Experimental and has a slow response even if applied before but in a different location.
2. Other conventional controllers that are well-calibrated perform better than the intuitive fuzzy design.
3. Fuzzy logic uses manual local parameters that consume a lot of time when calibrated. This feature is most suited for autonomous learning algorithms.
4. System configuration may become confusing because of the many system options that need to be set: conjunction, disjunction and implication.
5. The uses of non-standard file formats in commercial packages complicates simple applications.
6. Difficult to estimate membership functions.
7. There are numerous ways of interpreting fuzzy rules, combining the outputs of several fuzzy rules and defuzzifying the output.

2.6 Fuzzy C-Means Clustering Technique

Data clustering is a method of dividing data elements into clusters where items in the same cluster are as similar as possible, and the ones in various clusters are as different as possible. Items are placed into clusters depending on the aim of clustering and the nature of data.

Fuzzy C-Means (FCM) algorithm is one of the most commonly used fuzzy clustering algorithms. The FCM algorithm was firstly introduced by Ruspini. Then, it was extended by Dunn and Bezdek and has been broadly used in many applications such as cluster analysis, pattern recognition and image processing (Lu, Y.; Ma, T.; Yin, C.; Xie, X.; Tian, W.; Zhong, S., 2013). FCM allows data elements to be in two or more clusters. This algorithm assigns membership to each data point belonging to each cluster based on the interval between the data point and the cluster center. The closer the data is to the cluster center, the more it is a member of (membership) the cluster center. For each data point, the summation of its membership should be equal to one. FCM execute clustering by iteratively looking for a group of fuzzy clusters and the related cluster centers that show the frame of the data as good as possible. FCM depends on the user to determine the number of clusters in the data set. The membership and the cluster center should be updated after each iteration. The following formula should be used for the updating:

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/m-1)} \quad \text{... Equation 7}$$

$$v_j = (\sum_{i=1}^{n} (u_{ij})^m x_i) / (\sum_{i=1}^{n} (u_{ij})^m), \forall j = 1, 2, ...c \quad \text{.............. Equation 8}$$

Where,

n : The number of data points.

v_j : Represents the jth cluster center

$m \in [1, \infty]$, m : Fuzziness index

c: Number of cluster center

μ_{ij}: Membership of ith data to jth cluster center

d_{ij}: Euclidean distance between ith data and jth cluster center

Algorithmic steps for Fuzzy c-means clustering:

Let $X = \{x_1, x_2, x_3.....x_n\}$ be the set of data points and $V = \{v_1, v_2, v_3,....v_c\}$ be the set of centers.

1) Randomly select “c” cluster centers.

2) Calculate the fuzzy membership "μ_{ij}" using:

$$\mu_{ij} = 1 / \sum_{k=1}^{c} (d_{ij} / d_{ik})^{(2/m-1)}$$

3) Compute the fuzzy centers v_j using:

$$v_j = (\sum_{i=1}^{n} (\mu_{ij})^m x_i) / (\sum_{i=1}^{n} (\mu_{ij})^m), \forall j = 1, 2, ...c$$

4) Repeat steps 2) and 3) until the minimum “J” value is achieved or until $\|U^{(k+1)} - U^{(k)}\| < \beta$

Where,

k: Iteration step

β: Termination criterion in the interval [0,1].

$U = (\mu_{ij})_{n*c}$: The fuzzy membership matrix

J: The objective function

2.6.1 Advantages

1. Gives best results for overlapped data sets.
2. Data point may belong to more than one cluster.

   (Balaji, K.; Juby, Z. N., 2007)

2.6.2 Disadvantages

1. The designer should specify the number of clusters.
2. With lower value of β, better results are achieved but at the expense of more iterations.
3. Euclidean distance measures can unequally weight underlying factors.

   (Balaji, K.; Juby, Z. N., 2007)

2.7 Previous AID Algorithms and Models

Many incident detection algorithms are presented in the literature but only a few of them focus on incident detection in urban areas. Detecting incidents in urban areas is a challenging task due to the high similarity between incident patterns and no-incident patterns when the link is congested. Another difficulty faced in studying incident detection in urban streets is the lack of real data needed for calibration and verification of the suggested models (Hawas, 2007) (Hawas, Y. E.; Mohammad, M. S., 2015).

Different incident detection algorithms and models may use dissimilar traffic variables. For example, (Barria, A. J.; Thajchayapong, S., 2011) used some microscopic traffic variables and their variances to detect and classify traffic abnormalities. The variables used in the study were relative speed, inter-vehicle time gap and lane changing and it was built based on simulated data. (La-inchua, J.; Chivapreecha, S.; Thajchayapong, S., 2013) proposed a Fuzzy Inference System (FIS) to classify traffic status as normal and abnormal. It used mean speed (macroscopic variable) and standard deviation of inter-arrival time (microscopic variable) as inputs to the FIS, after that, the majority voting was applied to the outputs of FIS to improve the system performance (detection rate and mean time to detection). This study also depended on simulated data that was generated using AIMSUN software. In (Yaguang, K.; Anke, X., 2006) , the algorithm used four input variables: queue length, travel time, speed and occupancy. Again, fuzzy logic and simulated data were employed in the study. In (Lee, S.; Krammes, R. A.; Yen, J., 1998), fuzzy logic was used to detect lane-blocking incidents. Simulated data was used. The model used queue length, speed and occupancy measures and their averages over five minute periods to build the proposed fuzzy model. In (Cano, J.; Kovaceva, J.; Lindman, M.; Brännström, M.), the algorithm established the core of a video-based automatic incident detection at system intersections. Both simulated and real data were used in the study. (Hawas, 2007) and (Ahmed, F.; Hawas, Y. E., 2012) used detector readings to characterize the different incident patterns. The two studies used simulated data and applied neuro-fuzzy logic models.

All the previous mentioned studies focused on incident detection in urban areas and there are many more studies on freeway incident detection.

Chapter 3: Data Collection

This chapter presents the first practical step of this thesis, namely data collection. The development of Intelligent Transportation Systems (ITS) requires high quality traffic information in real-time. Gathering data about traffic patterns is becoming more important each year, which is generating high demand for means to modernize and evolve existing data gathering methods. The first part of this chapter presents the selected data source and the reasons for choosing it over other sources. The second part overviews the simulation setup and the parameters used. The last part of this chapter discusses the generated traffic data which will be used to develop and validate the fuzzy models described in subsequent chapters.

3.1 Data Source

There are two main sources of traffic data. There is the real data that can be collected with cameras and sensors. The second – and the most widely used in research – is simulated data generated by traffic simulators. Since real traffic-data are limited and not easily reachable, simulated data source is used in this research. One advantage of using a simulator to generate the required data is its simplicity and flexibility. The Graphical User Interface (GUI) supports the simplicity of such simulators. Simulation runs can be performed any time of day and regardless of weather conditions. Furthermore, the use of a simulator reduces the needed resource cost and enables us to generate and analyze rare scenarios. Given the above-mentioned advantages, this research uses traffic data generated using the commercially available PTV VISSIM simulation software. PTV VISSIM is a microscopic simulator that is widely used for studying problems related to road traffic control. In this research, the simulator is used to simulate incidents that arise as a consequence of a vehicle suddenly stopping in one of the simulated street lanes. The generated traffic data are collected before and after the incident.

3.2 Simulation Setup

The case intersection considered in this study is a simple four approach-intersection with the two intersection roads carrying vehicles in a single direction. The length of each approach is set to 300 m. The roads are generic three-lane roads with a width of 3.5 m per lane. Each lane is equipped with detectors in three different locations: upstream, midstream and downstream. The downstream detector is located at a distance of 10 m away from the stop line, the midstream detector is positioned at a distance of 150 m from the stop line, and the upstream detector is placed at a distance of 290 m away from the stop line. The intersection is operated with pre-timed (fixed-time) traffic light signals and the intersection does not have any secondary entrances or exits of any sort.

A layout of the described intersection is shown in Figure 5.

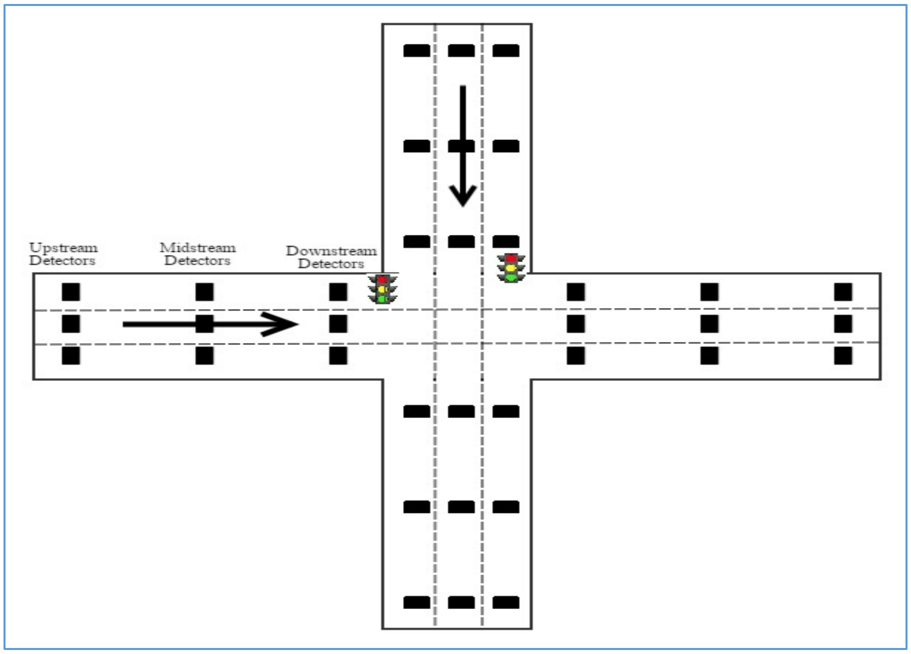


Figure 5: A layout of the intersection

Since this research is concerned with presenting a methodology to develop fuzzy logic models for incident detection, data collected from one intersection approach will be used to develop the models and thereafter, the same steps can be repeated to build a complete fuzzy model for all intersection approaches. This can be generalized later to any intersection if the required data is available. Therefore, the work here is performed on a single approach that is equipped with nine detectors (three per lane). The single approach diagram is shown in Figure 6.

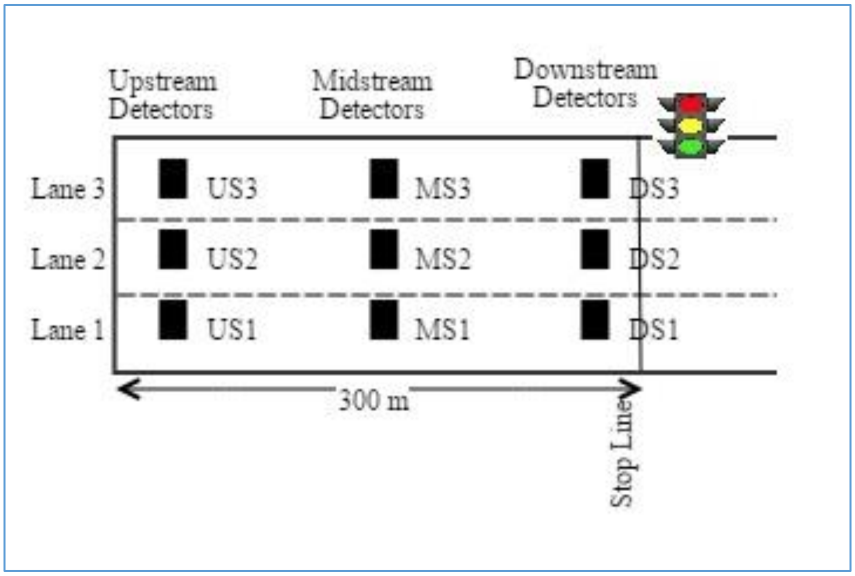


Figure 6: A layout of a single approach

The following are the detectors named based on their location:

1. US1: First lane upstream detector
2. MS1: First lane midstream detector
3. DS1: First lane downstream detector
4. US2: Second lane upstream detector
5. MS2: Second lane midstream detector
6. DS2: Second lane downstream detector
7. US3: Third lane upstream detector
8. MS3: Third lane midstream detector
9. DS3: Third lane downstream detector

The simulator parameters are set as follows:

- Period: 1200 Simulation seconds (20 min.)
- Number of runs: 5
- Simulation resolution: 10 Time steps/ Simulation Sec.

- Random seed: 42
- Simulation Speed: Maximum
- Number of cores: use all cores

3.3 Generated Traffic Data

The traffic data needed for this research are the readings obtained from the detectors. Those readings reflect different patterns in incident cases compared to the ones during regular traffic. Incidents considered in this study are the ones that can cause a lane blockage such as a stopped vehicle, traffic accidents, cargo spills, road maintenance, etc. In this research, the occurrences of incidents are analyzed one at a time where no more than one incident is assumed to happen at the same time. Various incident scenarios are generated using the pre-described intersection. The detectors' data are obtained and used in developing the fuzzy logic models. The simulation period is 20 minutes (1200 seconds). It means that in real life, detectors should be reset every 20 minutes in order for this logic to be applicable. Incidents are simulated at different locations along the selected approach. The detectors are read every minute. The accumulative detectors' counts are then used to develop the proposed logic. The simulation is run five times for each scenario and the data are obtained randomly from one of the runs. In all scenarios, the traffic signal cycle is kept fixed to 1 minute with each of the red and green cycles consisting of 30 seconds (0-30 Simulation Seconds is red and 30-60 Simulation Seconds is green). Also, the speed limit is set to 60 km/ h and the vehicle type percentages are set to: 80% cars, 15% busses and 5% trucks. The generated data sets are displayed in Table 4 below.

Table 4: Generated Datasets

Dataset Name	Volume	Incident Duration	
Dataset 1c	1000 vph	10 min	Datasets used for calibration
Dataset 2c	1000 vph	Varied	
Dataset 3c	(1000, 500, 1300) vph	Varied	
Dataset 1v	1000 vph	10 min	Datasets used for validation
Dataset 2v	1000 vph	Varied	
Dataset 3v	(1000, 500, 1300) vph	Varied	

The table above contains six datasets. Three of the datasets are used to generate the fuzzy models and the rest are used during the validation process. It should be noted that the first dataset of each stage (Dataset 1c, Dataset 1v) has a fixed incident duration time of 10 minutes, whereas the other datasets simulate different incident durations. Another important point to note is that the third dataset of each stage (Dataset 3c, Dataset 3v) has three different values for the approach volume.

Chapter 4: Fuzzy Models Development

This chapter describes the development of fuzzy models for incident detection in urban streets. It presents the inputs and the output of the developed fuzzy models. It also presents the linguistic terms used. Moreover, it discusses the use of Fuzzy C-Means Clustering technique in data clustering and presents the procedure followed to create the fuzzy rules.

Three fuzzy models are developed in this chapter. The models are generated using three different datasets (Dataset 1c, Dataset 2c and Dataset 3c).

4.1 Fuzzy Model Inputs

The first step in building a fuzzy model is to select and define the model's base variables. As mentioned earlier (chapter 3), each lane is equipped with three detectors that count the number of passing vehicles (cars, buses and trucks) during the simulation period. Each record of the extracted data consists of the accumulative count of each of the nine detectors and the corresponding incident status (incident or no-incident). In order to reduce the number of inputs, and accordingly, simplify the derivation of fuzzy rules, the differences in the detectors' readings are used as the model inputs. This will reduce the number of inputs from nine to six inputs and, as a result, it will also reduce the number of generated rules. Consequently, the inputs of the fuzzy models are as follows:

- US1 – MS1 (Accumulative count of the first lane upstream detector – Accumulative count of the first lane midstream detector)
- MS1 – DS1 (Accumulative count of the first lane midstream detector – Accumulative count of the first lane downstream detector)

- US2 – MS2 (Accumulative count of the second lane upstream detector – Accumulative count of the second lane midstream detector)
- MS2 – DS2 (Accumulative count of the second lane midstream detector – Accumulative count of the second lane downstream detector)
- US3 – MS3 (Accumulative count of the third lane upstream detector – Accumulative count of the third lane midstream detector)
- MS3 – DS3 (Accumulative count of the third lane midstream detector – Accumulative count of the third lane downstream detector)

4.2 Fuzzy Model output

In this research, fuzzy models are developed to detect incident status from the model inputs, and classify the resultant output either as “incident” or “no-incident”. The models will result in an index value. As the index value increases, the probability of incident increases as well. The model resulted values are in the interval [0,1]. The values are then rounded to either “0” or “1” based on the output numerical value with “0” corresponding to a “no-incident” status and “1” corresponding to an “incident” status.

4.3 Fuzzy Logic Model Structure

This section describes the structure of the fuzzy logic models. This structure is shown in Figure 7. The fuzzy logic models use six inputs and one output. The inputs are (US1 – MS1), (MS1 – DS1), (US2 – MS2), (MS2 – DS2), (US3 – MS3) and (MS3 – DS3). The output refers to the incident_status (index).

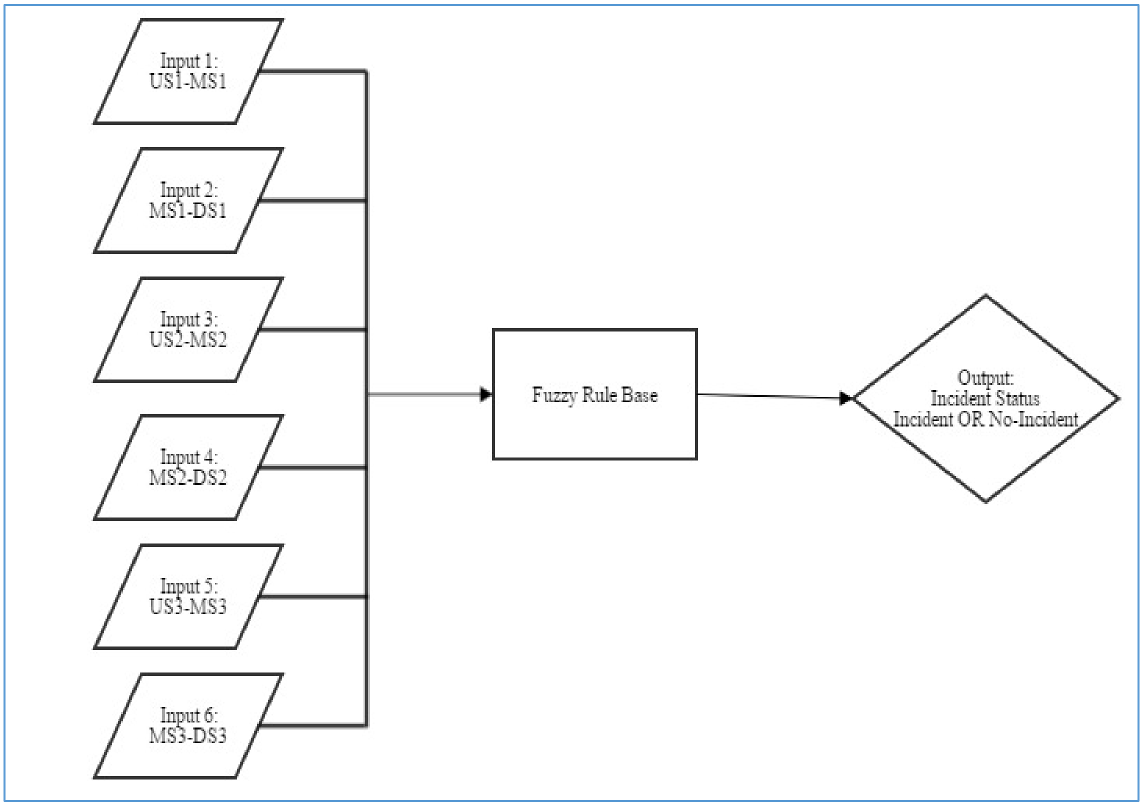


Figure 7: Fuzzy logic model structure

The flow chart describes the steps performed from left to right, processing six inputs that will eventually produce a single output. One of the important properties of a fuzzy logic system is the parallel nature of the rules. It means that the fuzzy rules will fire in parallel during the fuzzy inference process. The parallel firing of the rules enables considering the entire information content simultaneously (Mohaghegh, 2000). The steps adopted to develop the fuzzy logic system including membership functions establishment and fuzzy rule base (or Fuzzy Inference System "FIS") development are described in the following subsections.

4.4 Fuzzy Model Development Process

Fuzzy set theory does not confine the shape and the width of a fuzzy membership function to a single method; on the contrary, they can be determined using different methods, which means that there is no single method that should be adopted at all times. In general, a simple set of rules should be followed when creating membership functions for a fuzzy model. The set of rules are (Jantzen, 1998):

- Each membership function should be wide enough to cover all possible input and output values.
- An overlap should exist between the adjacent membership functions.
- Since the triangular shape is the simplest form of membership functions and they are easy to design, it is a good idea to start building the fuzzy model using this shape. It can be modified later based on the system performance.

The following steps describe the process followed to generate membership functions and if-then rules for the fuzzy models in this thesis:

- Step 1: The inputs and output spaces are divided into several fuzzy subsets and linguistic terms should be assigned to them.
- Step 2: The linguistic terms assigned in Step 1 are used in generating the fuzzy rules.
- Step 3: Similar fuzzy rules are merged.
- Step 4: The conflicting fuzzy rules are then counted and those with the highest number of counts remain in the system; whereas others are deleted.

4.4.1 Development of Fuzzy Model 1 (FM1)

PTV VISSIM was employed to generate about 30 different incident scenarios. These incidents were generated for a link length of 300 m, a volume of 1000 vph and an incident duration of 10 minutes. All scenarios were combined into a single dataset "Dataset 1c" to be used in the development process of FM1. Table 5 illustrates a sample of the entries in the combined data file. Each record of the dataset displays the values of the input variables and the corresponding output value. Output values are either "0" which represents a "no-incident" status or "1" which represents an "incident" status.

Table 5: Sample of "Dataset 1c" entries

Time (min.)	US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status
1	0	2	1	1	1	0	1
2	0	1	0	4	1	-5	1
3	1	0	-1	5	1	-4	1
4	-1	-1	2	6	2	-2	1
5	-2	-4	4	6	3	1	1
6	-2	-6	5	4	-1	3	1
7	-2	-7	5	5	-2	5	1
8	-2	-10	4	7	-1	6	1
9	-3	-11	6	8	1	5	1
10	-4	-15	8	10	0	10	1
11	-5	-21	9	13	2	11	1
12	-6	-24	8	15	0	11	0
13	-6	-24	7	18	-1	14	0
14	-4	-25	8	14	1	12	0
15	-6	-22	9	12	1	12	0
16	-5	-22	8	12	0	11	0
17	-6	-22	7	13	0	12	0
18	-6	-23	8	11	-1	12	0
19	-5	-23	8	12	0	13	0
20	-4	-24	8	11	-1	13	0

4.4.1.1 Generating Membership Functions (MFs)

Developing a fuzzy model requires defining all input and output variables. The process includes the definition of the linguistic variables, linguistic terms and their ranges, and membership value, μ, where μ represents the degree of confidence that a specific value belongs to a particular linguistic term (Hawas, 2007).

The definition of all input and output variables is displayed in Table 6. Real values were transformed into linguistic values that were represented by the linguistic terms. The values of each input variable (representing two detectors' accumulative counts difference) was divided into three linguistic terms; Zero, Positive and Very-Positive. The output variable (incident_status; represents an index to indicate the possibility of the incident) was divided into two linguistic terms; Low (low probability of incident) and High (high probability of incident).

Table 6: Definitions of FL input / output variables

Variable Type	Variable Name	Term Names	Unit
Input 1	US1-MS1	Zero (Z) Positive (P) Very-Positive (VP)	Vehicles
Input 2	MS1-DS1	Zero (Z) Positive (P) Very-Positive (VP)	Vehicles
Input 3	US2-MS2	Zero (Z) Positive (P) Very-Positive (VP)	Vehicles
Input 4	MS2-DS2	Zero (Z) Positive (P) Very-Positive (VP)	Vehicles
Input 5	US3-MS3	Zero (Z) Positive (P)	Vehicles

		Very-Positive (VP)	
Input 6	MS3-DS3	Zero (Z) Positive (P) Very-Positive (VP)	Vehicles
Output	Incident_Status	Low High	Index

After defining the linguistic terms for each variable, the range and membership value "µ" of each term should be defined. Fuzzy C-Means clustering technique was used with the help of MATLAB to split each input into three clusters: Zero (Z), Positive (P) and Very-Positive (VP). The following MATLAB command was used to cluster the first input "US1-MS1" into three clusters:

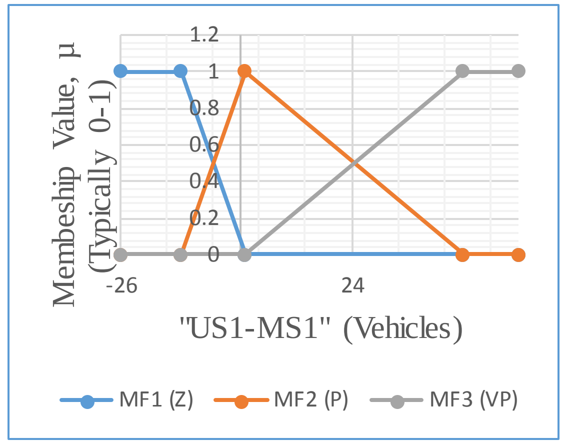
```
I = importdata('US1-MS1.dat');

[center,U,obj_fcn] = fcm(I,3)
```

This step produced three clusters with center values -13, 1, and 48. Each cluster was used to create a membership function. Cluster 1 represents the first membership function "Z", cluster 2 represents the second membership function "P", and cluster 3 represents the third membership function "VP". Since it is the simplest form of the membership functions to start with, triangular membership function shape was selected to represent all membership functions (Z, P and VP) and the center value of each cluster was used as a center value for its corresponding membership function. Then the shapes of "Z" and "VP" were changed to trapezoidal form to cover a wider range of input values.

Table 7 shows the "US1-MS1" membership function and the corresponding ranges.

Table 7: Membership functions set for input “US1-MS1” with corresponding ranges

Input	Range	MF
US1-MS1	< 1	Z
	-13 to 48	P
	> 1	VP



As mentioned earlier, the µ value represents the degree of confidence that a specific value belongs to a term; for example, for “US1-MS1” of 14 vehicles, μ_Z is 0, μ_P is 0.72 and μ_{VP} is 0.28. This indicates that the “US1-MS1” of 14 belongs to the term Positive with 72% confidence and to the term Very-Positive with 28% confidence. It can be observed that $\mu_Z + \mu_P + \mu_{VP} = 0 + 0.72 + 0.28 = 1$. See Figure 8.

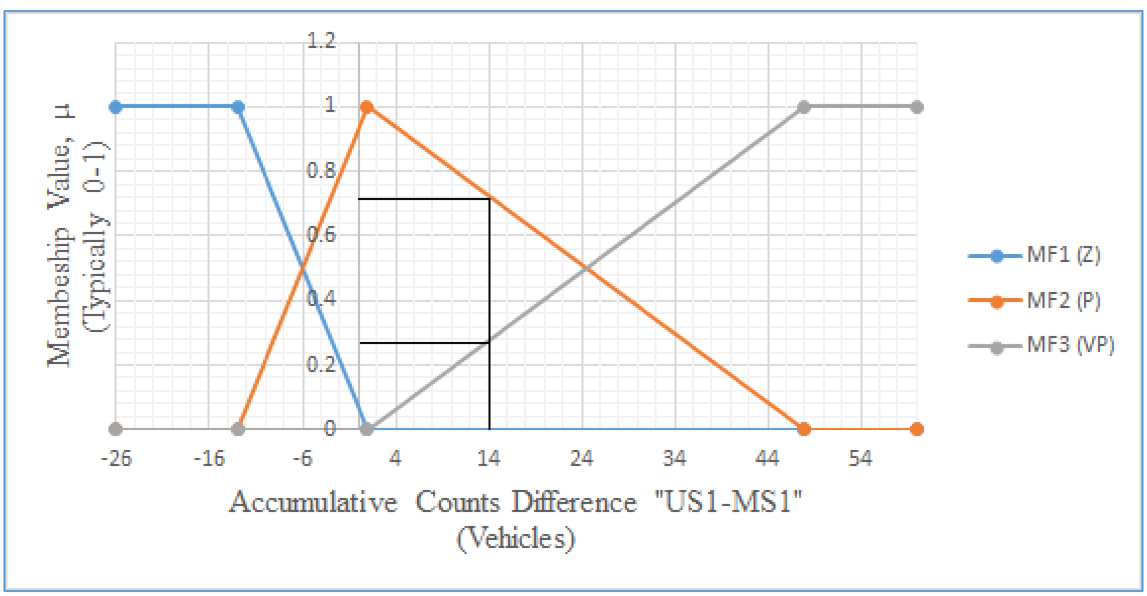
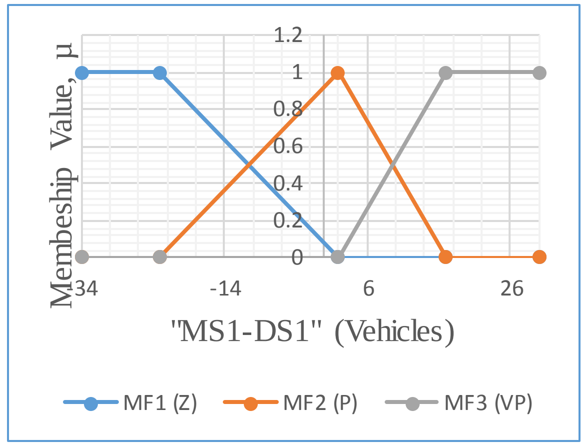


Figure 8: Derivation of µ value for “US1-MS1” of 14 vehicles

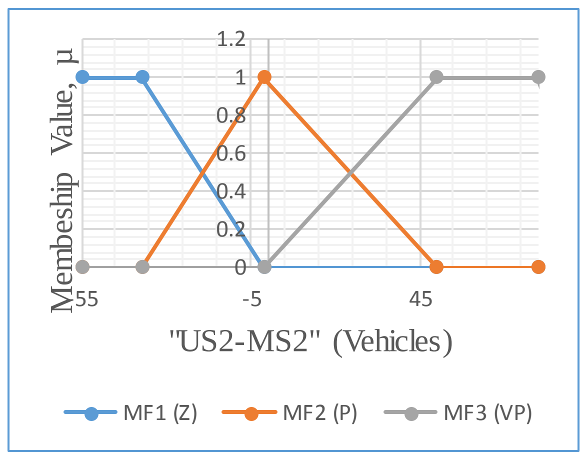
The same procedure was followed to create the membership functions for each input as shown in Table 8.

Table 8: MFs for the rest of the inputs with the corresponding ranges

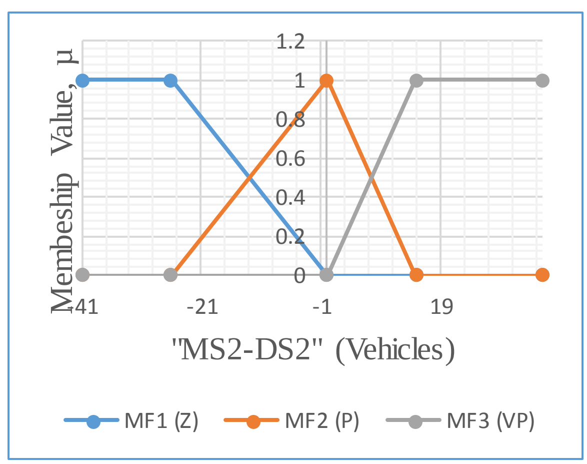
Input	Range	MF
MS1-DS1	< 2	Z
	-23 to 17	P
	> 2	VP



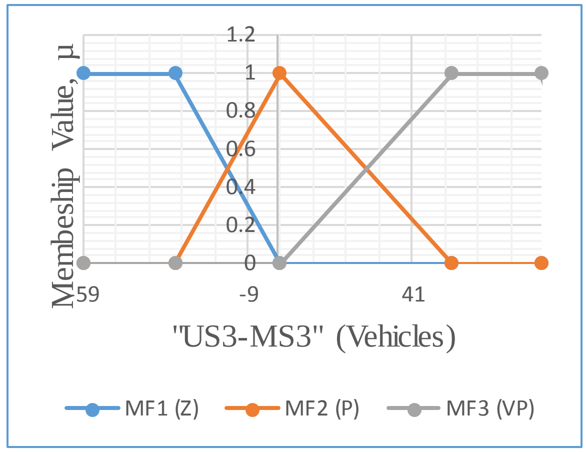
Input	Range	MF
US2-MS2	< -1	Z
	-37 to 50	P
	> -1	VP



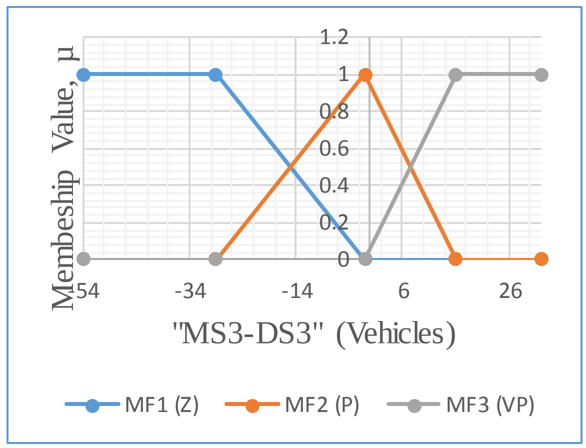
Input	Range	MF
MS2-DS2	< 0	Z
	-26 to 15	P
	> 0	VP



Input	Range	MF
US3-MS3	< 1	Z
	-31 to 53	P
	> 1	VP



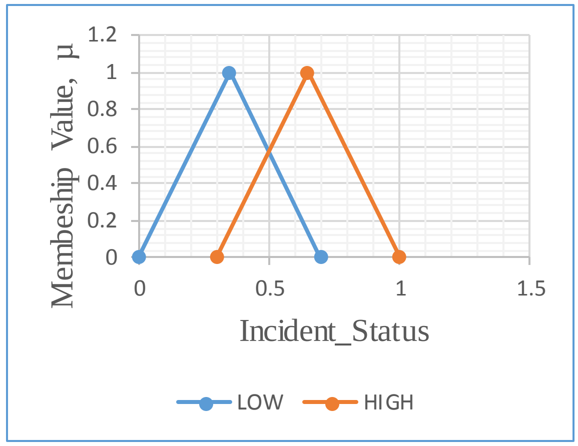
Input	Range	MF
MS3-DS3	< -1	Z
	-29 to 16	P
	> -1	VP



The output variable (incident_status) was represented by two membership functions that were chosen to be of triangular shape. “LOW” membership function represents low probability of an incident and “HIGH” membership function represents high probability of an incident. Since the incident_status represents a probability of an incident, its value should be in the interval [0, 1]. Table 9 shows the membership functions set for the output variable; incident_status.

Table 9: MFs for the output "Incident_Status" with the corresponding ranges

Input	Range	MF
Incident_Status	0 to 0.7	LOW
	0.3 to 1	HIGH



4.4.1.2 Generating Fuzzy Rules

The rule base is considered as the core of the fuzzy model since the quality of results in a fuzzy system depends mainly on these rules. To generate the fuzzy rules set, each numerical input and output was assigned to the membership function with the maximum membership grade (maximum μ value). For example, "US1-MS1" of 14 was considered to belong to membership function "P", "MS2-DS2" of 10 to membership function "VP", and "US3-MS3" of -39 to membership function "Z". Table 10 shows a sample of the fuzzified data file "Dataset 1c".

Table 10: Sample of the fuzzified data file "Dataset 1c"

US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status
P	P	P	P	P	P	HIGH
P	P	P	P	P	P	HIGH
P	P	P	P	P	P	HIGH
P	P	P	P	P	P	HIGH
P	P	P	P	P	P	HIGH
P	P	P	P	P	P	HIGH

P	P	P	P	P	P	HIGH
P	P	P	P	P	P	HIGH
P	Z	P	VP	P	P	HIGH
P	Z	P	VP	P	VP	HIGH
P	Z	P	VP	P	VP	HIGH
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW
P	Z	P	VP	P	VP	LOW

Now, a single rule was obtained from each input-output data pair. As such, each row represents "IF…AND…THEN…" rule. The "IF…AND…" part was generated from the input data, and the "THEN…" part was generated from the output data. For example, the rule represented by the highlighted row in Table 10 above was:

IF

"US1-MS1" is "P"

AND

"MS1-DS1" is "Z"

AND

"US2-MS2" is "P"

AND

"MS2-DS2" is "VP"

AND

"US3-MS3" is "P"

AND

“MS3-DS3” is “VP”

THEN

“Incident_Status” is “LOW”

The rules generated here were “AND” rules because of the dependency of the inputs on each other. A total of 600 rules were generated. Initial rules were generated in a similar fashion for all data pairs and they are given in Table 11. The last column “Count” shows how many times each rule was repeated throughout the dataset.

Table 11: Initial rules generated from data file “Dataset 1c”

US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status	Count
P	P	P	P	P	P	LOW	10
P	P	P	P	P	P	HIGH	144
P	Z	P	VP	P	P	HIGH	6
P	Z	P	VP	P	VP	LOW	16
P	Z	P	VP	P	VP	HIGH	9
P	P	P	VP	P	P	LOW	13
P	P	P	VP	P	P	HIGH	15
P	Z	P	VP	Z	VP	LOW	2
P	VP	P	VP	Z	P	HIGH	1
P	P	P	VP	Z	P	HIGH	1
P	P	P	VP	Z	Z	HIGH	4
P	VP	P	VP	Z	Z	LOW	9
VP	Z	P	VP	P	VP	LOW	4
VP	Z	P	VP	P	VP	HIGH	2
VP	Z	Z	VP	P	VP	LOW	13
VP	Z	Z	VP	P	VP	HIGH	4
P	P	P	P	P	VP	LOW	12
P	P	P	P	P	VP	HIGH	14
VP	Z	P	P	P	VP	HIGH	3
VP	Z	Z	P	Z	VP	HIGH	1
VP	Z	Z	VP	Z	VP	LOW	19
VP	Z	Z	VP	Z	VP	LOW	2

VP	Z	Z	VP	P	P	LOW	8
VP	Z	Z	VP	P	P	HIGH	7
VP	P	P	P	P	P	HIGH	1
VP	Z	Z	VP	Z	P	HIGH	2
VP	P	Z	VP	P	P	LOW	1
VP	P	Z	VP	P	P	HIGH	5
P	VP	P	P	P	P	LOW	4
P	VP	P	P	P	P	HIGH	9
P	VP	P	Z	P	P	LOW	20
P	VP	P	Z	P	P	HIGH	8
P	VP	P	P	P	Z	LOW	2
P	VP	P	P	P	Z	HIGH	4
P	VP	P	Z	P	VP	LOW	7
P	VP	P	Z	P	VP	HIGH	5
P	P	P	Z	P	VP	LOW	2
P	P	P	Z	P	VP	HIGH	4
Z	VP	P	Z	P	VP	LOW	4
Z	VP	P	Z	P	VP	HIGH	1
Z	P	P	Z	P	VP	LOW	1
P	VP	VP	Z	Z	VP	LOW	18
P	VP	VP	Z	Z	VP	HIGH	6
P	P	P	Z	P	P	HIGH	1
P	P	VP	Z	Z	VP	LOW	1
P	P	VP	Z	Z	VP	HIGH	4
Z	P	VP	Z	Z	VP	LOW	2
Z	P	VP	Z	Z	VP	HIGH	7
Z	VP	VP	Z	Z	VP	LOW	16
Z	VP	VP	Z	Z	VP	HIGH	4
P	P	VP	P	Z	VP	LOW	4
P	P	VP	P	Z	VP	HIGH	1
P	P	P	Z	Z	VP	HIGH	1
P	P	P	P	Z	P	HIGH	1
P	P	VP	P	Z	P	HIGH	2
P	VP	VP	P	Z	P	LOW	3
P	VP	VP	P	Z	P	HIGH	2
Z	VP	VP	P	Z	P	LOW	1
P	P	P	VP	P	Z	LOW	1
P	P	P	VP	P	Z	HIGH	1
P	VP	P	VP	P	Z	LOW	13

P	VP	P	VP	P	Z	HIGH	3
Z	VP	P	VP	P	Z	LOW	13
Z	VP	P	VP	P	Z	HIGH	6
Z	P	P	P	P	P	HIGH	3
Z	VP	P	P	P	Z	HIGH	6
Z	VP	P	VP	VP	Z	LOW	5
Z	VP	P	VP	VP	Z	HIGH	4
Z	VP	P	P	VP	Z	LOW	4
Z	P	P	VP	P	P	HIGH	1
Z	P	P	VP	P	Z	HIGH	2
Z	P	P	VP	VP	Z	HIGH	1
Z	VP	Z	VP	VP	Z	LOW	10
Z	VP	Z	VP	VP	Z	HIGH	2
P	P	Z	VP	P	Z	HIGH	1
P	P	Z	VP	VP	Z	HIGH	3
Z	P	Z	VP	VP	Z	LOW	25
Z	P	Z	VP	VP	Z	HIGH	4
P	P	Z	P	VP	P	HIGH	1
P	P	Z	VP	VP	P	HIGH	5
Z	P	Z	VP	VP	P	LOW	1
Z	P	Z	VP	VP	P	HIGH	4
P	P	Z	VP	P	P	HIGH	1
P	Z	P	P	P	VP	LOW	6
P	Z	P	P	P	VP	HIGH	1

From Table 10 and Table 11, it can be observed that there were many similar rules and many conflicting rules. Similar rules can be defined as the rules that have the same antecedents and consequents while conflicting rules can be defined as the rules that have the same antecedents but different consequents. Similar rules were replaced by a single rule. The conflicting fuzzy rules were counted and those with the highest number of counts remained in the system while others were deleted. For example, below are conflicting rules extracted from Table 11.

P	Z	P	VP	P	VP	LOW	16
P	Z	P	VP	P	VP	HIGH	9

Both rules have same antecedents (IF "US1-MS1" is "P" AND "MS1-DS1" is "Z" AND "US2-MS2" is "P" AND "MS2-DS2" is "VP" AND "US3-MS3" is "P" AND "MS3-DS3" is "VP") but different consequents (THEN "Incident_Status" is "HIGH", and THEN "Incident_Status" is "LOW"). After counting, it has been found that the first rule was repeated 16 times through the dataset and the second rule was repeated 9 times only. This has led to favor the first rule over the second one. Therefore, the rule with "HIGH" probability of incident was added to the rule set and the one with "LOW" probability was deleted. It should also be noted that the rules that appear once were not taken in consideration because of their small effect. This procedure resulted in a minimized rules set (41 rules) as shown in Table 12.

Table 12: Minimized rules set for FM1

Rule #	US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status
1	P	P	P	P	P	P	HIGH
2	P	Z	P	VP	P	P	HIGH
3	P	Z	P	VP	P	VP	LOW
4	P	P	P	VP	P	P	HIGH
5	P	VP	P	VP	Z	Z	LOW
6	VP	Z	P	VP	P	VP	LOW
7	VP	Z	Z	VP	P	VP	LOW
8	P	P	P	P	P	VP	HIGH
9	VP	Z	Z	VP	Z	VP	LOW
10	VP	Z	Z	VP	P	P	LOW
11	VP	P	Z	VP	P	P	HIGH
12	P	VP	P	P	P	P	HIGH
13	P	VP	P	Z	P	P	LOW
14	P	VP	P	P	P	Z	HIGH
15	P	VP	P	Z	P	VP	LOW

16	P	P	P	Z	P	VP	HIGH
17	Z	VP	P	Z	P	VP	LOW
18	P	VP	VP	Z	Z	VP	LOW
19	P	P	VP	Z	Z	VP	HIGH
20	Z	P	VP	Z	Z	VP	HIGH
21	Z	VP	VP	Z	Z	VP	LOW
22	P	P	VP	P	Z	VP	LOW
23	P	VP	VP	P	Z	P	LOW
24	P	VP	P	VP	P	Z	LOW
25	Z	VP	P	VP	P	Z	LOW
26	Z	VP	P	P	P	Z	HIGH
27	Z	VP	P	VP	VP	Z	LOW
28	Z	VP	Z	VP	VP	Z	LOW
29	Z	P	Z	VP	VP	Z	LOW
30	P	P	Z	VP	VP	P	HIGH
31	Z	P	Z	VP	VP	P	HIGH
32	P	Z	P	P	P	VP	LOW
33	P	Z	P	VP	Z	VP	LOW
34	P	P	P	VP	Z	Z	HIGH
35	VP	Z	P	P	P	VP	HIGH
36	VP	Z	Z	VP	Z	P	HIGH
37	P	P	VP	P	Z	P	HIGH
38	Z	P	P	P	P	P	HIGH
39	Z	VP	P	P	VP	Z	LOW
40	Z	P	P	VP	P	Z	HIGH
41	P	P	Z	VP	VP	Z	HIGH

The minimized rules of Table 12 were implemented with Fuzzy Logic Toolbox of MATLAB (MathWorks, Fuzzy Inference Process, 2015) using Mamdani inference mechanism. The results obtained with these rules are to be discussed in chapter 5 “Results and Discussion”.

4.4.2 Development of Fuzzy Model 2 (FM2)

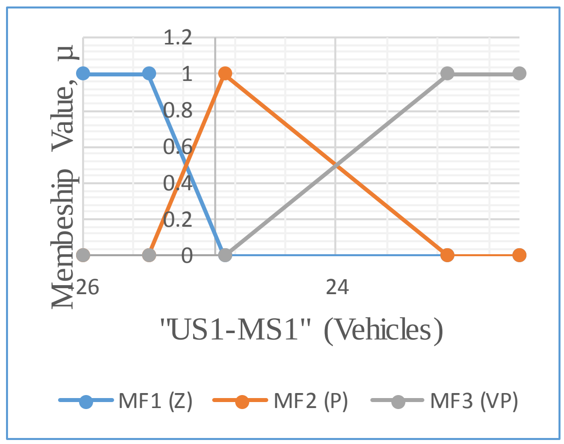
PTV VISSIM was used to generate a second set of incident scenarios (about 43 scenarios). These incidents were generated for link length of 300 m, volume of 1000 vph and different incident durations. All scenarios were combined into a single dataset “Dataset 2c” to be used in the development process of FM2.

4.4.2.1 Generating Membership Functions

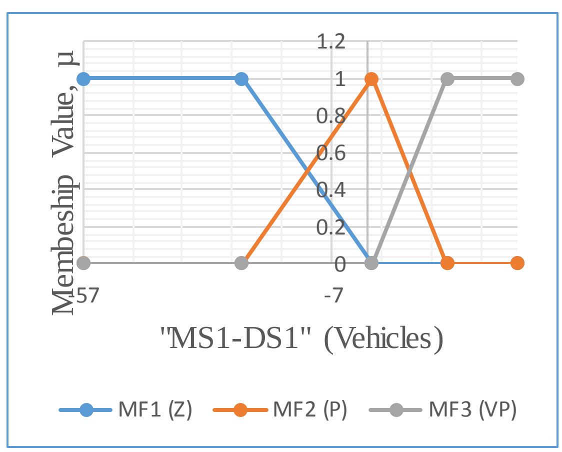
Membership Functions for inputs and output of FM2 are displayed in Table 13.

Table 13: Membership functions for FM2 inputs and output with corresponding ranges

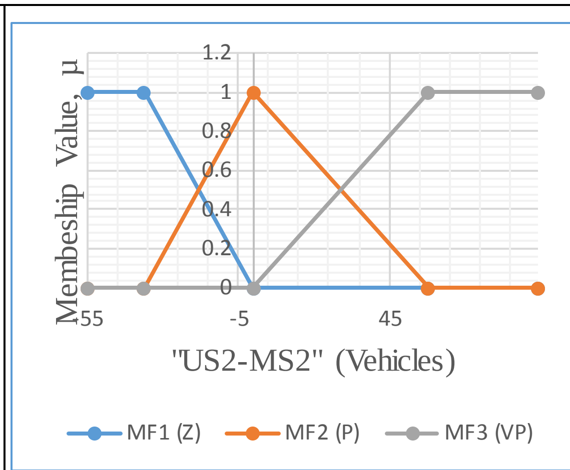
Input	Range	MF
US1-MS1	< 2	Z
	-13 to 46	P
	> 2	VP



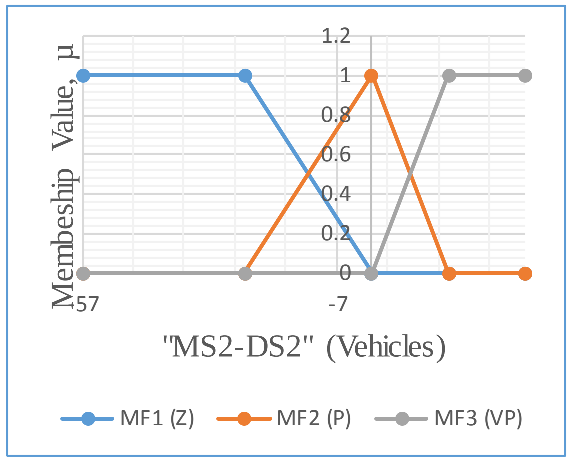
Input	Range	MF
MS1-DS1	< 1	Z
	-25 to 16	P
	> 1	VP



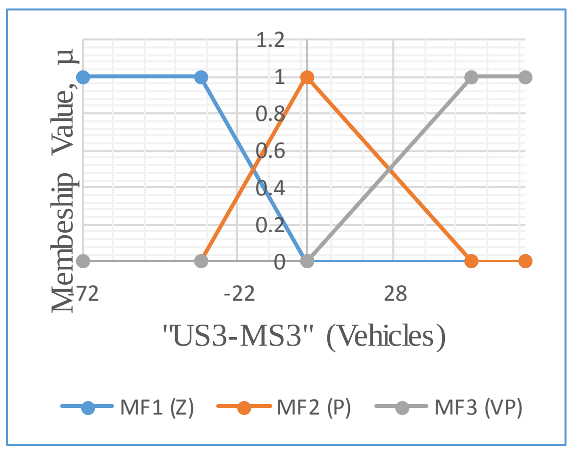
Input	Range	MF
US2-MS2	< 0	Z
	-36 to 58	P
	> 0	VP



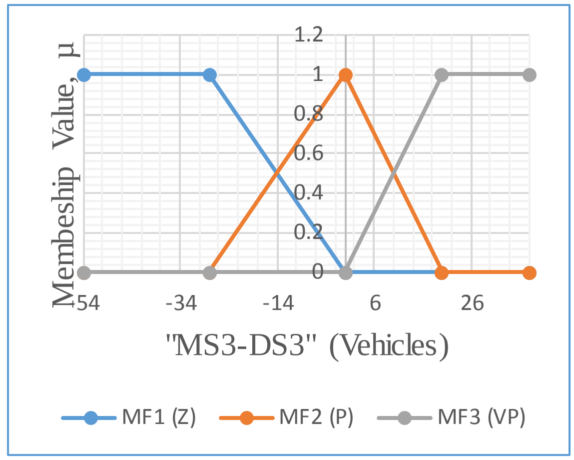
Input	Range	MF
MS2-DS2	< 0	Z
	-25 to 15	P
	> 0	VP



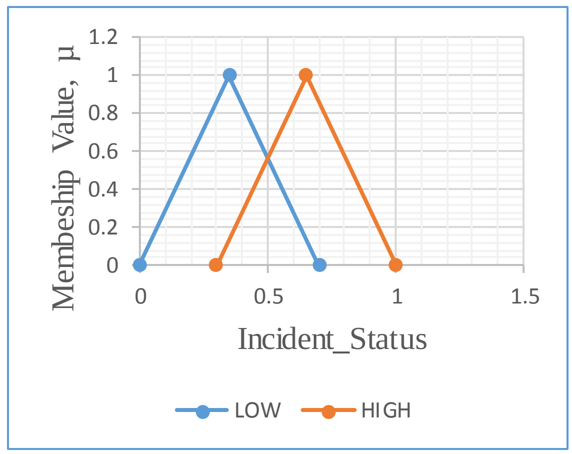
Input	Range	MF
US3-MS3	< 0	Z
	-34 to 53	P
	> 0	VP



Input	Range	MF
MS3-DS3	< 0	Z
	-28 to 20	P
	> 0	VP



Input	Range	MF
Incident_Status	0 to 0.7	LOW
	0.3 to 1	HIGH



It can be observed that the membership functions (MFs) for both FM1 and FM2 are very similar since the datasets used to develop these two models (Dataset 1c and Dataset 2c) had the same volume (1000 vph). Therefore, it can be said that the membership functions in the proposed methodology are highly dependent on the traffic volume.

4.4.2.2 Generating Fuzzy Rules

The same process followed to generate the fuzzy rules set of FM1 was also followed to generate the rules set of FM2. The resulted minimized rules set for FM2 consists of 51 rules displayed in in Table 14.

Table 14: Minimized rules set for FM2

Rule #	US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status
1	P	P	P	P	P	P	HIGH
2	P	Z	P	VP	P	P	HIGH
3	P	Z	P	VP	P	VP	LOW
4	P	P	P	VP	P	P	HIGH
5	P	P	Z	VP	P	P	HIGH
6	P	P	Z	VP	P	Z	HIGH
7	Z	P	Z	VP	VP	Z	LOW
8	Z	VP	Z	VP	VP	Z	LOW
9	P	P	P	P	P	VP	HIGH
10	P	Z	P	P	P	VP	LOW
11	Z	P	P	P	P	P	HIGH
12	Z	P	VP	Z	Z	VP	HIGH
13	Z	VP	VP	Z	Z	VP	LOW
14	P	P	P	Z	P	VP	HIGH
15	P	P	Z	Z	P	VP	HIGH
16	P	Z	P	Z	P	VP	LOW
17	P	P	Z	P	P	P	HIGH
18	VP	P	Z	P	P	P	HIGH
19	VP	P	Z	VP	P	P	HIGH
20	Z	VP	P	P	P	P	HIGH
21	Z	VP	P	VP	P	Z	LOW
22	Z	VP	P	VP	VP	Z	HIGH
23	P	VP	Z	Z	P	P	HIGH
24	Z	VP	P	P	P	Z	HIGH
25	Z	Z	P	VP	P	VP	LOW
26	P	VP	P	P	P	P	HIGH
27	P	VP	P	VP	Z	P	HIGH
28	P	VP	P	VP	Z	Z	LOW

29	VP	Z	Z	VP	P	VP	LOW
30	VP	Z	Z	P	P	VP	HIGH
31	VP	Z	Z	VP	Z	VP	LOW
32	VP	Z	Z	VP	P	P	LOW
33	VP	Z	Z	VP	Z	P	HIGH
34	P	VP	P	Z	P	P	LOW
35	P	VP	P	P	P	Z	HIGH
36	P	P	P	Z	P	P	HIGH
37	P	VP	P	Z	P	VP	HIGH
38	Z	P	P	Z	P	VP	LOW
39	Z	VP	P	Z	P	VP	LOW
40	P	VP	VP	Z	Z	VP	LOW
41	P	P	VP	Z	Z	VP	HIGH
42	P	VP	P	P	Z	P	HIGH
43	P	VP	VP	P	Z	P	LOW
44	Z	VP	VP	Z	Z	P	LOW
45	P	VP	VP	Z	Z	P	LOW
46	P	VP	P	VP	P	Z	LOW
47	Z	VP	P	P	VP	Z	LOW
48	Z	P	P	VP	P	P	HIGH
49	Z	P	P	VP	P	Z	HIGH
50	P	P	Z	VP	VP	P	HIGH
51	Z	P	Z	VP	VP	P	HIGH

The minimized rules of Table 14 were implemented with Fuzzy Logic Toolbox of MATLAB (MathWorks, Fuzzy Inference Process, 2015) using Mamdani inference mechanism. The results obtained with these rules are to be discussed in chapter 5 “Results and Discussion”.

4.4.3 Development of Fuzzy Model 3 (FM3)

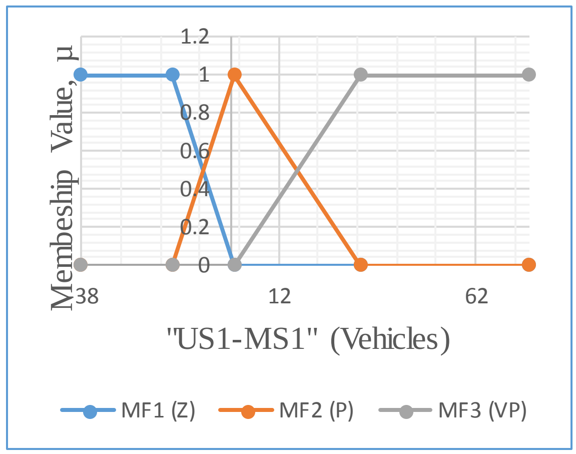
PTV VISSIM was also used to generate a third set of incident scenarios (about 40 scenarios). These incidents were generated for link length of 300 m, volumes of (500, 1000 and 1300) vph and different incident durations. All scenarios were combined into a single dataset "Dataset 3c" to be used in the development process of FM3.

4.4.3.1 Generating Membership Functions

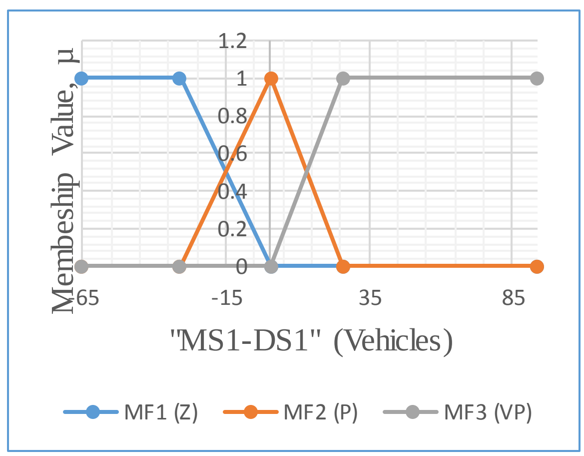
Membership Function sets for inputs and output of FM3 are displayed in Table 15.

Table 15: Membership functions for FM3 inputs with corresponding ranges

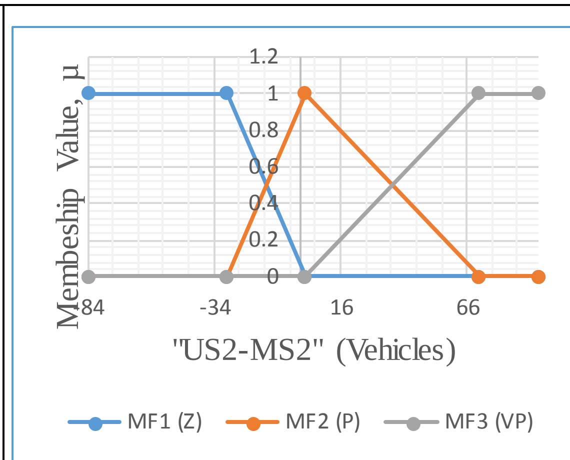
Input	Range	MF
US1-MS1	< 1	Z
	-15 to 33	P
	> 1	VP



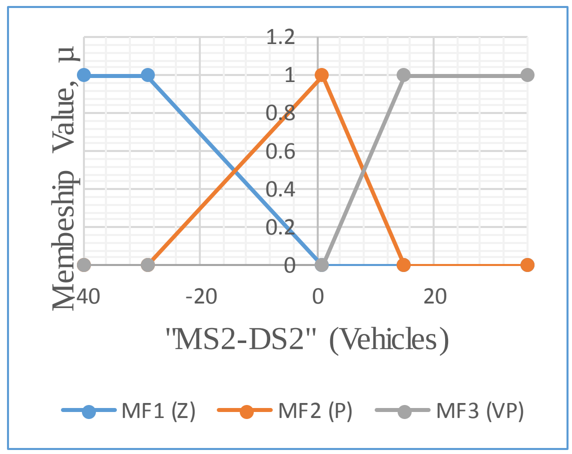
Input	Range	MF
MS1-DS1	< 1	Z
	-31 to 26	P
	> 1	VP



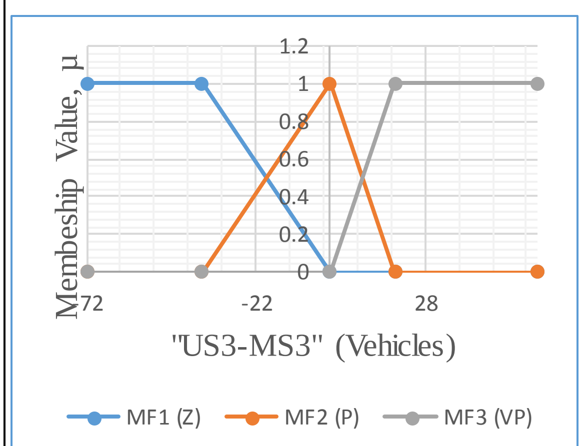
Input	Range	MF
US2-MS2	< 2	Z
	-29 to 71	P
	> 2	VP



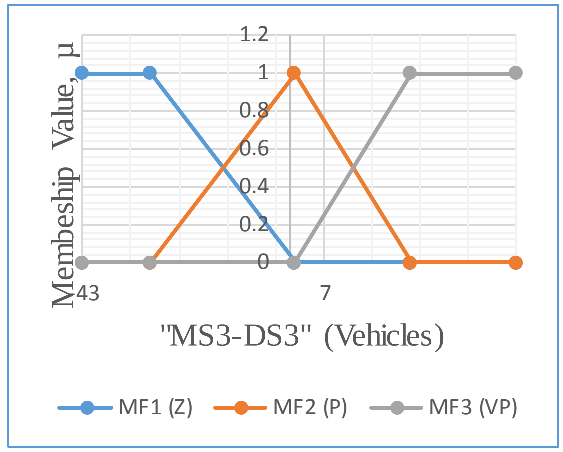
Input	Range	MF
MS2-DS2	< 1	Z
	-29 to 15	P
	> 1	VP



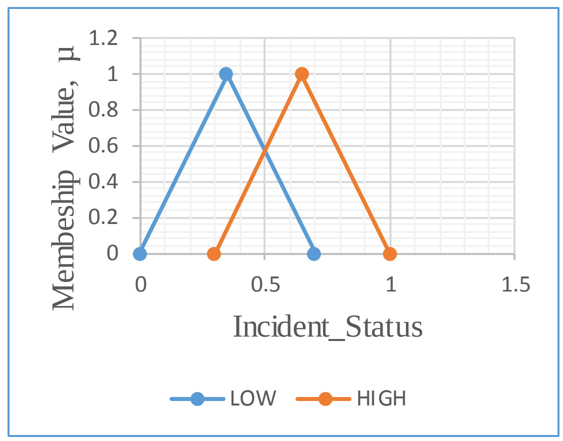
Input	Range	MF
US3-MS3	< 0	Z
	-38 to 19	P
	> 0	VP



Input	Range	MF
MS3-DS3	< 1	Z
	-29 to 25	P
	> 1	VP



Input	Range	MF
Incident_Status	0 to 0.7	LOW
	0.3 to 1	HIGH



It can be observed from Table 15 above that the membership function sets generated for FM3 had different ranges than the ones obtained for FM1 and FM2. This was because of the different volumes (500-1000-1300) used in generating the dataset for this model.

4.4.3.2 Generating Fuzzy Rules

The minimized rules set for FM3 consists of 48 rules displayed in Table 16.

Table 16: Minimized rules for FM3

Rule #	US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status
1	P	P	P	P	P	P	HIGH
2	P	P	P	VP	P	P	LOW
3	P	Z	P	VP	P	P	HIGH
4	P	Z	P	VP	P	VP	LOW
5	P	P	Z	VP	VP	P	HIGH
6	P	P	Z	VP	VP	Z	HIGH
7	Z	P	Z	VP	VP	Z	LOW
8	P	P	P	P	P	VP	HIGH
9	P	Z	P	P	P	VP	LOW
10	Z	P	P	P	P	P	HIGH
11	Z	P	VP	Z	Z	VP	HIGH
12	Z	VP	VP	Z	Z	VP	LOW
13	VP	Z	P	VP	P	P	HIGH
14	VP	Z	Z	VP	P	P	HIGH
15	VP	Z	Z	VP	P	VP	LOW
16	P	VP	VP	Z	Z	VP	LOW
17	P	P	Z	P	P	P	HIGH
18	P	P	Z	Z	P	P	HIGH
19	P	P	P	Z	P	VP	HIGH
20	VP	P	Z	VP	P	P	LOW
21	VP	P	Z	P	P	P	HIGH
22	P	P	P	VP	VP	Z	LOW
23	P	P	Z	P	VP	P	HIGH
24	Z	P	P	VP	VP	Z	HIGH
25	VP	P	Z	Z	VP	P	HIGH
26	VP	Z	P	VP	Z	VP	LOW
27	Z	VP	P	P	P	Z	LOW
28	Z	VP	P	VP	P	Z	LOW
29	P	Z	P	P	VP	P	HIGH
30	VP	Z	P	P	VP	P	LOW
31	P	VP	P	P	P	P	HIGH
32	P	VP	Z	P	P	P	LOW
33	P	Z	P	P	P	P	HIGH
34	Z	Z	P	P	VP	P	LOW
35	P	VP	Z	Z	P	VP	LOW

36	P	VP	Z	Z	VP	VP	LOW
37	P	P	Z	Z	VP	VP	HIGH
38	VP	P	Z	Z	P	VP	HIGH
39	VP	VP	Z	Z	P	VP	LOW
40	VP	Z	Z	Z	VP	P	LOW
41	P	Z	P	VP	Z	VP	LOW
42	P	P	P	P	Z	VP	LOW
43	P	P	P	VP	Z	VP	LOW
44	P	VP	Z	P	P	VP	HIGH
45	P	VP	Z	P	Z	VP	LOW
46	Z	VP	Z	P	Z	VP	LOW
47	Z	P	P	VP	P	Z	LOW
48	Z	Z	VP	VP	VP	Z	LOW

The minimized rules of Table 16 were implemented with Fuzzy Logic Toolbox of MATLAB (MathWorks, Fuzzy Inference Process, 2015) using Mamdani inference mechanism. The results obtained with these rules are to be discussed in chapter 5 “Results and Discussion”.

Chapter 5: Results and Discussion

In this chapter, the fuzzy logic process will be discussed and an example from this thesis will be displayed in details. Next, the results of implementing three fuzzy models (FM1, FM2 and FM3) will be presented and discussed. Each model will be validated with a new dataset generated using the software “PTV VISSIM”. FM1, FM2 and FM3 will be validated using “Dataset 1v”, “Dataset 2v” and “Dataset 3v”, respectively. The models will be evaluated using three main measures: good_detection rate, no_detection rate and false_alarm rate. Finally, a comparison between the results obtained will be presented and an additional trial to improve the performance will be discussed.

5.1 Fuzzy Logic Process

In this section, an overview of the fuzzy logic process will be presented with a detailed example. A fuzzy logic process can be defined as a way of mapping an input space to an output space using a collection of fuzzy membership functions and rules. There are three main parts that make up the fuzzy logic process. These parts are: Fuzzification, Fuzzy Inference System (FIS), which includes the membership functions and the fuzzy rules, and De-Fuzzification. The fuzzy logic components are shown in Figure 9. (G. V. S. Raju, Jun Zhou , 1990)

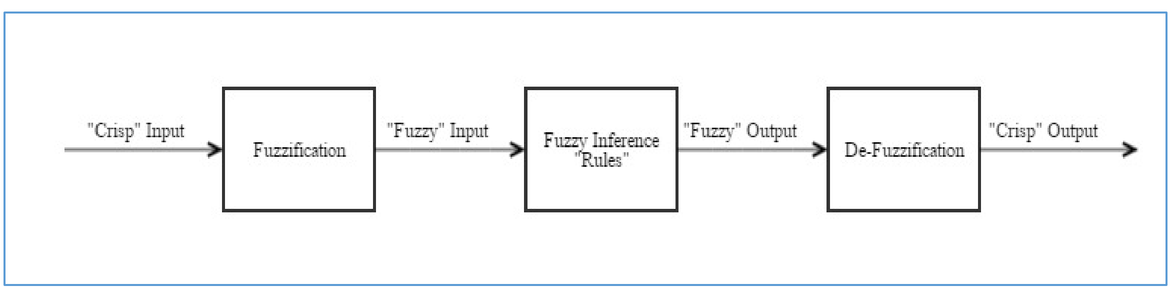


Figure 9: Fuzzy logic process

The fuzzy logic process can be explained as follows:

- The first step is to fuzzify the inputs, i.e., converting the crisp input values to linguistic values using fuzzy linguistic terms and membership functions. This step is called "Fuzzification".
- The second step is to apply the fuzzy rules to the inputs and find the resultant fuzzy output value. This step is known as the "Fuzzy Inference".
- The last step is to map the fuzzy value of the output to a crisp value, which is known as "De-Fuzzification".

Mamdani inference (Figure 10) was used in the designed systems (models). Since all inputs are dependent on each other, it is meaningful to use "AND" operator to combine the inputs. A single output was used in the system to return the incident_status every minute. The de-fuzzification process has resulted in a crisp index value that describes the probability of incident.

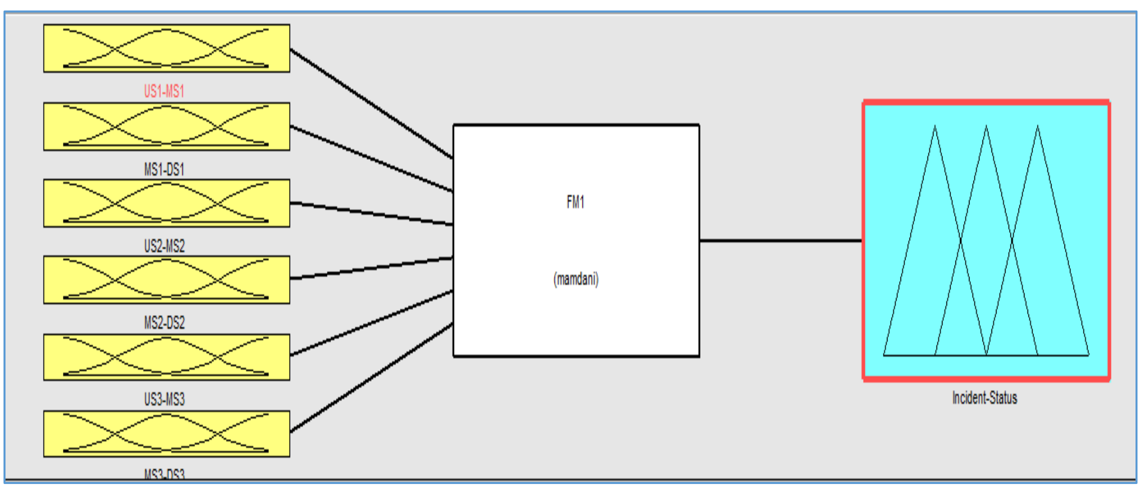


Figure 10: Mamdani FIS for the proposed models

In order to understand how the fuzzy logic works, the process is described in details in the following example.

In this example, FM1 is used and the following crisp input values are given:

"US1-MS1" = -4, "MS1-DS1" = -24,

"US2-MS2" = 8, "MS2-DS2" = 11,

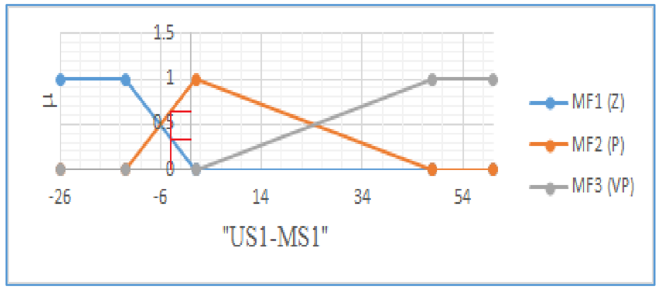
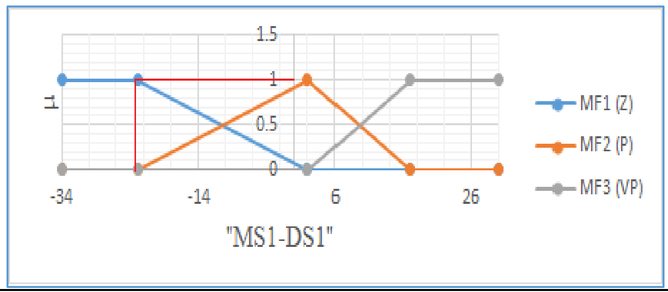
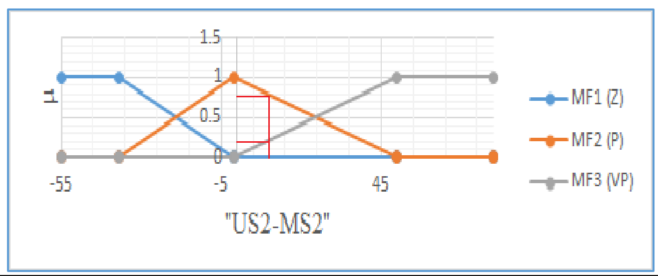
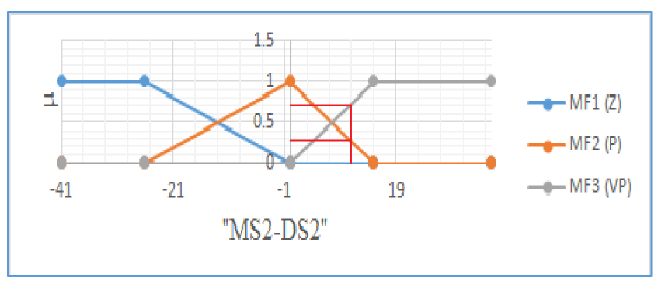
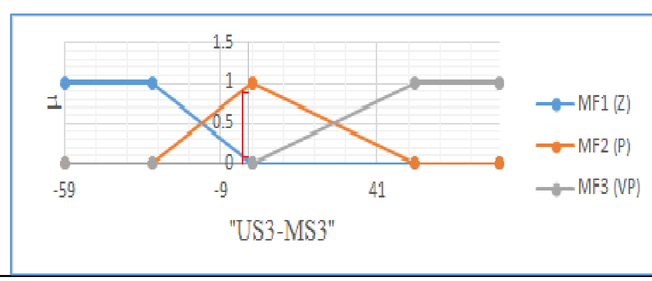
"US3-MS3" = -1, "MS3-DS3" = 13

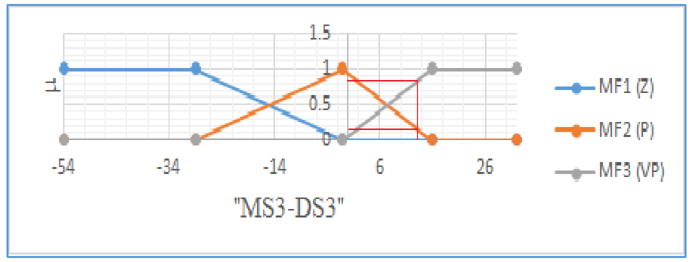
5.1.1 Step 1: Fuzzify Inputs

The first step is to convert crisp inputs to fuzzy. Each input in the previously built fuzzy model (FM1) has its own set of membership functions (each set consists of three membership functions: Z, P and VP). The given inputs in this example are compared with the related set of membership functions to determine the membership function that represents the crisp input. Each given input in this example may belong to one or two membership functions. "μ" value describes how strong each input belongs to each membership function. FM1 is built on 41 rules, and each of the rules requires converting the inputs into a number of different fuzzy linguistic terms. Before evaluating the rules, the inputs must be fuzzified according to each of these linguistic terms.

From row 1 in Table 17, it can be seen that "US1-MS1" of "-4" belongs to two membership functions (Z and P) with different degrees, that is $\mu_z = 0.35$ and $\mu_p = 0.65$. Also, "MS1-DS1" of "-24" entirely belongs to one membership function (Z) which means it has $\mu_z = 1$. Similarly, the rest of the inputs are assigned to their membership functions as shown in Table 17. The red lines on the figures show the derivation of "μ" values. Fuzzification step ends with all crisp inputs assigned to their membership functions required by the rules.

Table 17: "μ" values for each input

<table>
<tr><th>Input</th><th>Crisp Value</th><th colspan="2">μ values</th></tr>
<tr><td rowspan="3">US1-MS1</td><td rowspan="3">-4</td><td>$\mu_z = 0.35$</td><td rowspan="3">
</td></tr>
<tr><td>$\mu_p = 0.65$</td></tr>
<tr><td>$\mu_{VP} = 0$</td></tr>
<tr><td rowspan="3">MS1-DS1</td><td rowspan="3">-24</td><td>$\mu_z = 1$</td><td rowspan="3">
</td></tr>
<tr><td>$\mu_p = 0$</td></tr>
<tr><td>$\mu_{VP} = 0$</td></tr>
<tr><td rowspan="3">US2-MS2</td><td rowspan="3">8</td><td>$\mu_z = 0$</td><td rowspan="3">
</td></tr>
<tr><td>$\mu_p = 0.8$</td></tr>
<tr><td>$\mu_{VP} = 0.2$</td></tr>
<tr><td rowspan="3">MS2-DS2</td><td rowspan="3">11</td><td>$\mu_z = 0$</td><td rowspan="3">
</td></tr>
<tr><td>$\mu_p = 0.3$</td></tr>
<tr><td>$\mu_{VP} = 0.7$</td></tr>
<tr><td rowspan="3">US3-MS3</td><td rowspan="3">-1</td><td>$\mu_z = 0.1$</td><td rowspan="3">
</td></tr>
<tr><td>$\mu_p = 0.9$</td></tr>
<tr><td>$\mu_{VP} = 0$</td></tr>
</table>

<table>
<tr><td rowspan="3">MS3-DS3</td><td rowspan="3">13</td><td>$\mu_Z = 0$</td><td rowspan="3">
</td></tr>
<tr><td>$\mu_P = 0.15$</td></tr>
<tr><td>$\mu_{VP} = 0.85$</td></tr>
</table>

5.1.2 Step 2: Apply the Fuzzy Operator (AND)

After fuzzifying the inputs, the fuzzy operator “AND” is applied. The input to the fuzzy operator is six membership values from fuzzified input variables. Using “AND” operator results in selecting the minimum membership value of the antecedents. This truth-value represents the antecedents and it is then applied to the consequent membership function to get a single output fuzzy value.

In this example, only three rules (rule 2, rule 3 and rule 32) will result in non-zero values because none of the membership values of the antecedents is zero. Figure 11 shows MATLAB-Rule Viewer which displays “AND” operator evaluating the antecedents of rules 2, 3 and 32.

Figure 11: Rules evaluation in MATLAB-Rule Viewer

MATLAB-Rule Viewer in Figure 11 shows all the 41 rules of FM1. Each rule is represented by its relevant inputs and output membership functions. For example, rule 1 is represented by the input membership functions (P, P, P, P, P, P) and the output membership function (HIGH). Similarly, rule 2 is represented by the input membership functions (P, Z, P, VP, P, P) and the output membership function (HIGH). The colored parts of the membership functions represent the implementation of the fuzzy rules based on the given inputs in this example [-4, -24, 8, 11, -1 and 13]. In each rule, the involved parts of the input membership functions are yellow colored and the involved parts of the output membership functions are blue colored. By observing the blue colored parts in Figure 11, it can be noticed that only three rules from the set will contribute to the output estimation, namely, rule 2, rule 3 and rule 32 (Table 18). The implementation of the rest of the rules will return zero output. Now, for evaluating rule 2, the six inputs ((US1-MS1 is P), (MS1-DS1 is Z), (US2-MS2 is P), (MS2-DS2 is VP), (US3-MS3 is P) and (MS3-DS3 is P)) produce the fuzzy membership values 0.65, 1, 0.8, 0.7, 0.9 and 0.15, respectively. The fuzzy "AND" operator selects the minimum of the six values, 0.15, and the fuzzy operation for rule 2 is complete. The second rule to be evaluated is rule 3 where the six inputs ((US1-MS1 is P), (MS1-DS1 is Z), (US2-MS2 is P), (MS2-DS2 is VP), (US3-MS3 is P) and (MS3-DS3 is VP)) produce the fuzzy membership values 0.65, 1, 0.8, 0.7, 0.9 and 0.85, respectively. The fuzzy "AND" operator selects the minimum of the six values, 0.65, and the fuzzy operation for rule 3 is complete. The last rule to be evaluated in this example is rule 32 where the six inputs ((US1-MS1 is P), (MS1-DS1 is Z), (US2-MS2 is P), (MS2-DS2 is P), (US3-MS3 is P) and (MS3-DS3 is VP)) produce the fuzzy membership values 0.65, 1, 0.8, 0.3, 0.9 and 0.85, respectively. The fuzzy "AND" operator selects the minimum of the six values, 0.3, and the fuzzy operation for rule 32 is complete. It should be noted here that all the selected rules are evaluated in parallel. Now, the application of the fuzzy operator step is complete and the next step of the fuzzy process can start.

Table 18: Rules selected by the given inputs

Rule #	US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident Status
2	P	Z	P	VP	P	P	HIGH
3	P	Z	P	VP	P	VP	LOW
32	P	Z	P	P	P	VP	LOW

5.1.3 Step 3: Apply Implication Method

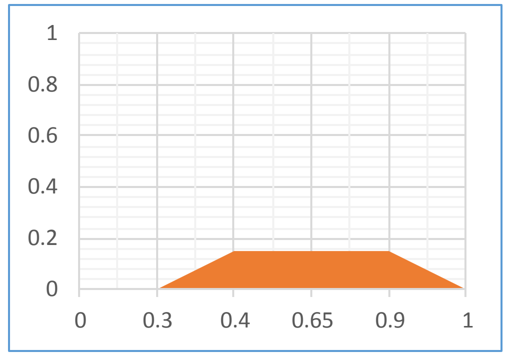
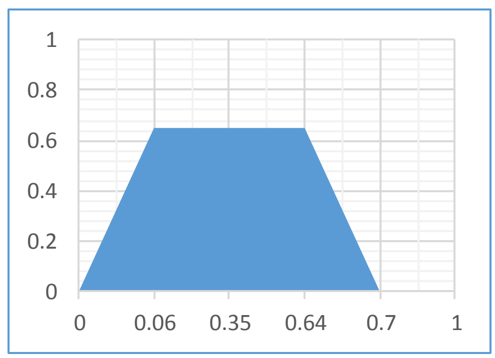
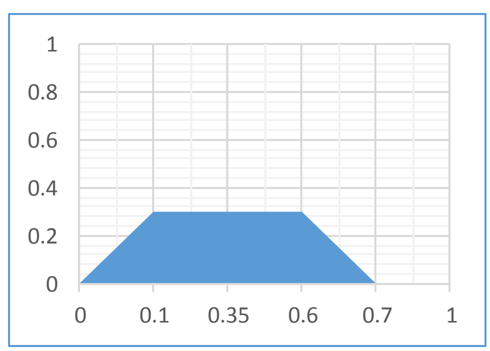
An implication method specifies how the fuzzy logic model scales the membership functions of the output, based on the weight of the corresponding rule. Applying the implication method requires a predetermination of the rules' weights. A rule weight is a number between 0 and 1. This weight is applied to the number resulted from the previous step (applying the fuzzy operator on the antecedents). A default weight of 1 is assigned to each rule that has no effect at all on the implication process. The rules weights can be modified, so they may have different influences on the output membership functions. However, in this model (FM1) and all the models built in this work (FM2 and FM3), all have the rules weight of 1.

5.1.4 Step 4: Outputs Aggregation

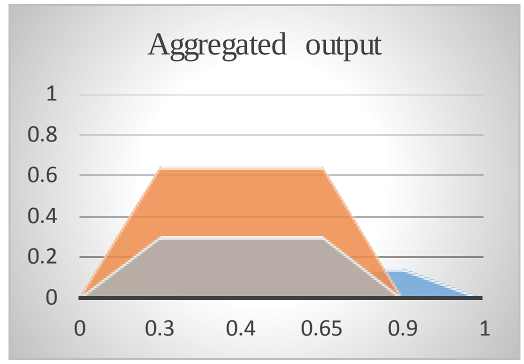
Aggregation is the process of merging the output membership functions that resulted from applying all the rules. All the consequents membership functions are combined into a single fuzzy set. This means aggregation process is done once for each output.

In Table 19 below, the three rules are placed together to show how the consequents membership functions are combined into a single fuzzy set that represents the output.

Table 19: Aggregation of the output

Rule #	US1-MS1	MS1-DS1	US2-MS2	MS2-DS2	US3-MS3	MS3-DS3	Incident_Status
2	μ_P = 0.65	μ_Z = 1	μ_P = 0.8	μ_{VP} = 0.7	μ_P = 0.9	μ_P = 0.15	μ_{HIGH} = 0.15 
3	μ_P = 0.65	μ_Z = 1	μ_P = 0.8	μ_{VP} = 0.7	μ_P = 0.9	μ_{VP} = 0.85	μ_{LOW} = 0.65 
32	μ_P = 0.65	μ_Z = 1	μ_P = 0.8	μ_P = 0.3	μ_P = 0.9	μ_{VP} = 0.85	μ_{LOW} = 0.3 

Apply aggregation method (max):



From Table 19, it can be seen that evaluating rule 2 results in "High" membership function with μ_{HIGH} = 0.15. Also, evaluating rule 3 results in "Low" membership function with μ_{LOW} = 0.65. The last evaluated rule was rule 32, results in "Low" membership function with μ_{LOW} = 0.3. The results of all three rules are then aggregated and the aggregated output is displayed in the last row of the same table. Figure 11 shows the aggregate output fuzzy set in solid blue which is the same result obtained in Table 19. Now, defuzzification process can start.

5.1.5 Step 5: Defuzzification

The fuzzy logic process ends with the defuzzification process. Defuzzification can be defined as converting the fuzzy output to crisp output. Although fuzziness is used to evaluate the rules, the fuzzy system produces a crisp number output, which is important for any further processing. The aggregate output fuzzy set is used as an input for the defuzzification process, which – in its own – produces a single number output. Although, several defuzzification methods exist, the centroid method (center of gravity method) is the defuzzification method that stands out and it is the method used in all the models built in this thesis. It pinpoints the exact vertical line that dissects the aggregate output set into two equal areas. The centroid defuzzification method finds the center of gravity of the output fuzzy set, where the center is represented by a single scalar value. By default, in MATLAB, the selected defuzzification method is the center of gravity method.

The centroid value in the given example is represented by a red line in MATLAB-Rule Viewer (Figure 11). This value represents the crisp output for the given inputs.

The centroid method calculation is performed by computing the centroid of the aggregate fuzzy output area. The strength of each output membership function is multiplied by the center of the same membership function. Subsequently, the results are summed. Finally, the area is divided by the sum of the output membership functions strengths, and a crisp output is resulted. The formula shown in Equation 9 is the fuzzy centroid formula (Engin, 2010).

$$Output = \frac{\sum_{i=1}^{n}(Center_i \bullet Strength_i)}{\sum_{i=1}^{n} Strength_i} \quad \text{.......... Equation 9}$$

Where n is the number of output membership functions involved. By using Equation 9, the result of the example's set of inputs is shown in Equation 10.

$$Output = \frac{((HIGH(Center) \times HIGH(Strength)) + ((LOW(Center) \times LOW(Strength))}{HIGH(Strength) + LOW(Strength)} \quad \text{...Equation 10}$$

$$Output = \frac{(0.65 \times 0.15) + (0.35 \times (0.65 + 0.3))}{(0.15) + (0.65 + 0.3)} = 0.391$$

Figure 12 shows the result of the given set of inputs using MATLAB-Rule Viewer. The rule viewer result is approximately the same as the one obtained from the calculations.

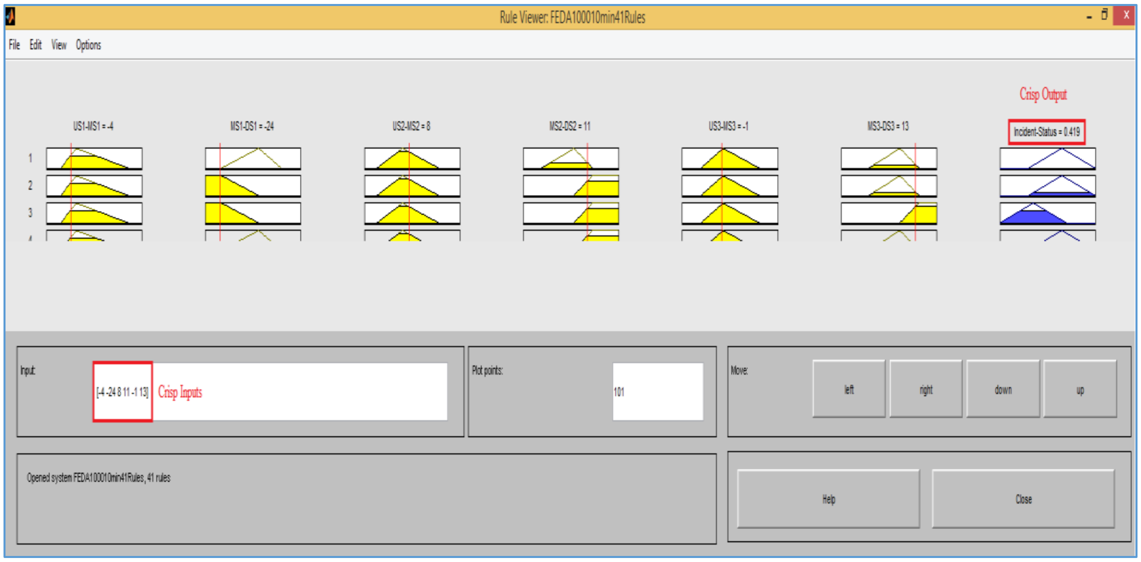


Figure 12: The result of the given set of inputs in the example using MATLAB-Rule Viewer

The error between the calculated result and the MATLAB produced result is given by Equation 11.

$$Error = 0.419 - 0.391 = 0.028 \text{.........} \text{ Equation 11}$$

5.2 Proposed Fuzzy Models Results and Validation

Three fuzzy models were developed in this work. Each model was validated with a new generated dataset. The models were evaluated using three main measures: good_detection rate, no_detection rate and false_alarm rate. In this case, time related issues were not considered in the validation process such as the lag between the actual incident and the detection. A resolution of 1 minute was used in the validation process. The difference between the actual and the detected incident_status of the same interval was used to validate the proposed logic. The efficiency of the developed logic was evaluated by measuring the capability of the logic in detecting the incident status correctly. It means that the proposed logic results in "High" term when there is an actual incident and results in "Low" term when there is actually no incident.

In order to validate the fuzzy models developed in this research, the following steps were followed:

1. Generate new datasets (Dataset 1v, Dataset 2v and Dataset 3v) using "PTV VISSIM".
2. Run each of the developed models (FM1, FM2 and FM3) on its validation dataset (Dataset 1v, Dataset 2v and Dataset 3v, respectively).
3. Round the crisp output (index) values to 0s and 1s:
   - If the resulted index is < 0.5, it should be rounded to "0" which represents no_incident status.
   - If the resulted index is ≥ 0.5, it should be rounded to "1" which represents an incident status.

4. Compare the detected incident status with the actual one for the same interval by calculating the difference: (Actual status – Detected status)
   - The difference of "0" represents good_detection intervals.
   - The difference of "1" represents no_detection intervals.
   - The difference of "-1" represents false_alarm intervals.
5. Find the percentages of good_detection intervals, no_detection intervals and false_alarm intervals compared to the total number of intervals of the particular simulation session.

Using MATLAB, Simulink model was built for the purpose of validation. Figure 13 shows the built Simulink model.

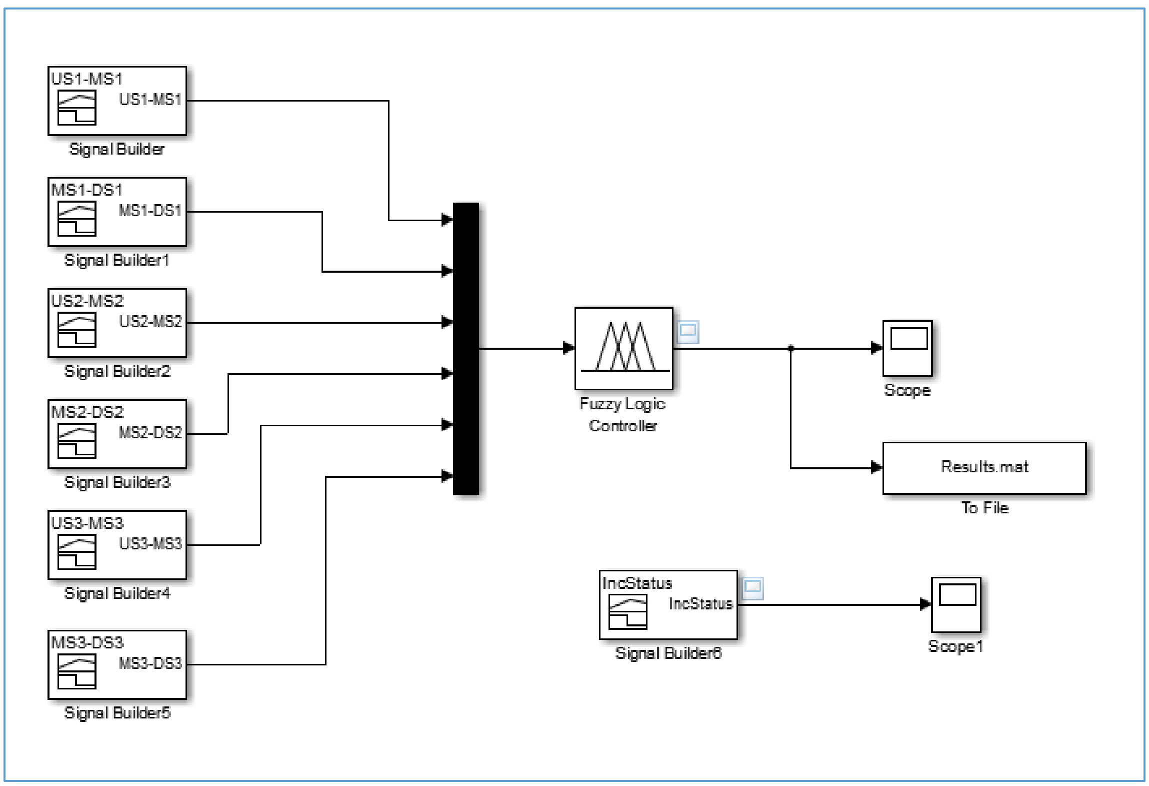


Figure 13: Simulink model

5.2.1 Fuzzy Model 1 (FM1) Results and Validation

5.2.1.1 Trial 1

FM1 was the first model developed in this work. It was built for a link length of 300 m, volume of 1000 vph and incidents duration of 10 minutes. To evaluate FM1, “PTV VISSIM” was used to generate new incident scenarios and extract the corresponding detector counts. Collected data was then used to build a new dataset (Dataset 1v) for the purpose of validation. The new dataset was also built for a link length of 300 m, a volume of 1000 vph and an incident duration of 10 minutes. “Dataset 1v” contained six inputs (“US1-MS1”, “MS1-DS1”, “US2-MS2”, “MS2-DS2”, “US3-MS3” and “MS3-DS3”) and one output representing the actual incident_status. The evaluation of the FM1 was based on the application of the model to the validation dataset (Dataset 1v), which comprised 16 incident scenarios. The evaluation gave an indication of the ability of the model in detecting incidents that the model had not previously seen. The comprised incident scenarios of (Dataset 1v) are displayed in Figure 14 below.

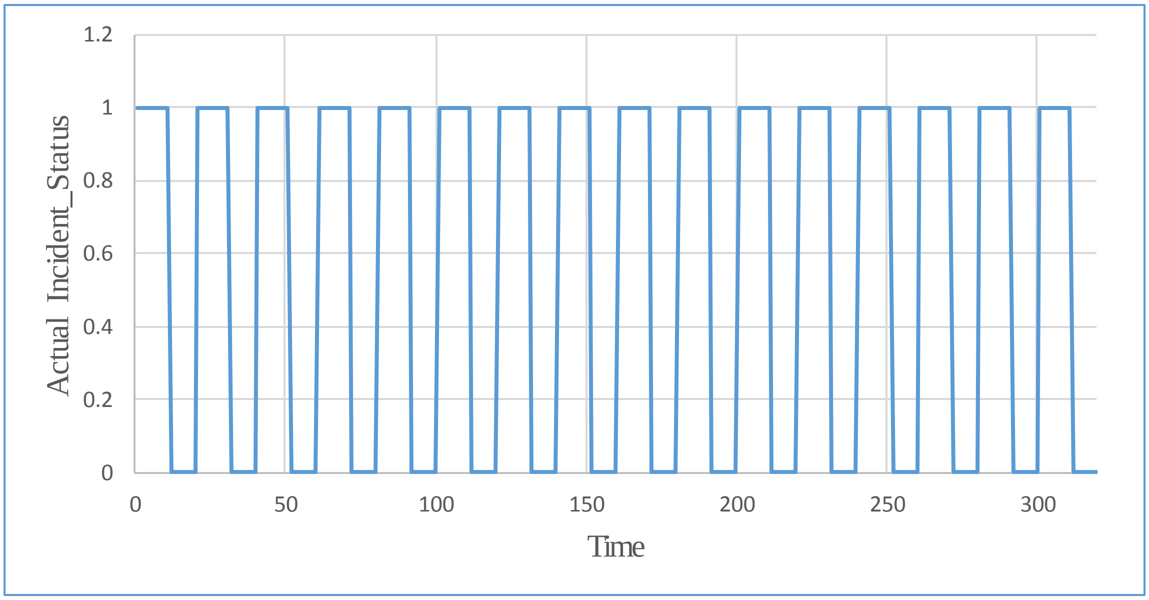


Figure 14: Actual incident_status for “Dataset 1v”

16 incidents with different locations and similar duration (10 minutes) are presented in Figure 14. The incidents were introduced over a period of 320 minutes. Figure 14 shows the actual incident_status. FM1 was then applied to the six inputs and the detected incident_status was collected and presented in Figure 15.

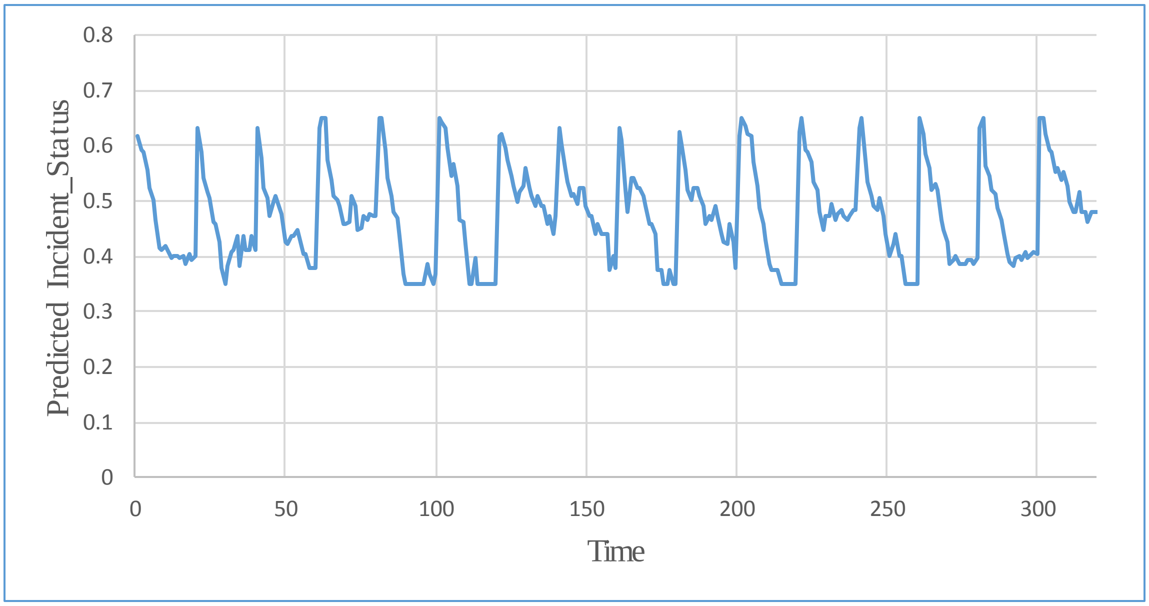


Figure 15: FM1 detected incident_status for “Dataset 1v”

The values resulted using the model were within [0, 1] range. As the index value increases, the probability of incident increases as well. Actual output values less than 0.5 were rounded to “0” which was an indication that no-incident has been detected, and values greater than or equal to 0.5 were rounded to “1” which signified that an incident has been detected. Figure 16 illustrates the detected incident_status crisp values before and after rounding. Figure 17 illustrates the rounded crisp values.

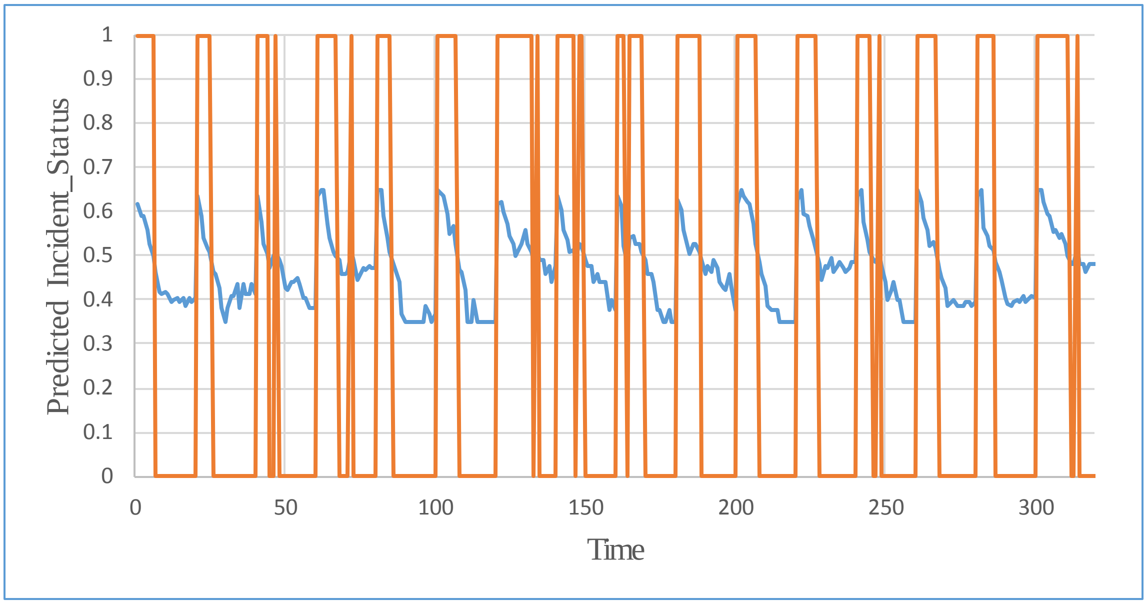


Figure 16: Detected incident_status before and after rounding for "Dataset 1v"

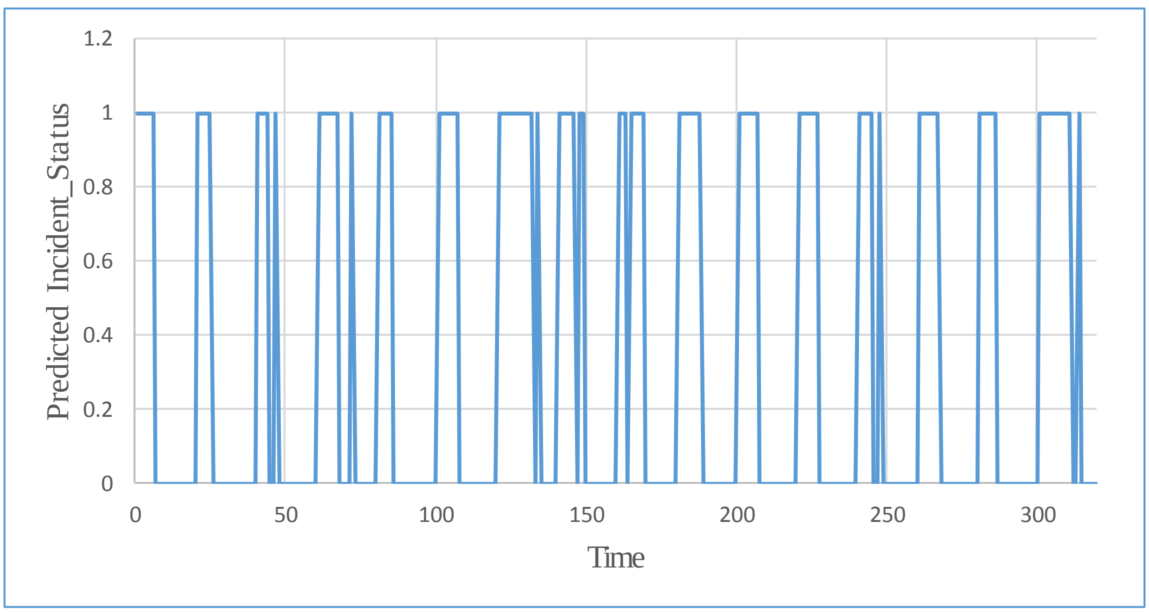


Figure 17: Predicted incident_status after rounding for "Dataset 1v"

The next step was comparing the actual values with the detected values of incident_status and then calculating the percentage of time where the two values matched.

The difference between the actual and the rounded values was calculated using Equation 12 and the results were displayed in Figure 18.

$$Error = (Actual\_Value) - (Predicted\_Value) \quad \ldots\ldots\ldots \text{ Equation 12}$$

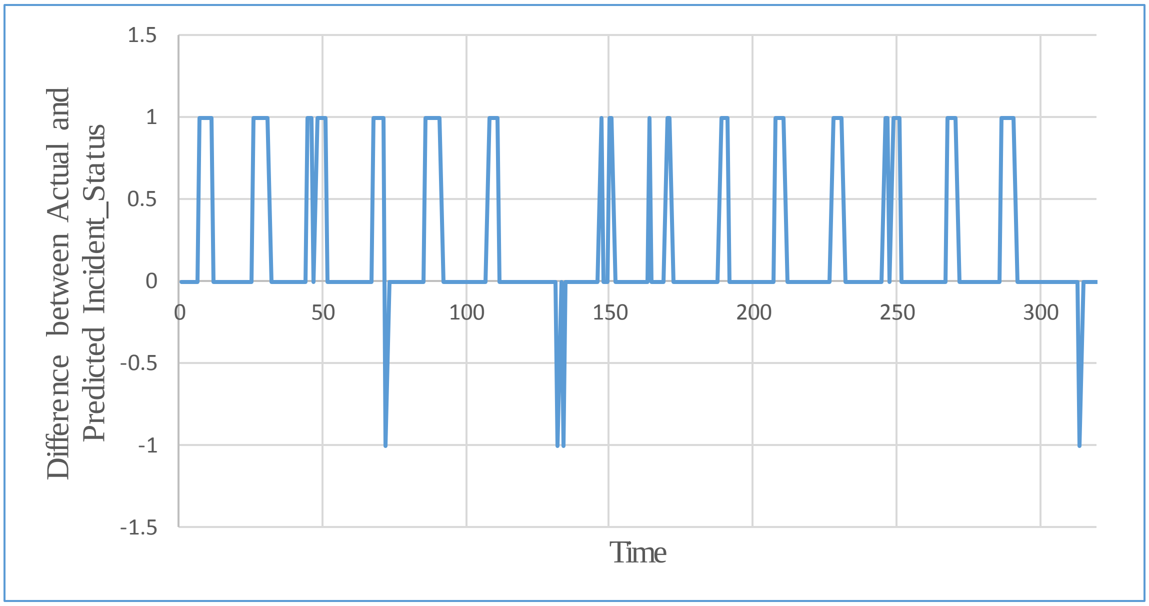


Figure 18: Difference between actual and detected incident_status for "Dataset 1v"

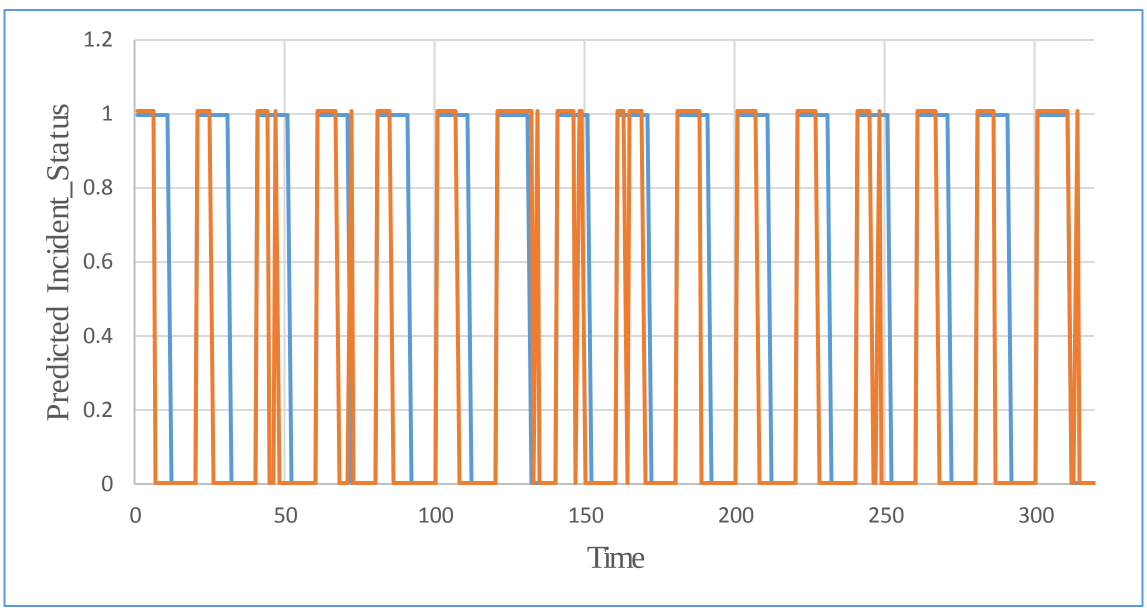


Figure 19: Actual and rounded detected incident_status for "Dataset 1v"

The application of Equation 12 resulted in values that varied between “1”, “-1” and “0”. The upper part of Figure 18 presents the undetected incident intervals where the result of Equation 12 was “1” which means that the fuzzy logic was not able to detect an existing incident. The lower part of the same figure shows the false detection intervals where the result of Equation 12 was “-1” which indicates that the fuzzy logic was detecting an incident that did not exist (false). The zero difference between actual and detected incident_status reflects accurate detection intervals of the proposed fuzzy logic. It means that FM1 was detecting incidents where they really existed.

The simulation was conducted over a period of 320 minutes and the proposed fuzzy logic model (FM1) was set to update its status every minute. Therefore, it can be said that FM1 has been tested 320 times. It is important to note here that the purpose of this system is to detect and return the incident status every minute, which means it is not to detect the presence of the incident as a whole. Over the period of 320 minutes, there were 176 incident intervals and 144 no-incident intervals. 254 intervals were detected properly as incident or no-incident intervals. 66 intervals returned inaccurate incident_status. Of those 66 intervals, there were 62 instances where the proposed system (FM1) failed to report an existing incident. In the remaining 4 intervals, the system falsely reported the occurrence of an incident. The verification measures were calculated using Equation 13, Equation 14 and Equation 15. It resulted in about 79% good_detection intervals, 20% no_detection intervals and 1% false_alarms. All calculations are presented below. The results indicate a good incident detection ability of the proposed fuzzy logic (FM1).

$$\% \ Good\_Detection = \frac{Count \ of \ all \ the \ "0" \ difference \ intervals}{Total \ count \ of \ simulation \ intervals} \times 100$$ …. Equation 13

$$\% \ Good\_Detection = \frac{254}{320} \times 100 \approx 79$$

$$\% \ No\_Detection = \frac{Count \ of \ all \ the \ "1" \ difference \ intervals}{Total \ count \ of \ simulation \ intervals} \times 100$$ ………. Equation 14

$$\% \ No\_Detection = \frac{66}{320} \times 100 \approx 20$$

$$\% \ False\_Alarms = \frac{Count \ of \ all \ the \ "-1" \ difference \ intervals}{Total \ count \ of \ simulation \ intervals} \times 100$$ ……. Equation 15

$$\% \ False\_Alarms = \frac{4}{320} \times 100 \approx 1$$

5.2.1.2 Trial 2

This trial was conducted to study the effect of changing the membership functions of the output. New limits were defined for the output membership functions as shown in (Figure 20).

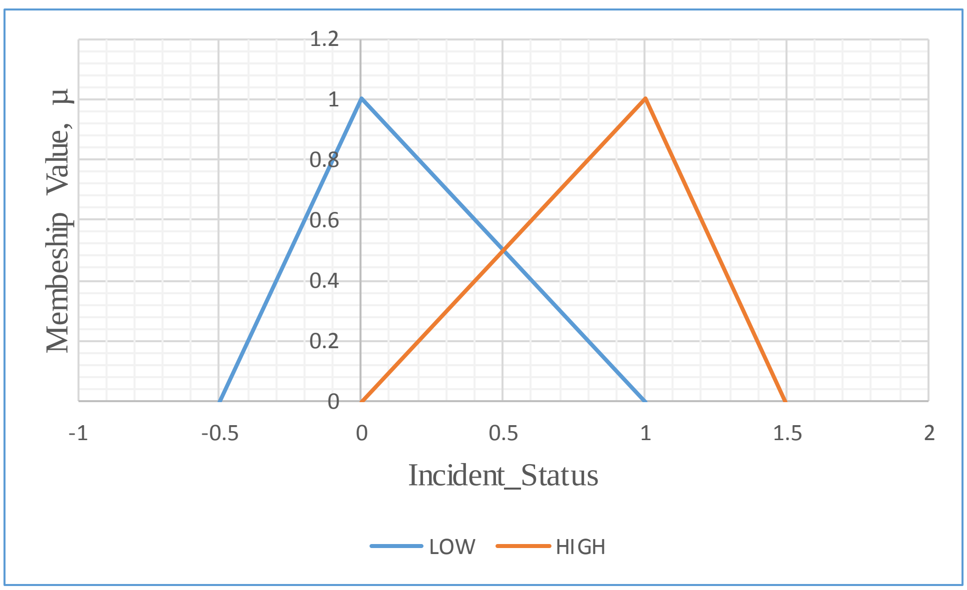


Figure 20: New output membership functions

The results of applying the modified FM1 on (Dataset 1v) are displayed in (Figure 21).

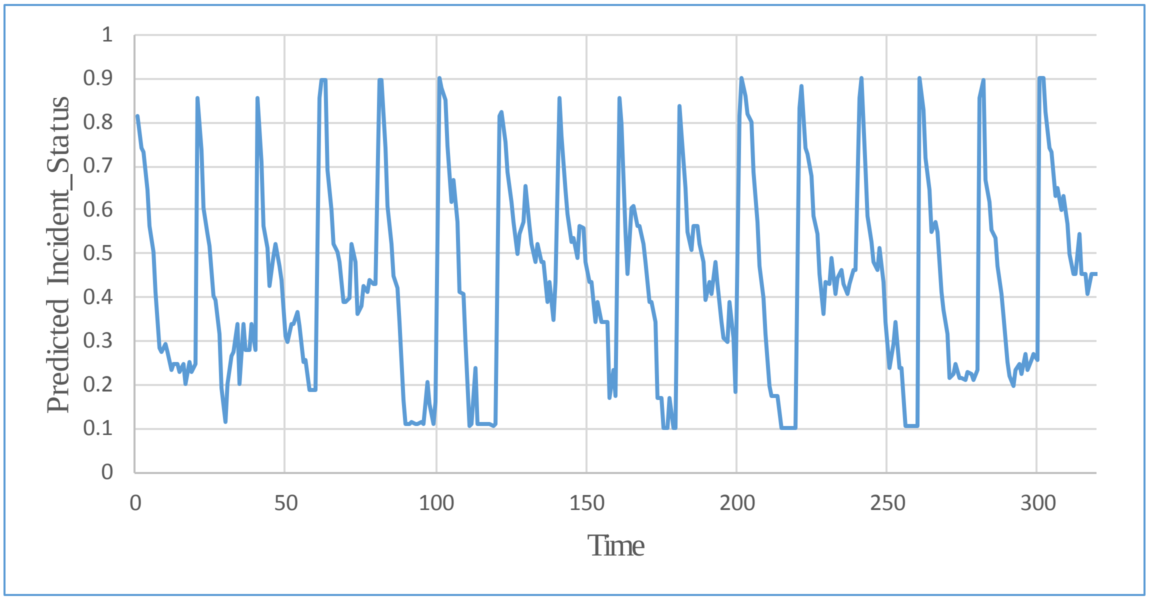


Figure 21: Modified FM1 detected incident_status for "Dataset 1v"

The verification measures were calculated and listed below:

% Good_Detection = 79

% No_Detection = 20

% False_Alarms = 1

By comparing the results of the previous trials, it was noticed that the output in the second trial was in the range [0.1, 0.9] while it was in the range [0.3, 0.7] in the first trial. Although changing the limits of the output membership functions has affected the output values, it did not affect the overall performance of the model in the studied case.

5.2.2 Fuzzy Model 2 (FM2) Results and Validation

5.2.2.1 Trial 1

FM2 was the second model developed in this work. It was built for a link length of 300 m, a volume of 1000 vph and different incident duration. Similar to FM1, to evaluate FM2, "PTV VISSIM" was used to generate new incident scenarios and extract the corresponding detectors counts. Collected data was then used to build a new dataset (Dataset 2v) for the purpose of validation. The new dataset was also built for a link length of 300 m, a volume of 1000 vph and different incident duration that varied from 5 to 15 minutes. Similar to "Dataset 1v", "Dataset 2v" contained six inputs ("US1-MS1", "MS1-DS1", "US2-MS2", "MS2-DS2", "US3-MS3" and "MS3-DS3") and one output that was the actual incident_status. The evaluation of the FM2 was based on the application of the model to the validation dataset (Dataset 2v), which comprised 60 incident scenarios. This evaluation gave an indication of the ability of the model in detecting incidents that the model had not previously seen. The comprised incident scenarios of (Dataset 2v) are displayed in Figure 22.

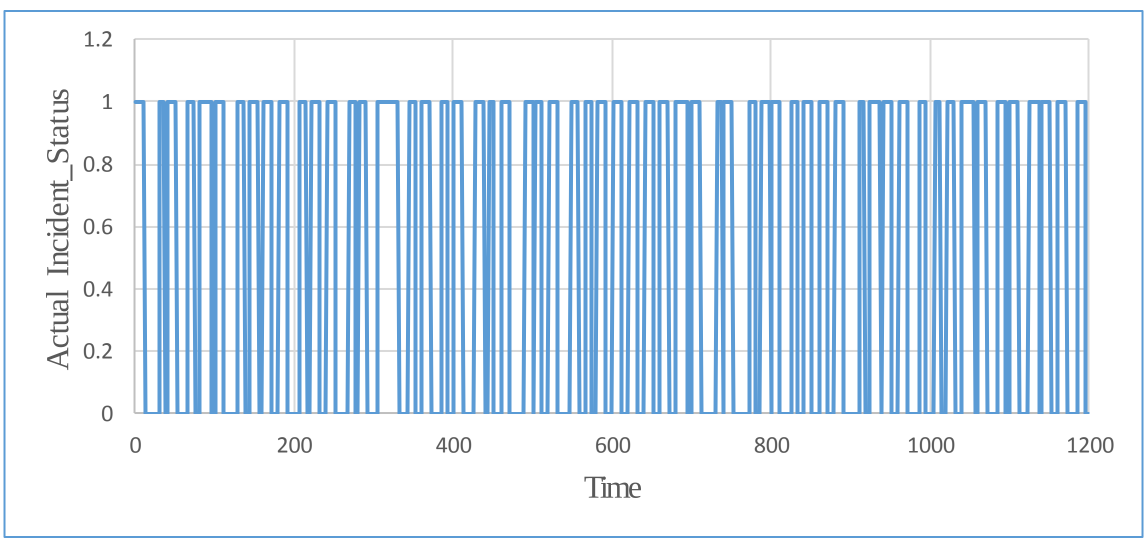


Figure 22: Actual incident_status for "Dataset 2v"

60 incidents with different locations and different duration (5 to 15 minutes) are presented in Figure 22. They are introduced over a period of 1200 minutes. Figure 22 shows the actual incident_status. FM2 was applied to the six inputs. The detected incident_status is collected and presented in Figure 23.

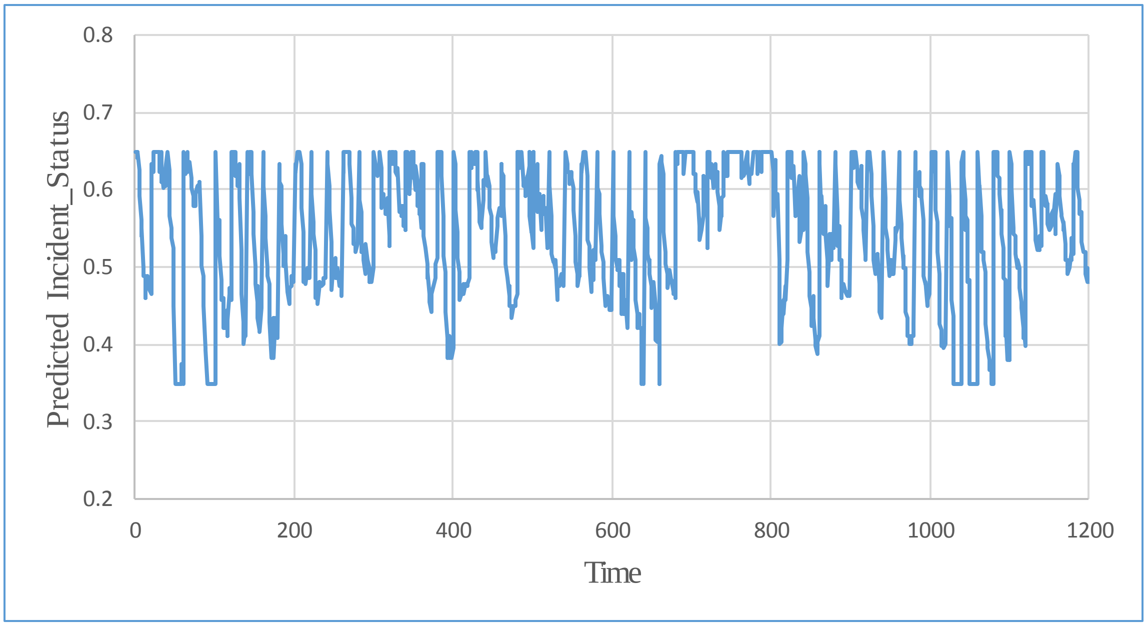


Figure 23: FM2 detected incident_status for "Dataset 2v"

The values resulted using the model were within [0,1] range. Output values were rounded to zeroes and ones similar to what was done in the previous section. Figure 24 illustrates the detected incident_status crisp values before and after rounding. Figure 25 illustrates the rounded crisp values.

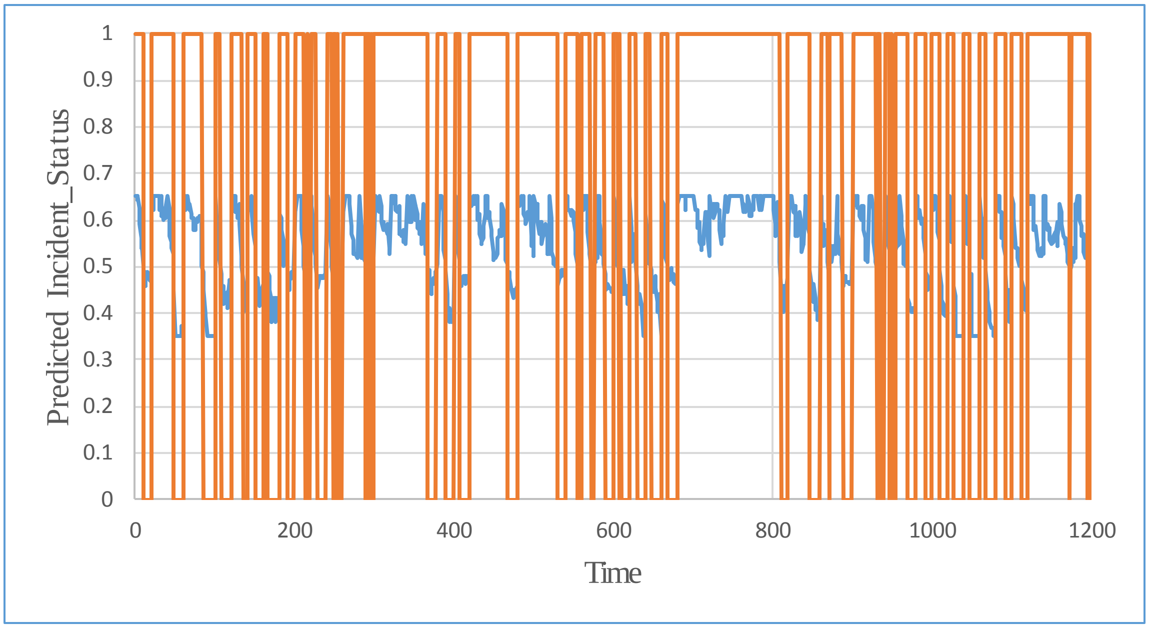


Figure 24: Detected incident_status before and after rounding for "Dataset 2v"

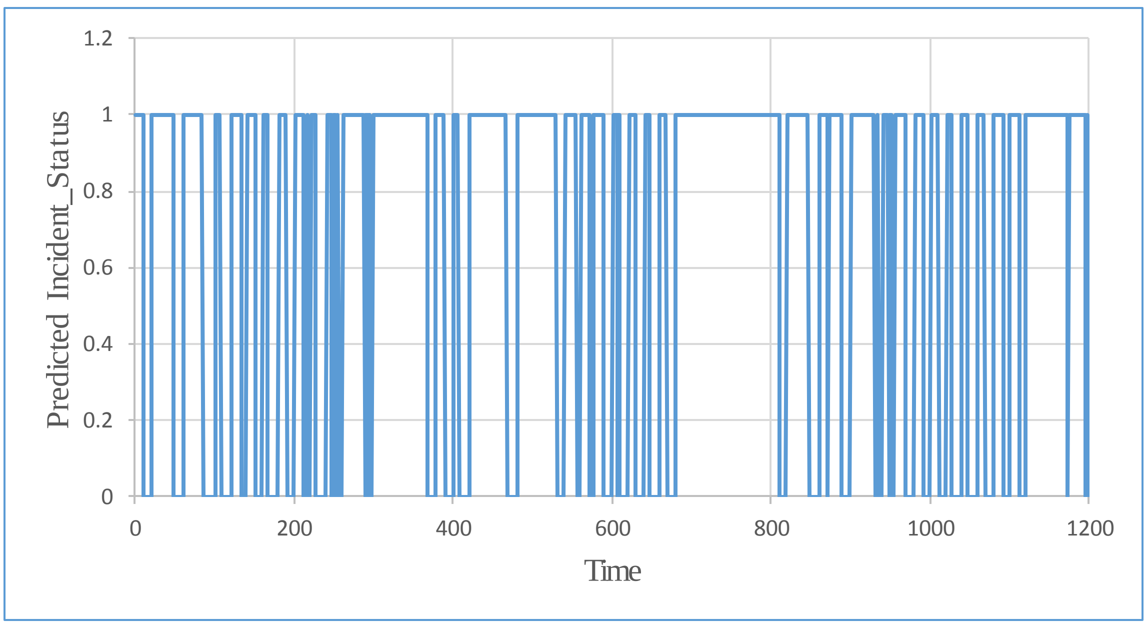


Figure 25: Detected incident_status after rounding for "Dataset 2v"

The next step was comparing the actual values with the detected values of incident_status and calculating the percentage of time where the two values matched. The difference between the actual and the rounded values was calculated using Equation 12. The results are displayed in Figure 26.

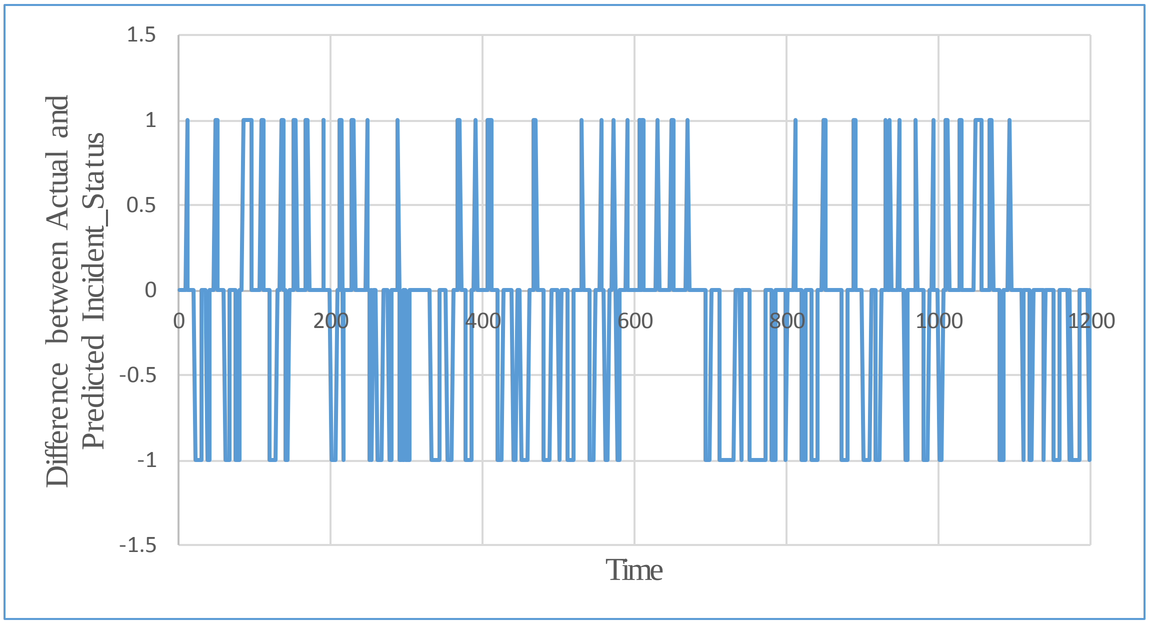


Figure 26: Difference between actual and detected incident_status for "Dataset 2v"

Similar to FM1, the application of Equation 12 resulted in values that varied between "1", "-1" and "0" where "1" represented no_detection intervals, "-1" represented false_alarm intervals and "0" represented good_detection intervals.

This simulation was conducted over a period of 1200 minutes and the proposed fuzzy logic system (FM2) was updating its status every minute. Therefore, it has been tested 1200 times. Again, the purpose of this system is to detect and return the incident status every minute not to detect the presence of the incident. Over the period of 1200 minutes, there were 628 incident intervals and 572 no-incident intervals. 780 intervals were detected properly as incident or no-incident intervals. 420 intervals returned inaccurate incident status. From the 420 intervals, there were 117 instances where the proposed system (FM2) failed to report an existing incident. In the remaining 303 intervals, the system falsely reported the occurrence of an incident. The verification resulted in about 65% good_detection intervals, 10% no_detection intervals and 25% false_alarms. Calculations were performed using Equation 13, Equation 14 and Equation 15 respectively. The calculations are displayed below.

$$\% \; Good\_Detection = \frac{780}{1200} \times 100 \approx 65$$

$$\% \; No\_Detection = \frac{117}{1200} \times 100 \approx 10$$

$$\% \; False\_Alarms = \frac{303}{1200} \times 100 \approx 25$$

5.2.2.2 Trial 2

In this trial, the developed model (FM2) was tested on a different dataset (Dataset 2v-a). The new dataset was extracted from the previous one (Dataset 2v) excluding the incidents scenarios that lasted less than 10 minutes. The extracted dataset comprised 41 incident scenarios. This trial was executed to measure the performance of the proposed logic (FM2) in detecting incidents that were greater than or equal to 10 minutes in terms of duration. The comprised incident scenarios of (Dataset 2v-a) are displayed in Figure 27 below.

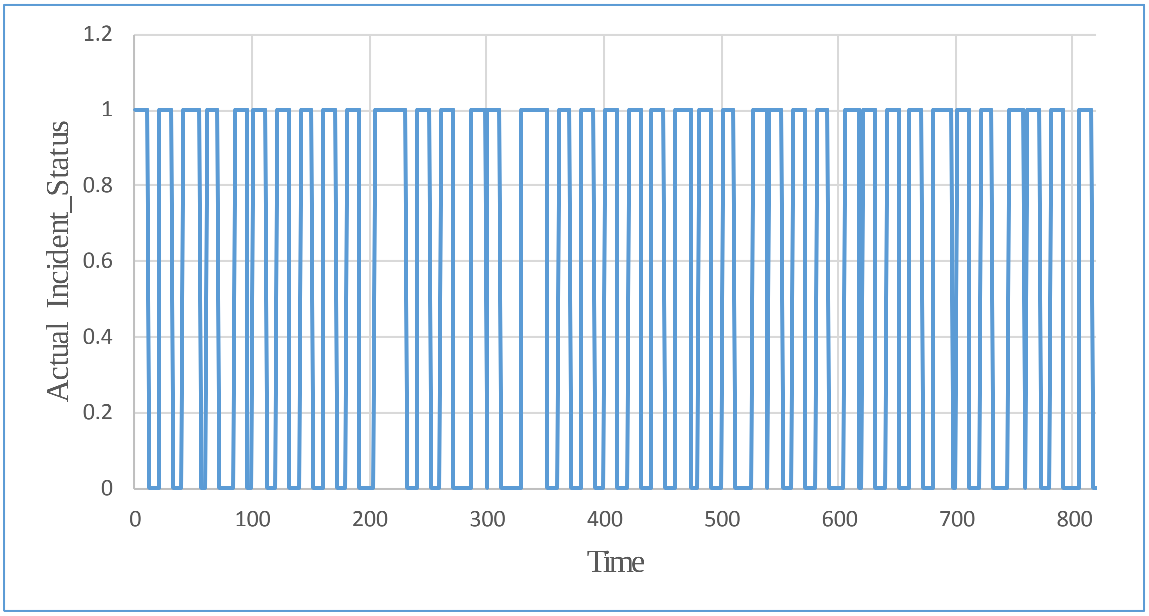


Figure 27: Actual incident_status for "Dataset 2v-a"

41 incidents with different locations and different durations (≥ 10 minutes) are presented in Figure 27. They were introduced over a period of 820 minutes. Figure 27 shows the actual incident status. FM2 was applied to the six inputs. The detected incident_status is collected and presented in Figure 28.

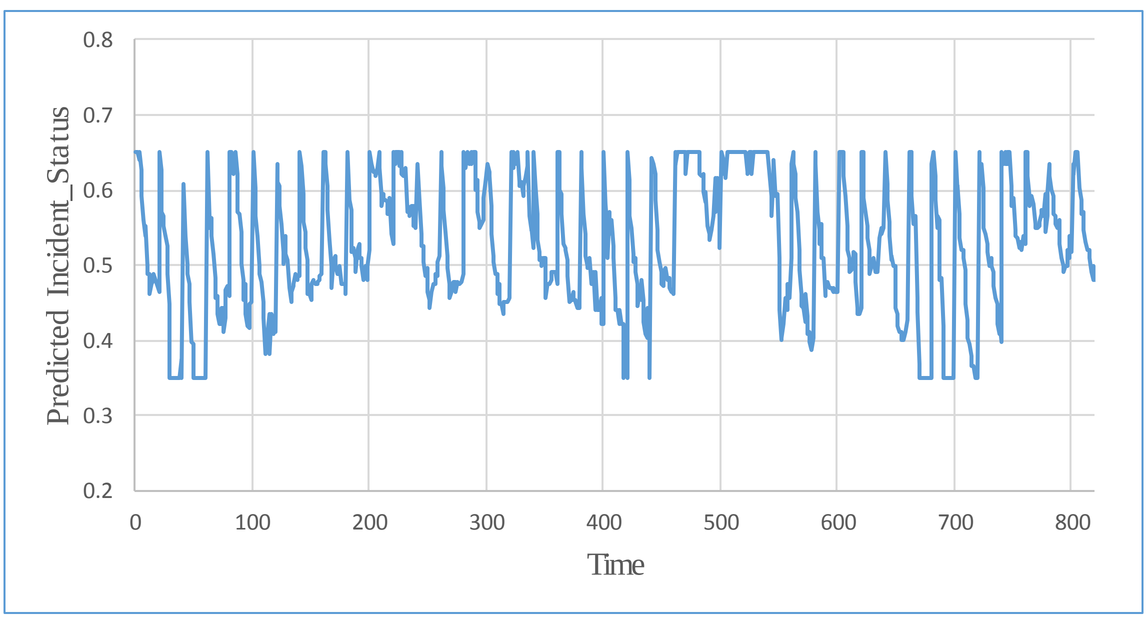


Figure 28: FM2 detected incident_status for "Dataset 2v-a"

The values resulted using the model were within [0,1] range. Output values were rounded to 0s and 1s, similar to what was done earlier. Figure 29 illustrates the detected incident_status crisp values before and after rounding. Figure 30 illustrates the rounded crisp values.

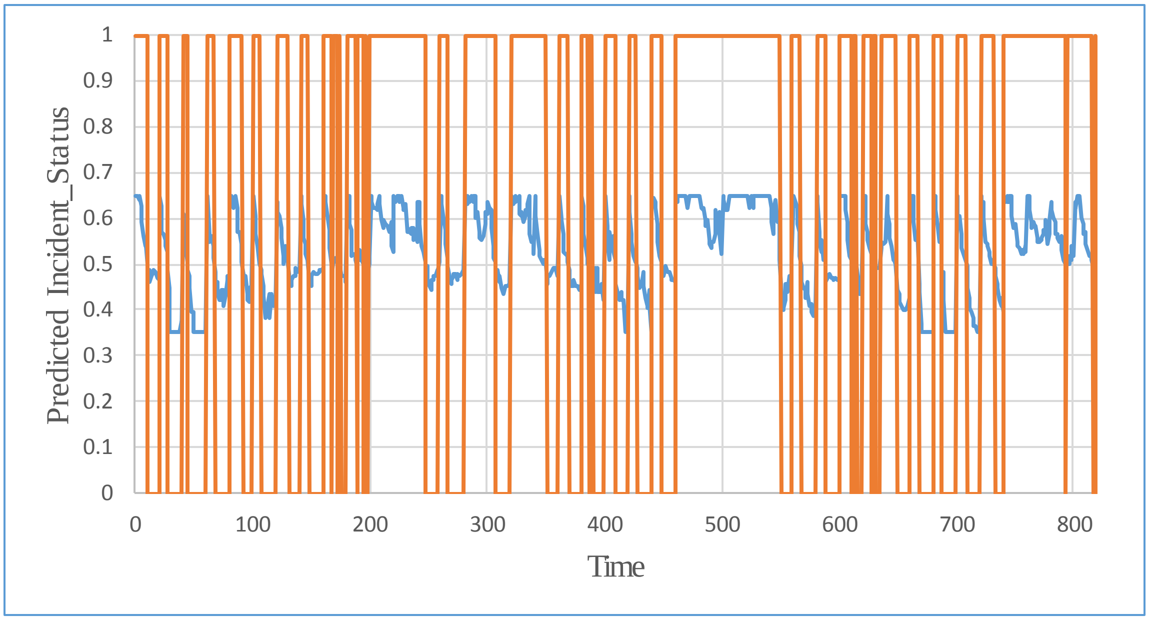


Figure 29: Detected incident_status before and after rounding for “Dataset 2v-a”

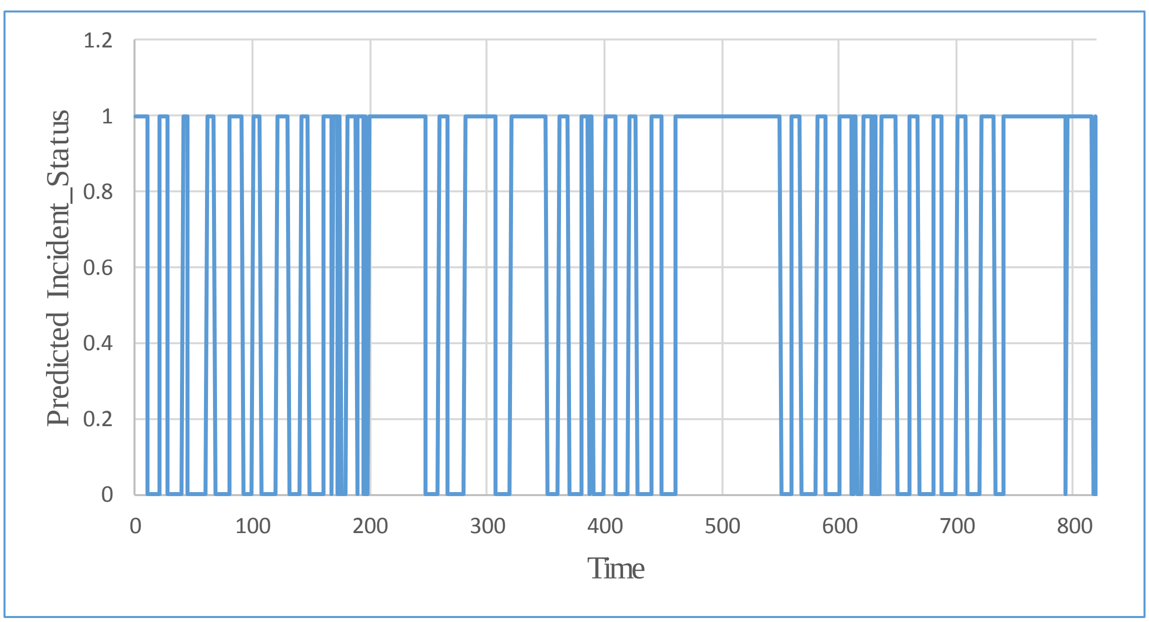


Figure 30: Deteted incident_status after rounding for “Dataset 2v-a”

The next step was comparing the actual values with the detected values of incident-status and then calculating the percentage of time where the two values matched. The difference between the actual and the rounded values was calculated using Equation 12. The results are displayed in Figure 31.

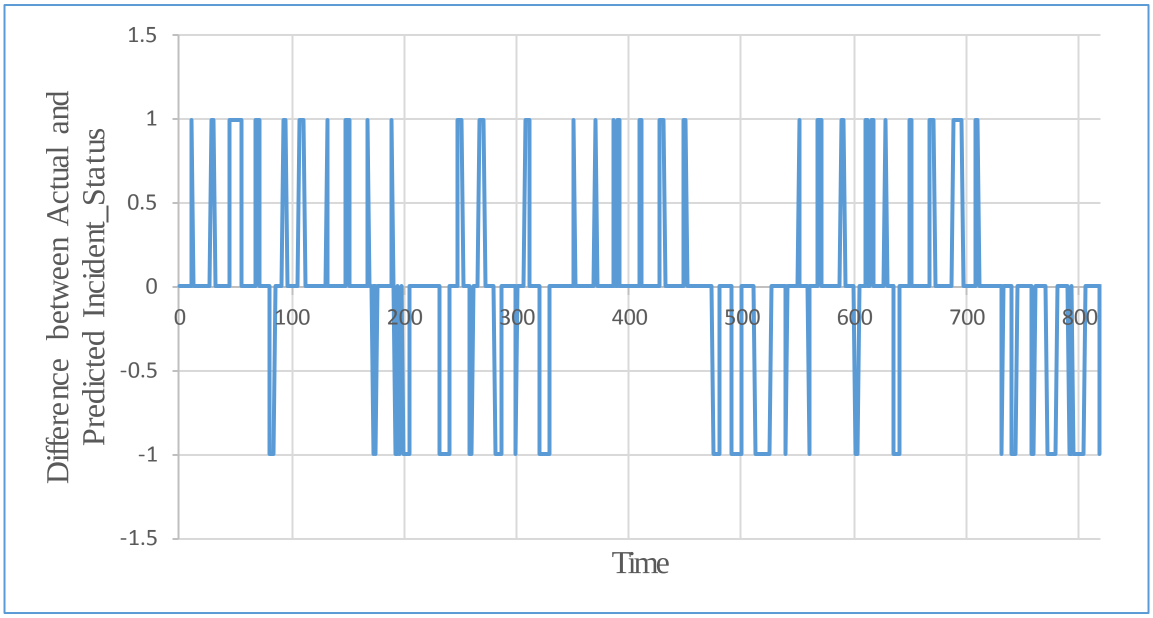


Figure 31: Difference between actual and detected incident_status for "Dataset 2v-a"

The simulation was conducted over a period of 820 minutes and the proposed fuzzy logic system (FM2) was updating its status every minute. Therefore, it has been tested 820 times. Over the period of 820 minutes, there were 477 incident intervals and 343 no-incident intervals. 610 intervals were detected properly as incident or no-incident intervals. 210 intervals returned inaccurate incident status. From the 210 intervals, there were 96 instances where the proposed system failed to report an existing incident. In the remaining 114 intervals, the system falsely reported the occurrence of an incident. The verification resulted in about 74% good_detection intervals, 12% no_detection intervals and 14% false_alarms. All calculations were performed using Equation 13, Equation 14 and Equation 15 respectively. The calculations are presented below.

$$\%\ Good\_Detection = \frac{610}{820} \times 100 \approx 74$$

$$\%\ No\_Detection = \frac{96}{860} \times 100 \approx 12$$

$$\%\ False\_Alarms = \frac{114}{820} \times 100 \approx 14$$

From the results obtained in trial 2, an improvement of around 9% in detection was noticed. The no_detection rates were near in both trials (10% in the first trial and 12% in the second trial). Also, the false_alarm rate was reduced by 11%. These improvements show that FM2 works better in detecting longer duration incidents, which is meaningful, because longer duration incidents have clearer incident patterns than shorter ones, and thus make incident detection easier.

5.2.3 Fuzzy Model 3 (FM3) Results and Validation

5.2.3.1 Trial 1

FM3 was the third model developed in this work. It was built for a link length of 300 m, different volumes (500, 1000 and 1300 vph) and different incident duration. To evaluate FM3, "PTV VISSIM" was used to generate new incident scenarios and extract the corresponding detector counts. Collected data was then used to build a new dataset (Dataset 3v) for the purpose of validation. The new dataset was also built for a link length of 300 m, different volumes (500, 1000 and 1300 vph) and different incident duration.

Similar to "Dataset 1v" and "Dataset 2v", "Dataset 3v" contained six inputs ("US1-MS1", "MS1-DS1", "US2-MS2", "MS2-DS2", "US3-MS3" and "MS3-DS3") and one output representing the actual incident_status. The evaluation of the FM3 was based on the application of the model to the validation dataset (Dataset 3v), which comprised 181 incident scenarios.

The simulation was conducted over a period of 3620 minutes and the proposed fuzzy logic system (FM3) was updating its status every minute. The results are summarized in Table 20 and the related figures are displayed in appendix A.

Table 20: Evaluation of FM3 using "Dataset 3v"

Simulation Period (minutes)	3620
# Actual Incident Intervals	1903
# Actual No-Incident Intervals	1717
% Good_Detection	54%
% No_Detection	9%
% False_Alams	37%

5.2.3.2 Trial 2

In this trial, the developed model (FM3) was tested on a different dataset (Dataset 3v-a). The new dataset was extracted from the previous one (Dataset 3v) excluding the incidents scenarios that last less than 10 minutes. The extracted dataset comprised 108 incident scenarios. This trial was executed to measure the performance of the proposed logic (FM3) in detecting incidents that were greater than or equal to 10 minutes in terms of duration.

The simulation was conducted over a period of 2160 minutes and the proposed fuzzy logic system (FM3) was updating its status every minute. The results are summarized in Table 21 and the related figures are displayed in appendix A.

Table 21: Evaluation of FM3 using “Dataset 3v-a”

Simulation Period (minutes)	2160
# Actual Incident Intervals	1300
# Actual No-Incident Intervals	860
% Good_Detection	63%
% No_Detection	10%
% False_Alams	27%

From the results obtained in trial 2, an improvement of around 9% in detection was noticed. The no_detection rates were close in both trials (9% in the first trial and 10% in the second trial). Also, the false_alarm rate was reduced by 10%. Similar to trial 2 for “FM2”, these improvements show that FM3 works better in detecting longer duration incidents.

5.2.4 Fuzzy Models Comparison

Five experiments were conducted in order to evaluate and validate the three fuzzy models (FM1, FM2 and FM3). The results obtained from these experiments are displayed in Table 22 and the properties of the validation datasets used are summarized in Table 23. It can be observed from Table 22 that the best performance was obtained using FM1 that was built for a single volume link and fixed incidents duration of 10 minutes. It resulted in about 79% good_detection, 20% no_detection and 1% false_alarms. In FM2, the performance was affected by the variety of incident durations used. It resulted in about 65% good_detection, 10% no_detection and 25% false_alarms. A second trial was conducted for FM2 but with incident duration greater than or equal to 10 minutes. The trial resulted in 74% good_detection, 12% no_detection and 14% false_alarms. The results show an improvement in both good_detection and false_alarm rates. The detection rate was improved by 9% and the false_alarms were decreased by 11%, which is considered a good improvement. FM3 was developed for different link volumes (500, 1000 and 1300 vph) and varied incident duration. It resulted in 54% good_detection, 9% no_detection and 37% false_alarms. This was the poorest performance among all. In order to improve it, a second trial was conducted using incidents that have duration greater than or equal to 10 minutes. The new trial resulted in 63% good_detection, 10% no_detection and 27% false_alarms. Similar to the second trial of FM2, an improvement in the FM3 performance was noticed. The detection rate was increased by 9% and the false_alarms were reduced by 10%. Hence, it can be said that the devised fuzzy logic is highly influenced by two main factors, namely, the link volume and the incident duration. It works better for single volume links (FM1 and FM2 have better results than FM3). It also performs better in detecting incidents with duration not less than 10 minutes (trial 2 in both FM2 and FM3 have better results than trial 1). The link volume has a great effect on the range selection of the membership functions during the development of the fuzzy logic. The incident duration has a noticeable effect on incident patterns. In conclusion, the performance of the developed models are highly affected by the link volume and the incident duration which makes it essential to consider both of them when developing fuzzy models for incident detection using the proposed methodology.

Table 22: Summary of the results

Fuzzy Model	Dataset for Calibration	Dataset for Validation	Simulation Period (minutes)	# Actual Incident Intervals	# Actual No-Incident Intervals	% Good_Detection	% No_Detection	% False Alarms
FM1	Dataset 1c	Dataset 1v	320	176	144	79%	20%	1%
FM2	Dataset 2c	Dataset 2v	1200	628	572	65%	10%	25%
		Dataset 2v-a	820	477	343	74%	12%	14%
FM3	Dataset 3c	Dataset 3v	3620	1903	1717	54%	9%	37%
		Dataset 3v-a	2160	1300	860	63%	10%	27%

Table 23: Properties of validation datasets

Dataset for Validation	# Scenarios	# Entries	Volume (vph)	Incidents Duration (minutes)
Dataset 1v	16	320	1000	10
Dataset 2v	60	1200	1000	Varied
Dataset 2v-a	41	820	1000	>= 10
Dataset 3v	181	3620	500-1000-1300	Varied
Dataset 3v-a	108	2160	500-1000-1300	>= 10

5.2.5 Combined Fuzzy Models

Since the volume is an effective factor in incident detection using the suggested fuzzy models, and since a single intersection may have different volumes throughout the day, it would be a good idea to include the volume in the process of incident detection. One of the trials conducted in this research to improve the performance of the proposed models was considering the volume as an input. In the trial, the volume had its own set of membership functions (low, medium and high) and it was involved in the rules of the fuzzy inference system. Unfortunately, the trial resulted in no significant improvement over the previously obtained results. A better suggestion was to use the volume to select the best fuzzy model to be implemented. The structure of the suggested model is displayed in Figure 32. The figure shows an additional step prior to the implementation of the fuzzy model. This step is to classify the volume of the link into, as an example, three. It uses the current volume as an input and classifies it based on few simple if-then conditions. The output of this step is a volume class (low, medium or high). Such classification can assist in choosing the best fuzzy model to be implemented so that the six inputs will enter the selected model and a decision on incident_status will be made based on the rules of that model.

In order to study the effectiveness of this thought, the suggested model was built and tested. For developing this system, it was required to build a fuzzy model for each class of volume. So, three models were built, one for detecting incidents in case of low volume (<= 500 vph), a second model for medium volume (>500 vph and <1300 vph) and the last model for high volume (>= 1300 vph). In other words, several fuzzy models were built for different volumes and the best one was selected based on the required volume.

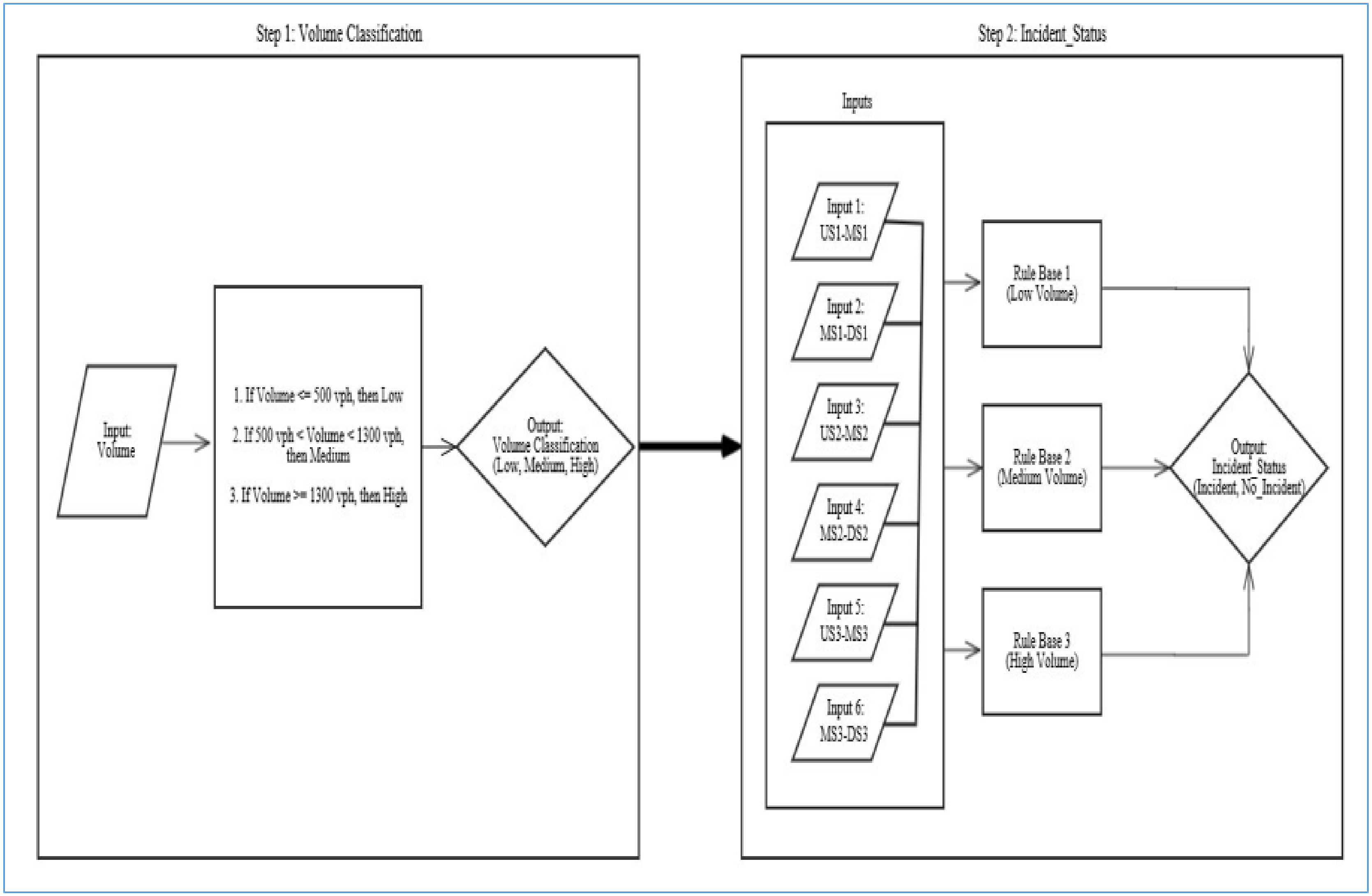


Figure 32: Combined fuzzy models

The previous model was built using three simulated datasets, one for each model. Another set was generated to test it. The dataset used for testing contained different incident scenarios under different volumes (500, 1000 and 1300 vph). All incident scenarios had a duration of 10 minute or more. The results obtained from applying the last suggested model are displayed in Table 24.

Table 24: Evaluation of the combined models

Simulation Period (minutes)	2400
# Actual Incident Intervals	1420
# Actual No-Incident Intervals	980
% Good_Detection	65%
% No_Detection	16%
% False_Alams	19%

Table 25: Comparison between the combined models and FM3

	Combined Models	FM3
% Good_Detection	65%	63%
% No_Detection	16%	10%
% False_Alams	19%	27%
Dataset used for testing	Volume: 500, 1000 and 1300 vph Incidents Duration: >= 10 min.	Volume: 500, 1000 and 1300 vph Incidents Duration: >= 10 min.

From the results displayed in Table 24 and Table 25, the combined models displayed a very small improvement in terms of good_detection rate when compared to FM3 but it had a good improvement in terms of false_alarms. In numbers, the good_detection was increased by 2% while the false_alarms were decreased by 8% (See Table 25). The reason of comparing the combined models with FM3 was that they were both designed to detect incidents under different volumes.

From the previous discussed results, it can be concluded that using separate volume based models performs better than using a single set of rules to detect incidents under different volumes.

5.2.6 Comparison with Other Work

There are two main difficulties with comparing this work to other similar work. The first one was mentioned in the verification issues in chapter 2 where there is lack of a standard methodology to obtain the verification measures. This makes it difficult to compare the evaluation results using different algorithms. The second reason is that most of the incident detection methods available in the literature are detect the incident availability as a whole, while the proposed models in this work are used to detect and return the incident status every minute.

In (Hawas, 2007), the developed system was able to detect the incident status every 10 seconds and the verification measures were calculated in a way similar to this work. As a comparison, the detection rate in (Hawas, 2007) was better than the detection rate achieved in this research; however, FM1 developed in this research had a smaller percentage of false alarms. Fuzzy logic theory was implemented in this research because of its relative simplicity when compared to neuro-fuzzy used in (Hawas, 2007). The number of rules generated here did not exceed 51 rules while it reached 2625 rules in (Hawas, 2007). The simplicity of the system translates into easier rules track quick reconfiguration of the rules when needed.

Chapter 6: Summary, Conclusion and Recommendations

This chapter presents a summary of the work completed in this thesis. It briefly presents the results obtained and concludes with suggestions for future work that can be pursued for further improvement.

6.1 Summary

Traffic incidents play a vital role in non-recurrent congestions that cause traffic delays in urban areas. Incident detection is the first step toward solving this issue. Detecting traffic incidents in urban streets will help to clear such incidents and accordingly can lead to increased safety for travelers and decreased overall traffic delays. In order to effectively achieve that, a methodology to develop fuzzy models for incident detection was presented in this thesis.

In this thesis, traffic incident management system, incident detection and fuzzy logic were discussed. The structure of the suggested fuzzy models was presented first, and later, the fuzzy models were built, tested and evaluated. Finally, the results were displayed, evaluated and discussed.

Three fuzzy models (FM1, FM2 and FM3) were developed for incident detection in urban areas. Each model had six inputs that represented detector count differences ("US1-MS1", "MS1-DS1", "US2-MS2", "MS2-DS2", "US3-MS3" and "MS3-DS3") and one output that represented incident_status. Simulated traffic data were used to develop, test, and evaluate all three models. The simulated data were generated using the commercial software "PTV VISSIM" simulator. Three different datasets (Dataset 1c, Dataset 2c and Dataset 3c) were used to develop the models and three additional datasets (Dataset 1v, Dataset 2v and Dataset 3v) were used to test and validate them. The performance of the models was assessed using three measurements: good_detection, no_detection and false_alarm rates. A comparison of the three models performances was presented. Finally, a trial for improving the performance of detecting incidents under different volumes was conducted and its effectiveness was evaluated.

6.2 Conclusion

Several important observations can be drawn from the analysis of testing and validation results. Fuzzy models built for single volume (FM1 and FM2) displayed better performance than the fuzzy model built for multiple volumes (FM3). The best detection was obtained using FM1 that was developed for a single volume (1000 vph) and the poorest detection was using FM3 that was built for different volumes (500, 1000 and 1300 vph). This suggests that there is no single set of rules that can fit all link volumes. It also shows the importance of considering the link volume when designing a fuzzy system for incident detection. Another important observation was the effect of the incident duration on the detection process. Incidents with duration of 10 minutes or larger were easier to detect than the ones with less than 10 minute duration due to their effect on the detectors' count patterns. It is an important factor since the suggested models were able to detect incidents where they had clear influence on the detectors' count patterns. In the combined models trial, there was a small improvement in the detection rate when compared to FM3. In the same trial, better results were obtained in terms of false alarm rate. The developed models displayed promising results and suggest that the detection of the suggested fuzzy models can be further improved.

6.3 Recommendations

In this section, few suggestions are provided to extend and improve this work as listed below:

- Use of real data to validate the models can give a more realistic indication of their incident detection ability.
- Use of more than one data source such as CCTV cameras can aid in incident detection and improve efficiency.
- Extend the ability of the suggested models to give additional information about the detected incident such as its location.
- Implementing the fuzzy models on a FPGA:

  The design of an FPGA-based fuzzy model can be very simple. It consists of an FPGA, analog-to-digital (A/D) converters for the inputs, a digital-to-analog (D/A) converter for the output and a ROM chip (McKenna, M.; Wilamowski, B.M., 2001).
- The trial that included traffic volume wasn't successful which led to having a separate model for each range of traffic volume and thence, three different models were introduced. Until now, no successful attempt has been made to accurately detect incidents using a single model for a relaxed range of traffic volumes. Even the models based on training, e.g., neural network based approaches, assume a relatively tight range of traffic volume similar to the ones used in this thesis. It would be worthwhile to investigate a new approach to develop a single model applicable to a wide range of traffic volumes. To date, no such work exists, but

based on the trials and models developed here, the following approach seems promising: The idea is to fuzzify the traffic volume to produce a volume indicator value (VIV) represented by a fraction between 0 and 1. In the simplest form, VIV can be set to the ratio of the traffic volume to the maximum traffic volume used during simulation to determine the fuzzy functions. During simulation, several volumes would be used to simulate incidents. Based on the mathematical relationship observed relating traffic volume and fuzzy function limits (e.g., range of Z, P and VP functions), VIV can be used to set the limits of fuzzy function Z, P and VP. Although, the mathematical relationship that relates traffic volume to fuzzy limits is not know, it can be extrapolated and used to map a VIV (based on an appropriately chosen traffic volume) to fuzzy function limits (or center value of each of Z, P and VP). The general idea is graphically depicted in Figure 33. In addition to the function center values, the fuzzy inference rules need not be the same for the different volumes; hence, there a method is needed to choose fuzzy rules based on VIV. To choose fuzzy rules based on VIV may require a type of training (neural network or other methods).

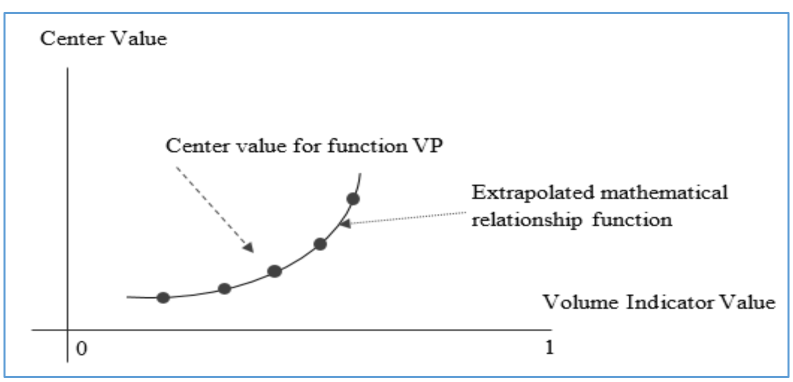


Figure 33: Graphical representation

Bibliography

Ahmed, F.; Hawas, Y. E. (2012). A Threshold-Based Real-Time Incident Detection System for Urban Traffic Networks. *Procedia - Social and Behavioral Sciences*, 1713-1722.

Albertos, P.; Sala, A. (1998). Fuzzy Logic Controllers: Methodology, Advantages and Drawbacks., (pp. 833-844). Valencia.

Balaji, K.; Juby, Z. N. (2007, November 26). *Fuzzy C-Means*. Retrieved from https://static.aminer.org: https://static.aminer.org/pdf/PDF/000/271/379/unsupervised_clustering_methods_for_medical_data_an_application_to_thyroid.pdf

Barria, A. J.; Thajchayapong, S. (2011). Detection and Classification of Traffic Anomalies Using Microscopic Traffic Variables. *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, 695-704.

Bay, Y.; Wang, D. (2006). Fundamentals of Fuzzy Logic Control – Fuzzy Sets, Fuzzy Rules and Defuzzifications. In Y. Bay, H. Zhuang, & D. Wang, *Advanced Fuzzy Logic Technologies in Industrial Applications* (pp. 17-36). Springer.

Cano, J.; Kovaceva, J.; Lindman, M.; Brännström, M. (n.d.). *AUTOMATIC INCIDENT DETECTION AND CLASSIFICATION AT INTERSECTIONS*. Retrieved from http://www-nrd.nhtsa.dot.gov: http://www-nrd.nhtsa.dot.gov/pdf/esv/esv21/09-0234.pdf

Deniz, O.; Celikoglu, H. B. (2011). Overview to some existing incident detection algorithms: a comparative evaluation. *Procedia Social and Behavioral Sciences*, 1-13.

Engin, Z. (2010). *A Short Fuzzy Logic Tutorial*. Ankara: Bilkent University.

Farradyne, P. (2000). *Traffic Incident Management Handbook*. Federal Highway Administration Office of Travel Management.

FHWA. (2013, December 3). *Review of Literature*. Retrieved from http://www.fhwa.dot.gov/: http://ops.fhwa.dot.gov/eto_tim_pse/docs/incident_mgmt_perf/section2.htm#incident

G. V. S. Raju, Jun Zhou . (1990). FUZZY LOGIC PROCESS CONTROLLER . *IEEE International Conference on Systems Engineering* (pp. 145 - 147). IEEE.

Hawas, Y. E. (2007). A fuzzy-based system for incident detection in urban street networks. *Transportation Research Part C 15*, 69–95.

Hawas, Y. E.; Mohammad, M. S. (2015). *A System for Incident Detection in Urban Traffic Networks*. Retrieved from http://itsc2015.exordo.com/: http://itsc2015.exordo.com/files/papers/631/final_draft/PID3793673.pdf

Jantzen, J. (1998). *Design Of Fuzzy Controllers*. Lyngby: Technical University of Denmark, Department of Automation.

Kaehler, S. D. (2003, May 11). *Fuzzy Logic Tutorial*. Retrieved from http://www.seattlerobotics.org: http://www.seattlerobotics.org/encoder/mar98/fuz/flindex.html

Kamboj, V.; Kaur, A. (2013). Comparison of Constant SUGENO-Type and MAMDANI-Type Fuzzy Inference System for Load Sensor. *International Journal of Soft Computing and Engineering (IJSCE)*, 204-207.

Klir, G. J. (1995). Fuzzy Logic. *IEEE*, 10 - 15.

La-inchua, J.; Chivapreecha, S.; Thajchayapong, S. (2013). A New System for Traffic Incident Detection Using Fuzzy Logic and Majority Voting. *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (pp. 1-5). Krabi: IEEE.

Lee, S.; Krammes, R. A.; Yen, J. (1998). Fuzzy-logic-based incident detection for signalized diamond interchanges. *Transportation Research Part C: Emerging Technologies* (pp. 359–377). Elsevier Science Ltd.

Lu, Y.; Ma, T.; Yin, C.; Xie, X.; Tian, W.; Zhong, S. (2013). Implementation of the Fuzzy C-Means Clustering Algorithm in Meteorological Data. *International Journal of Database Theory and Application*, 1-18.

MathWorks. (2015). *Comparison of Sugeno and Mamdani Systems*. Retrieved from http://www.mathworks.com: http://www.mathworks.com/help/fuzzy/comparison-of-sugeno-and-mamdani-systems.html

MathWorks. (2015). Fuzzy Inference Process. In MathWorks, *Fuzzy Logic Toolbox - User's Guide* (pp. 20-54). Natick: The MathWorks, Inc.

McKenna, M.; Wilamowski, B.M. (2001). Implementing a Fuzzy System on a Field Programmable Gate Array. *International Joint Conference on Neural Networks* (pp. 189 - 194). Washington, DC: IEEE.

MENDEL, J. M. (1995). Fuzzy Logic Systems for Engineering: A tutorial. *IEEE*, 345-377.

Mohaghegh, S. (2000). Virtual-Intelligence Applications in Petroleum Engineering: Part 3—Fuzzy Logic. *JPT - Journal of Petroleum Technology*.

Nedjah, N.; Mourelle, L. (2005). Fuzzy Operators. In N. Nedjah, & L. Mourelle, *Fuzzy Systems Engineering: Theory and Practice.* New York: Springer.

Ozbay, K.; Kachroo, P. (1999). Incident detection. In K. Ozbay, & P. Kachroo, *Incident Management in Intelligent Transportation Systems* (pp. 61-81). Norwood: Artech House.

Sameena Naaz, Afshar Alam, Ranjit Biswas. (2011). Effect of different defuzzification methods in a fuzzy based load balancing application . *IJCSI International Journal of Computer Science Issues*, 261-267.

Shahbandari, S. (2015, February 7). *Traffic congestion costs more than Dh700,000 per kilometre in Dubai*. Retrieved from http://gulfnews.com/news/uae: http://gulfnews.com/news/uae/traffic-congestion-costs-more-than-dh700-000-per-kilometre-in-dubai-1.1452783

Yaguang, K.; Anke, X. (2006). Urban Traffic Incident Detection Based On Fuzzy Logic. *IEEE Industrial Electronics, IECON*, (pp. 772-775).

Zadeh, L. A. (2008). Is there a need for fuzzy logic? *North American Fuzzy Information Processing Society, NAFIPS* (pp. 1 - 3). New York City, NY: IEEE.

Appendix A

The following figures display part of the data (between minute 1800 and minute 2300) for trial 1 of FM3.

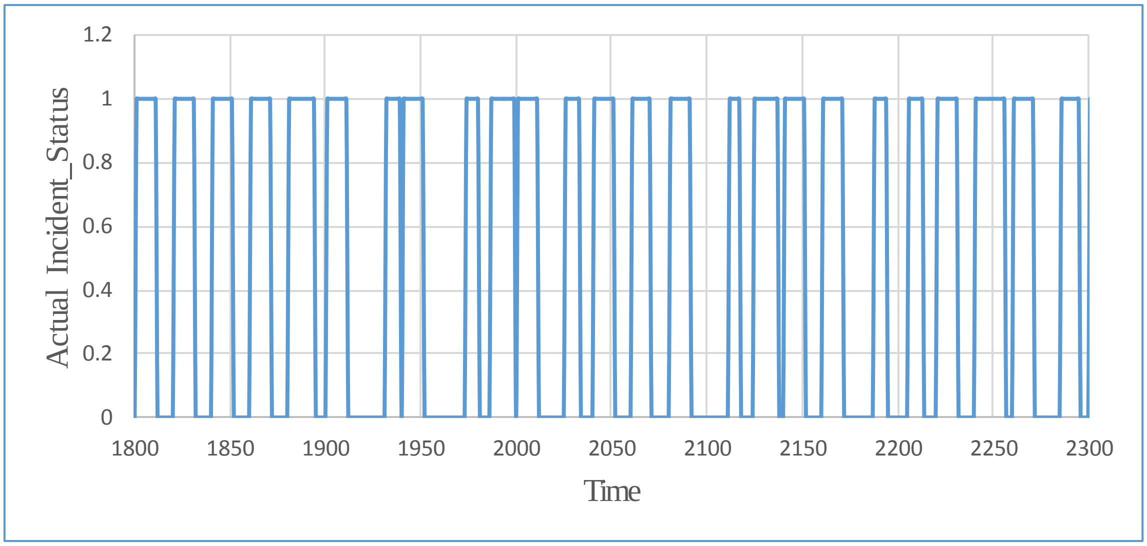


Figure 34: Actual incident_status for “Dataset 3v”

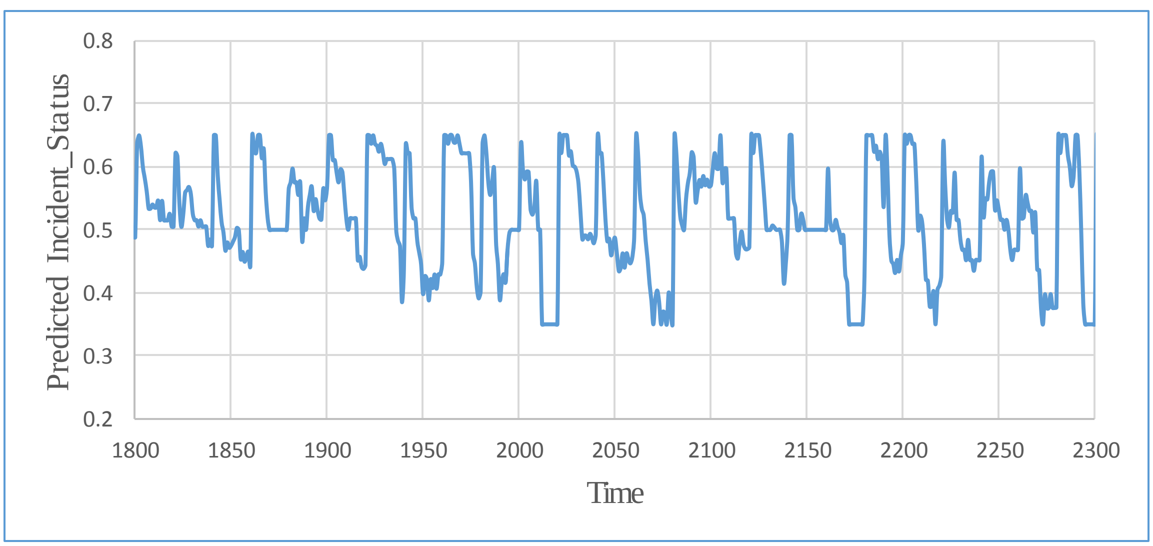


Figure 35: FM3 detected incident_status for “Dataset 3v”

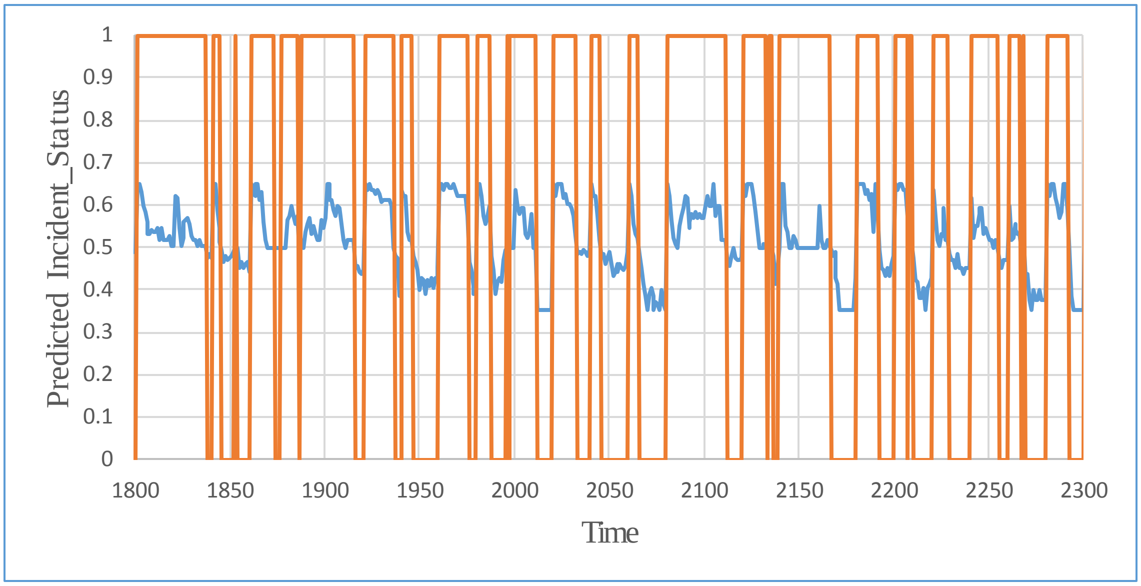


Figure 36: Detected incident_status before and after rounding for "Dataset 3v"

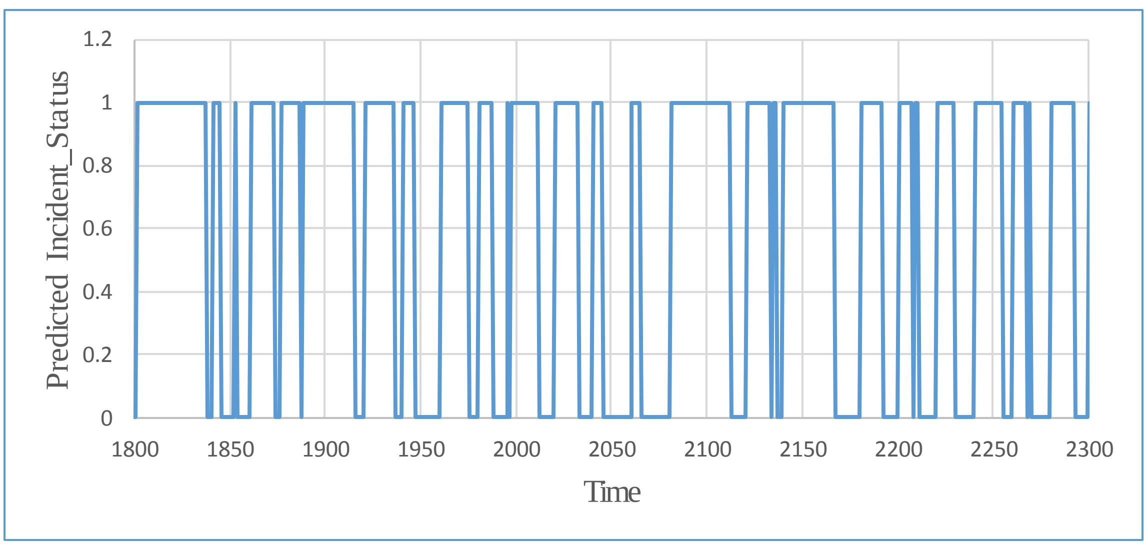


Figure 37: Detected incident_status after rounding for "Dataset 3v"

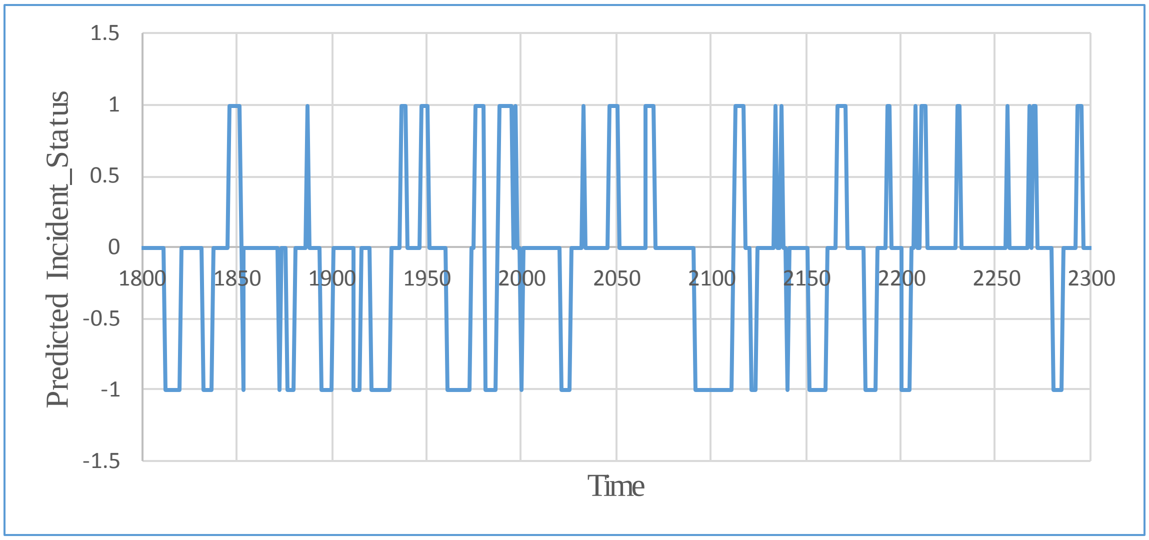


Figure 38: Difference between actual and detected incident_status for “Dataset 3v”

The following figures display part of the data (between minute 1500 and minute 2000) for trial 2 of FM3.

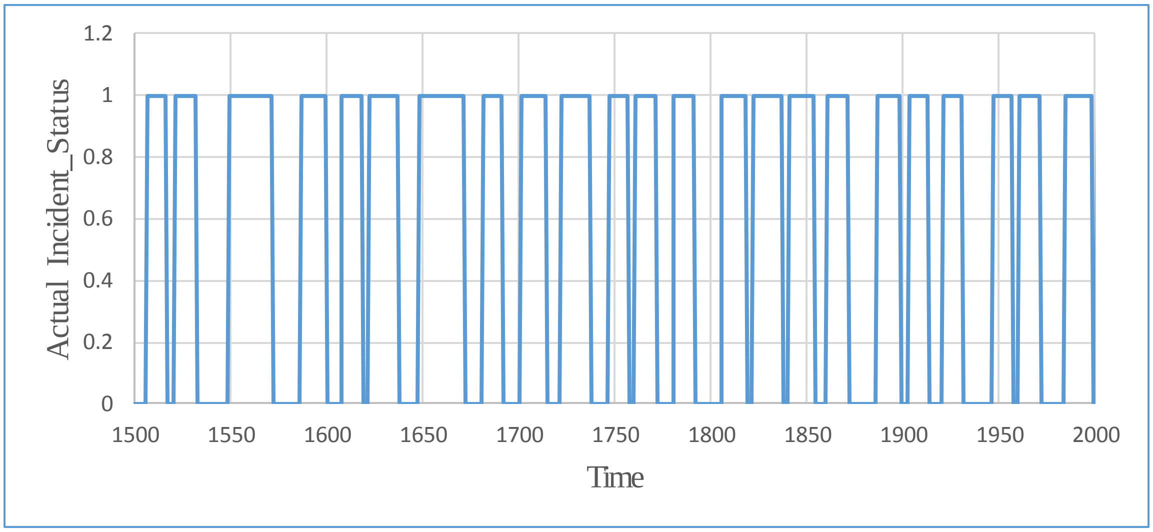


Figure 39: Actual incident_status for “Dataset 3v-a”

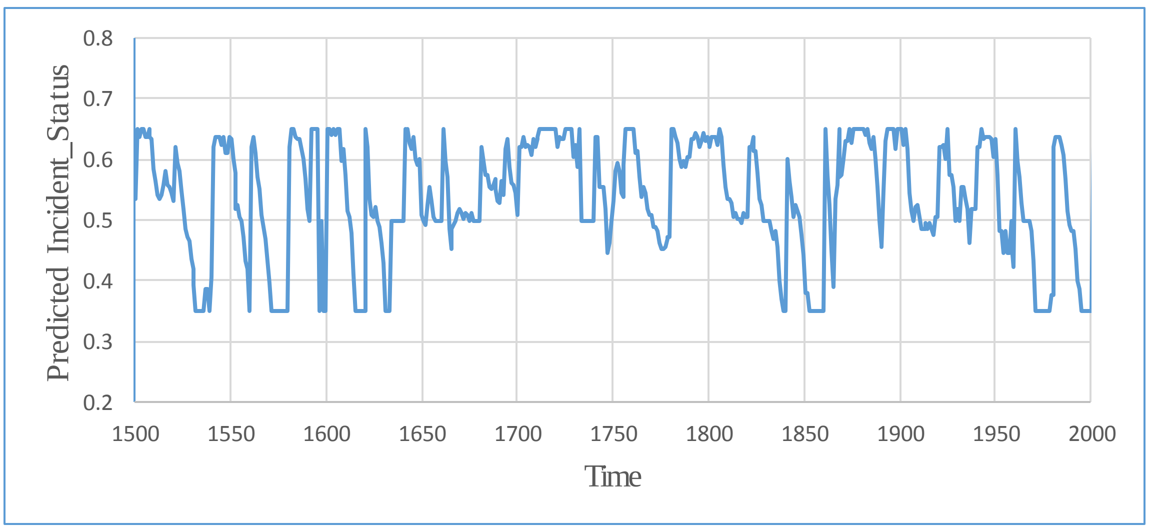


Figure 40: FM3 detected incident_status for “Dataset 3v-a”

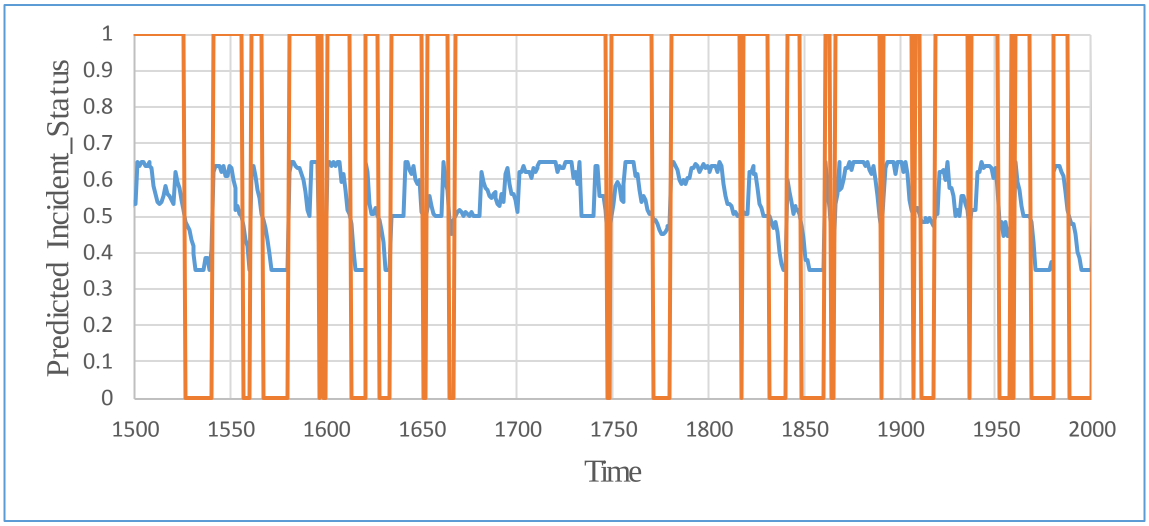


Figure 41: Detected incident_status before and after rounding for “Dataset 3v-a”

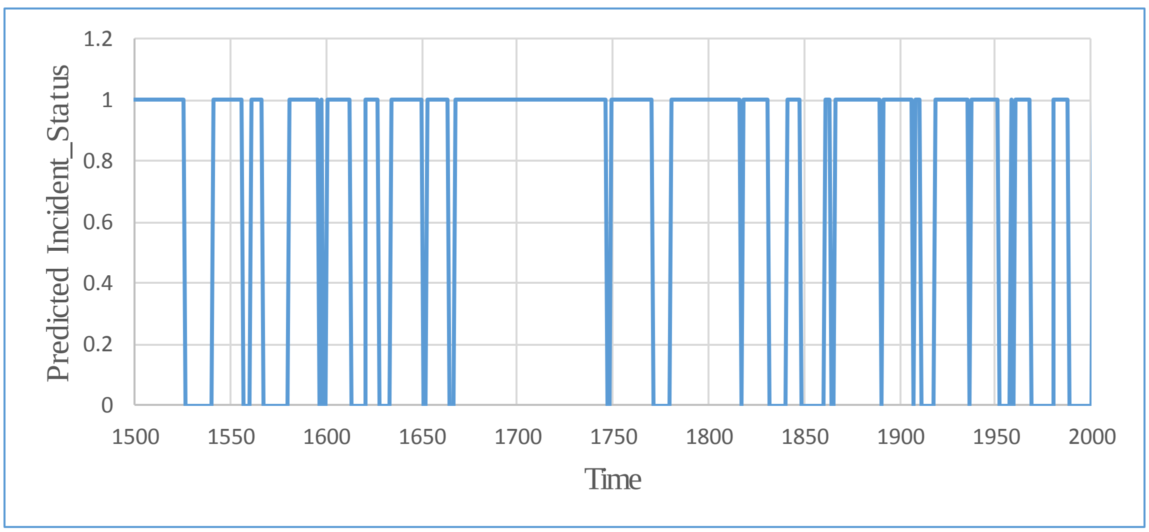


Figure 42: Detected incident_status after rounding for “Dataset 3v-a”

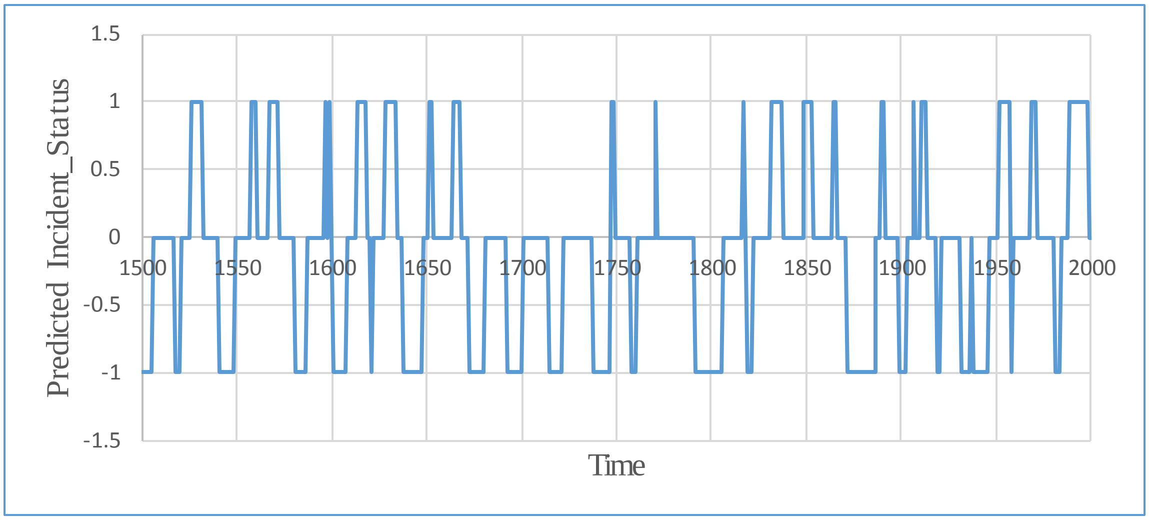


Figure 43: Difference between actual and detected incident_status for “Dataset 3v-a