Artificial Intelligence and Cybersecurity: Building an Automotive Cybersecurity Framework Using Machine Learning Algorithms.

by

Nevrus Kaja

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in The University of Michigan-Dearborn
2019

Doctoral Committee:

Associate Professor Di Ma, Co-Chair
Professor Adnan Shaout, Co-Chair
Associate Professor Hafiz Malik
Professor Armen Zakarian

Nevrus Kaja

nkaja@umich.edu

ORCID iD: 0000-0002-9625-7032

DEDICATION

In dedication to my fiancée and my family, both of which have always been a tremendous support for me, pushing me further with every step of my career and life.

ACKNOWLEDGEMENTS

I would like to thank and express my appreciation to my undergraduate and graduate advisor, Dr. Adnan Shaout, for his contribution and effort in developing, advising, and supporting me during my studies. Dr. Shaout has been a true mentor for the last eight years, and without his support, I would not have been able to achieve such an accomplishment. I also wish to thank my committee co-chair, Dr. Di Ma, and the other members of the committee, Dr. Hafiz Malik and Dr. Armen Zakarian, for their support throughout my studies.

Most importantly, I would like to thank my loving and supporting family. My fiancée, Anisa, who has always been a life mentor, a supporting pillar, a friend, and a true partner throughout this journey. Her love, consulting, and motivation have always guided me down the right path. My parents, their endless love, and my sisters have always encouraged me forward and been proud of every step I have taken. This accomplishment would not have been possible without them.

In addition, I would like to thank my friends who have always pushed me to go further. I would like to express my gratitude to my colleagues at work and the whole department of Electrical and Computer Engineering at the University of Michigan – Dearborn. This University prepared me to conquer the future.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Automotive technology has continued to advance in many aspects. As an outcome of such advancements, autonomous vehicles are closer to commercialization and have brought to life a complex automotive technology ecosystem [1]. Like every other technology, these developments bring benefits but also introduce a variety of risks. One of these risks in the automotive space is cybersecurity threats. In the case of cars, these security challenges can produce devastating results and tremendous costs, including loss of life. Therefore, conducting a clear analysis, assessment and detection of threats solves some of the cybersecurity challenges in the automotive ecosystem. This dissertation does just that, by building a three-step framework to analyze, assess, and detect threats using machine learning algorithms.

First, it does an analysis of the connected vehicle threats while leveraging the STRIDE framework[2].

Second, it presents an innovative, Fuzzy based threat assessment model (FTAM). FTAM leverages threat characterizations from established threat assessment models while focusing on improving its assessment capabilities by using Fuzzy logic. Through this methodology, FTAM can improve the efficiency and accuracy of the threat assessment process by using Fuzzy logic to determine the “degree” of the threat over other existing methods. This differs from the current threat assessment models which use subjective assessment processes based on table look-ups or scoring.

Thirdly, this dissertation proposes an intrusion detection system (IDS) to detect malicious threats while taking in consideration results from the previous assessment stage. This IDS uses the dataset provided from Wyoming Connected Vehicle Deployment program [3] and consists of a two-stage intrusion detection system based on supervised and unsupervised machine learning algorithms. The first stage uses unsupervised learning to detect whether there is an attack present and the second stage classifies these attacks in a supervised learning fashion. The second stage also addresses data bias and eliminates the number of false positives. The simulation of this approach results in an IDS able to detect and classify attacks at a 99.965% accuracy and lowers the false positives rate to 0%.

Chapter 1. Introduction

Urban populations are increasing at a quick pace [4]. This densification of the population is playing an important role in urban mobility and transportation overall. Considering these factors and customer demands, almost every major automotive manufacturer is researching advanced vehicle connectivity technologies and working on plans to launch driverless (autonomous) cars. Testing is currently underway, and experiments are growing rapidly. The prolonged future of autonomous cars is right around the corner and customers are waiting to exploit them. This new form of transportation will have a huge impact on our society. It will not only lead to a boom in innovations, but it will also bring the potential for new challenges which require innovative ways of thinking and solving [5].

From the other side, vehicle cybersecurity research and experiments have shown that malicious attackers can penetrate a broad range of physical and remote attack surfaces in a car. Multiple research papers such as [6]–[10] give clear examples of vehicle breaches and have already made news headlines. With increased complexity and accessibility in intelligent automotive systems, the potential for additional attacks and vulnerabilities against safety and privacy increases even more. To cope with these challenges, this dissertation studies and proposes a three-step framework, which solves some of the cybersecurity issues in the V2X automotive area using machine learning algorithms.

1.1. Problem Definition

The problem definition for this dissertation evolves from the following issues/gaps in the available research area.

- Cybersecurity in automotive systems is a widely researched area, but specifically intrusion detection systems in V2X BSM datasets are not explored. Because connected automotive technology has not been widely deployed and commercialized, there are not many holistic studies for V2X cybersecurity as related to threat analysis, assessment, and detection [11] .
- From the literature survey, multiple models for threat assessment and characterizations do exist such as EVITA, NHTSA, HEAVENS, OCTAVE, and others [11]–[13]. These threat assessment techniques are key to private organizations attempting to define the severity of a threat, but there is no defined framework which uses and considers this output in the specific implementation of other security systems (i.e., intrusion detection systems). This dissertation establishes a correlation between threat assessment and intrusion detection algorithms.
- All of threat or risk assessments studied in this dissertation are done in a subjective manner, using tables, or scoring. During this process, the threat characterization assignment is not done using discrete variables. They are often assessed based on users prospective using linguistic variables and non-discrete definitions. Due to the Fuzzy nature of this process, this threat assessment process often leads to inefficient scoring and inaccurate assessments. This research is one of the firsts to attempt using a Fuzzy logic method to resolve threat assessment inefficiencies.

- When considering intrusion detection systems in the automotive V2X, they are often not seen as a feasible solution for implementation either due to a high number of false positives or due to high computational requirements. This dissertation attempts to propose a feasible IDS for implementation.
- Lastly, there is simply not enough publicly available V2X or DSRC datasets for research, so the overall results are scarce. This dissertation uses one of the latest datasets released in this field.

1.2. Purpose and Significance of the study

Autonomous and connected vehicles have promised to lower the number of accidents caused by human behavior[14]. Some studies estimate that vehicles enabled by connected and self-driving technologies could reduce the number of accidents by up to 40% [15]. NHTSA estimates that V2X safety applications could eliminate or mitigate the severity of up to 80% of unimpaired vehicle crashes[16]. So, in overall the business case for connected and autonomous vehicles is clear; They will save lives and money if implemented correctly. Although there is a high potential for life-saving solutions, cybersecurity can be a bad syndrome preventing technology adoption. If these systems (autonomous or connected) are prone to cybersecurity attacks, then there is a chance that the number of accidents can increase significantly rather than decrease. Because the control is being passed from humans to technology, cybersecurity threats can do significantly more damage than they would in a driver-controlled scenario. The purpose of the research in this dissertation is to lay out those foundations as well. This study attempts to do quantitative and qualitative research in V2X cybersecurity while using artificial intelligence-based technologies.

V2X security is also a relatively new area from a research perspective. V2X has not been widely deployed from automotive manufacturers yet, and that is the main reason why there is not enough available research on its security detection aspects. Currently, there are not many extensive and complete datasets available; therefore, there has not been a lot of experimentation with the intrusion detection systems in V2X systems. A simple search for "intrusion detection systems in v2x" in major research databases yields almost no relevant results. The dataset [17] used in this dissertation was gathered and compiled only a few months prior to being used. Therefore, this research study is significant from a qualitative perspective and one of the first research works using this particular V2X dataset from a quantitative perspective as well.

1.3. Research Hypotheses

The core question of this dissertation is: how can Artificial Intelligence-based technologies improve cybersecurity in automotive connected and intelligent systems? According to this question, there are three main hypotheses that this dissertation:

- Use of the Fuzzy Logic methodology increases the efficiency and accuracy in threat assessment models.
- Two-stage Intrusion Detection System using supervised and unsupervised learning is a feasible solution for V2X threat detection using basic safety message data.
- There is a correlation between threat assessment and the algorithm or the configuration of an Intrusion Detection System.

1.4. Limitations

Throughout this dissertation, there have been a set of general and specific limitations. Some of these limitations are listed below.

- There are a limited number of datasets available to evaluate intrusion detection solutions in the V2X – DSRC space. Basic Safety Messages are used from the Wyoming Connected Vehicle Pilot project [17]. By using a newly released dataset with no published research available at the time of the study, there is a limitation to performing state-of-the-art benchmarking and comparison with other relevant works.
- There is limited processing power available in today's vehicles, making it challenging to fully explore advanced performance-based and machine learning algorithms in building intrusion detection systems. That is why Deep Learning-based techniques have been avoided in the second stage of the proposed intrusion detection system.
- Threat analysis has been based on currently available research and not every threat has been tested in an actual vehicle or been taken into consideration for Threat vs. IDS relationship as shown in 6.1 [18].
- When assessing threats, there is no ground truth to measure the performance of the proposed FTAM solution. This is because different organizations assess threats in a different manner and this process is organization or technology specific. To cope with this, the FTAM performance is benchmarked and compared with other available models rather than validation with ground truth. It will be up to the end users to validate such a model.
- Due to the nature and limited duration of the study, there is a limitation in establishing a defined relationship between the assess and detect phases. This dissertation does prove

the concept of the relationship, but further detailed and functional testing in real-world datasets need to happen before these correlations are concluded.

1.5. Research Methodology and Procedures

This study started with an extensive literature review of the following areas:

- Automotive technology and connectivity;
- Fuzzy Logic theory;
- Cybersecurity and Intrusion Detection Systems;
- Supervised and Unsupervised Learning algorithms.

The results from the literature survey will be provided in Chapter II. This dissertation uses methods, results, literature survey, and builds upon my previously published work as referenced in [19]–[26]. Problem definition as provided in 1.1 identified gaps in the current state-of-the-art development. After a holistic view of the area, a three-step framework was then defined with the objective of covering some of these gaps and proving out the claims laid out from the research hypothesis given in 1.3. These steps are provided below and follow their respective methodologies.

- Analyze: In this step, an analysis of automotive threats with a focus on V2X is done. This dissertation attempts to stay close to established standards, so the proposed solution has a higher chance of implementation and adoption. In this step, Microsoft STRIDE model is used for threat analysis [27]. Results are shown in Chapter III.
- Assess: The assessment stage builds a Fuzzy-based Threat Assessment Model (FTAM). In this stage, five established models are used to benchmark and drive the design of the new proposal. FTAM attempts to close gaps and eliminate or mitigate drawbacks

identified from the literature survey for models. Although a new model is proposed, all the threat characterization levels are based on established models. From a Fuzzy logic perspective, a multistage methodology proposed from [28] is used. Building six Fuzzy inference systems is time-consuming, but all the Fuzzy rules are accounted for due to the possibility that they can all happen. The methodologies of Mamdani [29] and Sugeno [30] were considered for use in this stage. Sugeno is more computationally efficient and works better with linear and mathematical techniques. This solution required a more efficient method for human input, and that is one of the main reasons why FTAM uses the Mamdani method. As it will be described in 1, multiple established threat assessment models were used to benchmark and drive the design based on STRIDE threats.

- Detect: The third detection stage aims to design an intrusion detection system. The methodology used in this stage is similar to a two-stage architecture approach used in previously published papers [20], [21], [31]. This method is validated to reduce the number of false positives, lower bias in data, and help with computational requirements in a variety of applications (vision and cybersecurity). The first stage aims to simply detect a threat, while the second stage attempts to classify it and increase the accuracy and lower the number of false positives. This stage uses the dataset from the United States Department of Transportation, Intelligent Transportation Systems Joint Program Office – Wyoming Connected Vehicle Pilot Deployment Program [3], [17]. This dataset will be described in 5.1.2.

The rest of this dissertation is organized as follows:

- Chapter I gives an overview of the paper, along with problem definition, purpose of the research, research methodologies used and its limitations.

- Chapter II gives the literature review on automotive cybersecurity, connectivity, and intrusion detection systems.
- Chapter III performs an analysis of different threats in the automotive arena with a focus on V2X ones.
- Chapter IV is focused on the assessment phase. After a review of current existing threat assessment models, it goes on to explain in detail the proposed Fuzzy based Threat Assessment Model (FTAM).
- Chapter V focuses on intrusion detection systems. It first provides an overview of the dataset used in this dissertation and then proposes a new, two-stage intrusion detection system. This IDS leverages the results from the FTAM proposed in Chapter IV.
- Chapter VI attempts to bring everything (Chapter III, IV and V) together in order to review and evaluate the complete framework. In addition, it provides details on results, performance evaluation, benchmarking, and conclusion.

Chapter 2. Literature Survey

This chapter will expand on the literature survey for this dissertation. Section 2.1 will provide a historical perspective of the automotive industry, and 2.2 will give some insights on the vehicle architecture. Section 2.3 will start discussing automotive cybersecurity, 2.4 will dive deeper on the connected vehicle aspects, and the last section 2.5 will discuss different types of Intrusion Detection Systems (IDS) and their state-of-the-art results.

2.1. Automotive Industry: A historical perspective

The automotive industry has been around since 1770 when French engineer and mechanic Nicolas-Joseph Cugnot created the first self-propelled road vehicle [32]. Since then the technology has advanced, and an estimated number of 100,000 patents have been created from such evolution [33]. The revolution and wide commercialization came in 1913, when Henry Ford launched Ford Model T, an iconic car in the automotive industry. The continued evolution of automotive has brought to life many innovations and autonomous vehicles. From a technology perspective, autonomous cars have been in research long before today. The ability of cars to drive themselves has been an inspiring problem for many researchers and automotive companies. Currently, there are three main areas where the automotive field is heading towards[34]:

- Connectivity
- Autonomous mobility
- Electrification

All three of these include certain cybersecurity aspects within.

While Henry Ford once said that cars are “opening the highways to all mankind,” autonomous vehicles now are enabling mobility to all mankind [35]. They have the potential to revolutionize the way we move. Every breakthrough or advancement in technology comes with security and other risks [36] . Autonomous vehicles will have a huge impact on society but also face a few barriers before they go in full commercialization. The paper in [37] gives autonomous vehicle barriers and impacts as shown in Table 2.1.

Table 2.1: AV Impact and Barriers [37]

Impacts	Barriers
Safety	**Security**
Congestion reduction	AV Legislation
Travel behaviors	Litigation
Freight transportation	**Privacy**
Vehicle ownership	Vehicle costs
Economic factors	Digital Mapping
Urban development	Infrastructure
Mobility	
Insurance Models	

As seen in Table 2.1, security and privacy are barriers in autonomous vehicle development, and they will result in an impact on safety and other factors. Research has been the lead of advanced technologies in vehicle development. To achieve and develop autonomous and other advanced automotive technologies, it requires extensive study and collaboration among many different areas. Major universities have put a significant amount of resources into autonomous and connected vehicle research, and they have been a crucial driver in this area. Some of the major research universities along with their centers are provided in Table 2.2.

Table 2.2: Autonomous Vehicle Research Universities

Research University	Description
Carnegie Mellon University	Carnegie Mellon's robotics laboratory has developed and published research in autonomous vehicles and automated driving for over 30 years [38]
Tsinghua University	Tsinghua University has a massive research program focused on automated driving and electric vehicles. In 2016, they opened a Joint Research Center for Intelligent Mobility with Nissan [39].
Stanford University	Stanford AI Lab was one of the pioneers to win the 2005 DARPA Grand Challenge in autonomous vehicles. Since then, they have been one of the leads in AV research [40].
University of Michigan	Mcity is a 32-acre, 10-million-dollar one-of-a-kind urban test facility for AV. The project scope looks at AC research but also evaluates how autonomous driving will shape urban planning [41].
Massachusetts Institute of Technology	MIT explores the various dimensions of autonomous cars. Their Media Lab is researching the use of blockchain technology or morality for self-driving cars [42].
University of California – Berkley	DeepDrive program at Berkley focuses on researching computer vision and machine learning technologies for automotive applications [43].

Table 2.2 is not an exhaustive list of all universities or research groups conducting state-of-the-art research in the automotive industry, but it provides some of the pioneers in the field. These university research centers have also served as the main sources from which this dissertation has derived its literature review.

Autonomous vehicles are coming to the market gradually. As their evolution goes on, Society of Automotive Engineers (SAE) have developed the J3016 standard to define the level of automation for on-road vehicles [44]. Table 2.3 provides a description of this standard.

Table 2.3: SAE J3016 Autonomy Levels [44]

Autonomy Level	Name	Vehicle Control
0	No Automation	Full driver control. Human drivers perform all driving actions.
1	Driver Assistance	Driver assistance technologies. The human driver is assisted by the system only in specific functions.
2	Partial Automation	The system can perform partial automation. The driver should always be ready to take over.
3	Conditional Automation	System drives and monitors environments. The driver should be ready to take over when requested by the system.
4	High Automation	The system handles the driving and monitoring environments. The driver is not required after the system takes over.
5	Full Automation	The system handles all aspects of driving.

Although autonomous vehicles are being deployed and put into production by automakers, it has required a set of industry, governmental, non-profit, and academic institutions to bring together important pieces to enable such a system. A chronological series of these events and contributions from these players is shown below:

- 1948 – Modern cruise control invented
- 1968 – Electronic cruise control
- 1980 – Carnegie Mellon University's Navlab
- 1995 – Laser-based adaptive cruise control
- 2001 – Lane-departure warning system
- 2003 – Pre-crash mitigation system
- 2004 – DARPA Grand Challenge
- 2010 – Google driverless car debuts
- 2012 – Nevada licenses autonomous cars
- 2014 – SAE J3016 created
- 2014 – NHTSA issues a draft of proposed rulemaking for AV

- 2015 – University of Michigan's MCity
- 2019 – Multiple pilots on autonomous vehicles. Commercially available programs deployed (i.e., Aptiv-Lyft Partnership in Las Vegas or Waymo One)

Table 2.4 provides plans from the major automotive manufacturers about their autonomous vehicle production. Please note that not all these plans are confirmed from their respective manufacturers and this information is current as of the time this paragraph was written (6/2018).

Table 2.4: OEM Autonomous Vehicle Planned Launches

OEM	Autonomy Level	Planned Launch	Comments
Audi	3 4	2018 2021	World's first-to-market Level 3 AV [45]
Tesla	4	2019	Tesla is currently selling "Self-Driving Capability" [46]
Ford	4	2021	Ride-sharing or ride-hailing production [47]
Mercedes	4	2021	"Drive Pilot" – partnership with Bosch [48]
BMW	4 5	2021 2030	Partnership with Intel and others [49]
Volvo	4	2021	Partnership with NVIDIA [50]
Honda	3 4	2020 2025	Vision to create a collision-free society [51]
Hyundai	3 4	2020 2030	Targeting for the highway in 2020 and urban driving in 2030 [49]
Toyota	4	2020	Self-Driving on the Highway [49]

2.2. Advanced Vehicle Architecture

To understand vehicle cybersecurity, one must start by looking at the recent vehicle architectures and their points of interaction. For the purposes of this dissertation, we define an "advanced vehicle" as a recently developed vehicle system with multiple complex automotive technology systems. These vehicles include a wide range of sensors, actuators, driver assist systems, and in cases also offer various levels of autonomous driving capabilities. The terms

"Advanced Vehicles" or "Autonomous Vehicles" are used interchangeably to depict such systems throughout this dissertation. The following three items are the major component categories that have major potential vulnerabilities in regards to cybersecurity [52]:

- Sensors used to monitor the surrounding environments and make critical decisions on its driving patterns.
- Communications provide a resource for the vehicle to interact and engage with other services to provide convenience and safety features for the driver.
- Computing power and capabilities to manage and process the required information.

Sections below will provide a summary of these components in each of the categories. Figure 2.1 provides a detailed architecture for an autonomous or advanced vehicle system.

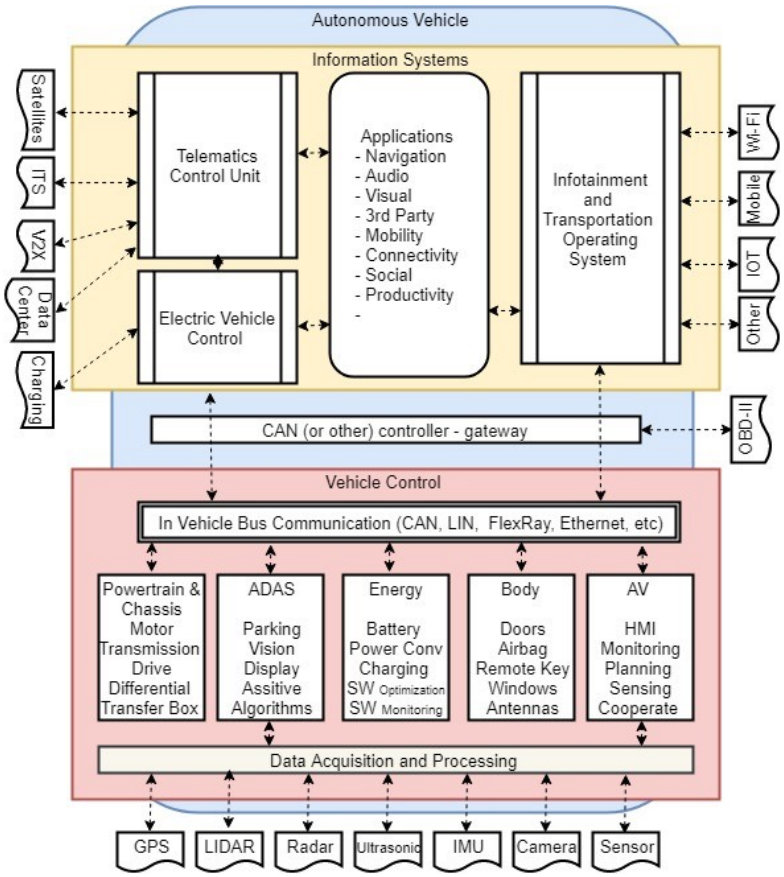

Figure 2.1: Advanced or Autonomous Vehicle Architecture

2.2.1. Vehicle Sensors

The objective of vehicle sensors is to provide detailed monitoring of the surrounding environment and offer redundancy to maximize safety. Figure 2 provides an overview for a typical autonomous vehicle's suite of sensors that work together to provide a 360-degree view of the environment around the vehicle.

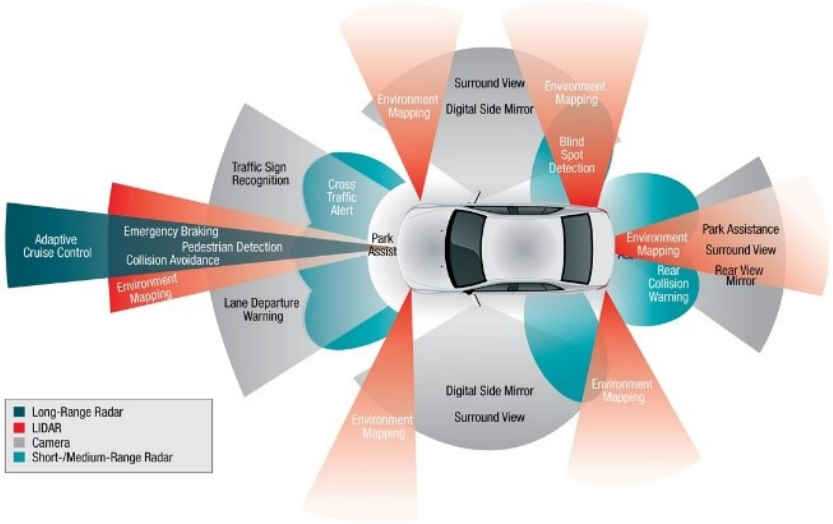

Figure 2.2: AV Sensors Diagram from [5]

Table 2.5 is a detailed list of these sensors for each category along with the estimated number of them found in a Level 4 Autonomous Vehicle.

Table 2.5: Vehicle Sensors

Sensor	#	Description
Ultrasonic	12	Provides short range distance for parking and backup.
Long Range Radars	2	Long range radar uses radio waves to determine long-range distances between the obstacle and the sensor.
Short Range Radars	6	Short range radar uses radio waves to determine short range distances between the obstacle and the sensor.
Cameras	10	Cameras collect images to monitor the environment.
LIDAR	1	LIDARs are 360-degree sensors that use a light beam to determine distance.
Infrared Sensors	X	Infrared sensors detect lane markings, pedestrians in low lighting by emitting or detecting infrared radiation.
GPS	1	Global Positioning System

2.2.2. Vehicle Communications

For any vehicle to engage and interact with itself, with the driver or with other systems, it must have a communication system. Today's advanced vehicle and tomorrow's autonomous vehicles contain a variety of communication interfaces. These interfaces are usually categorized in physical and remote connections. Figure 2.3 provides a visual representation of such connections.

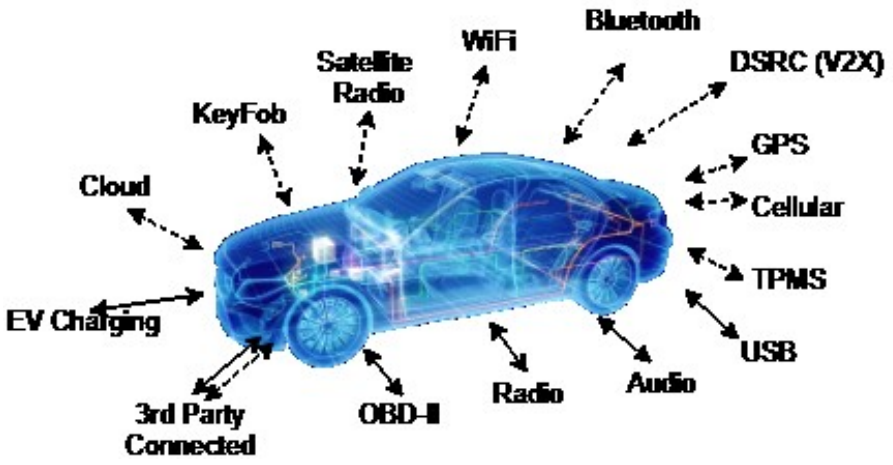

Figure 2.3: AV Communications

Disconnected arrows represent remote access while continuous arrows represent physical access.

2.2.3. Vehicle Computing

The third main element in vehicles related to cybersecurity is the computing power and algorithmic capabilities. Since the early 1980s, vehicles have implemented microprocessors in their vehicle designs, mainly due to the complexity of features being offered. Based on different estimates, an autonomous or advanced vehicle will include an average of over 100 ECUs. Often the challenge is that these ECUs come from different suppliers with different capabilities and responsibilities. So, to unify this system and make it secure, there is a great amount of effort required from OEMs and suppliers.

2.3. Automotive Cybersecurity

With autonomous vehicles, the automotive world enters an area where security takes another level. Traditional computer security has already proven to be challenging on its own, and its consequences have often been disastrous[53]. Autonomous vehicle cybersecurity is another step which adds complexity into the system and brings the potential for even bigger disasters.

Cyberattacks in a computer system result in mostly financial losses, but autonomous vehicle cyberattacks have the potential to impact human lives even in a small-scale attack.

Automotive security is often considered an emerging area, but vehicle security has been studied for quite some time. Traditionally, vehicles have been isolated systems, with critical components controlled by mechanical systems and separated from electronic or digital controls. This way, hackers have not been able to penetrate or control a vehicle due to the physical requirements for an attack. Other researchers and OEMs have reasonably argued that if a physical connection is required to perform cyberattacks then these attacks are simply not feasible or vulnerable enough [54][55]. One automotive cybersecurity solution is encryption-based. These solutions are proven to be inefficient for the CAN protocol [56], [57], due to computational and data overheads. To address the limitations of crypto-based solutions, researchers have proposed other methods such as Intrusion Detection Systems [56]. For instance, Hafeez et al. propose an intrusion detection at the physical layer. This is achieved by estimating the frequency response of each transmitter and training a neural network to use the frequency response as a signature to identify the electronic control unit/transmitter [58].

Lately, with the digitization of critical vehicle controls and drive-by-wire systems, the vehicle is turning into a "computer on wheels," opening the doors to many vulnerabilities and cyberattacks. Many of these vehicle systems (such as CAN) were not designed with security in mind, but to simplify in-vehicle communications or OEM processes. With the introduction of the internet, computers started to connect with each other more and more. This essentially brought a revolution in the technology industry, and researchers found it difficult to put security measures in place because the technology or internet protocols were not built with security in mind. A

similar analogy is happening in the car industry [59]. Now that cars are starting to have inter and intra communication systems, it will be difficult to put cybersecurity measures in place.

Vehicle security started to emerge in the last decade, once media put heavy attention to it. Table 2.6 shows a chronological series of events which have catalyzed the importance of vehicle security.

Table 2.6: Automotive Cybersecurity Developments

Year	Event	Security Compromise
2002	Forbes published an article "How to Hack your Car" [60] Many news outlets followed	Compromise engine, hybrid, and other performance factors
2005	Car Whisperer tool from Trifinite exploited standard passkeys in Bluetooth connections [61]	Bluetooth eavesdropping
2007	The Telegraph published news about Hackers intercepting navigation systems [62]	Navigation break-in
2010	Koscher et al. published "Experimental Security Analysis of a Modern Automobile" [63]	Disable brakes, stop engine using physical access
2011	Koscher et al. published "Comprehensive Experimental Analyses of Automotive Attack Surfaces" [8]	Exploits remote attacks
2013	Valasek and Miller published "Adventures in Automotive Networks and Control Units" [10]	Tutorial to Alter vehicle behavior via remote attacks.
2014	Valasek and Miller publish "A survey of remote automotive attack surfaces" [9]	The first resource for automotive network architecture review
2016	Valasek and Miller published "CAN message injection" [64]	Taking vehicle control via CAN message injection

2016	Keen Security Lab of Tencent published remote Tesla attacks [65]	Tesla vehicle vulnerabilities
2016	Liu et al. presented at DEFCON “Can you trust Autonomous Vehicles: Contactless attacks against sensors of self-driving vehicle” [6]	Sensor attack on self-driving vehicles

Back in 1985, only a few vehicle features were offered through electronic control, but nowadays and in the future, almost all the features are based on x-by wire or digital systems. While the vehicle technology and its complexity have increased exponentially, the vehicle cybersecurity capabilities have only developed linearly as shown in Figure 2.4 [66] [67]. This is often fueled by customer demand and the need for a fast-paced innovation in the automotive industry.

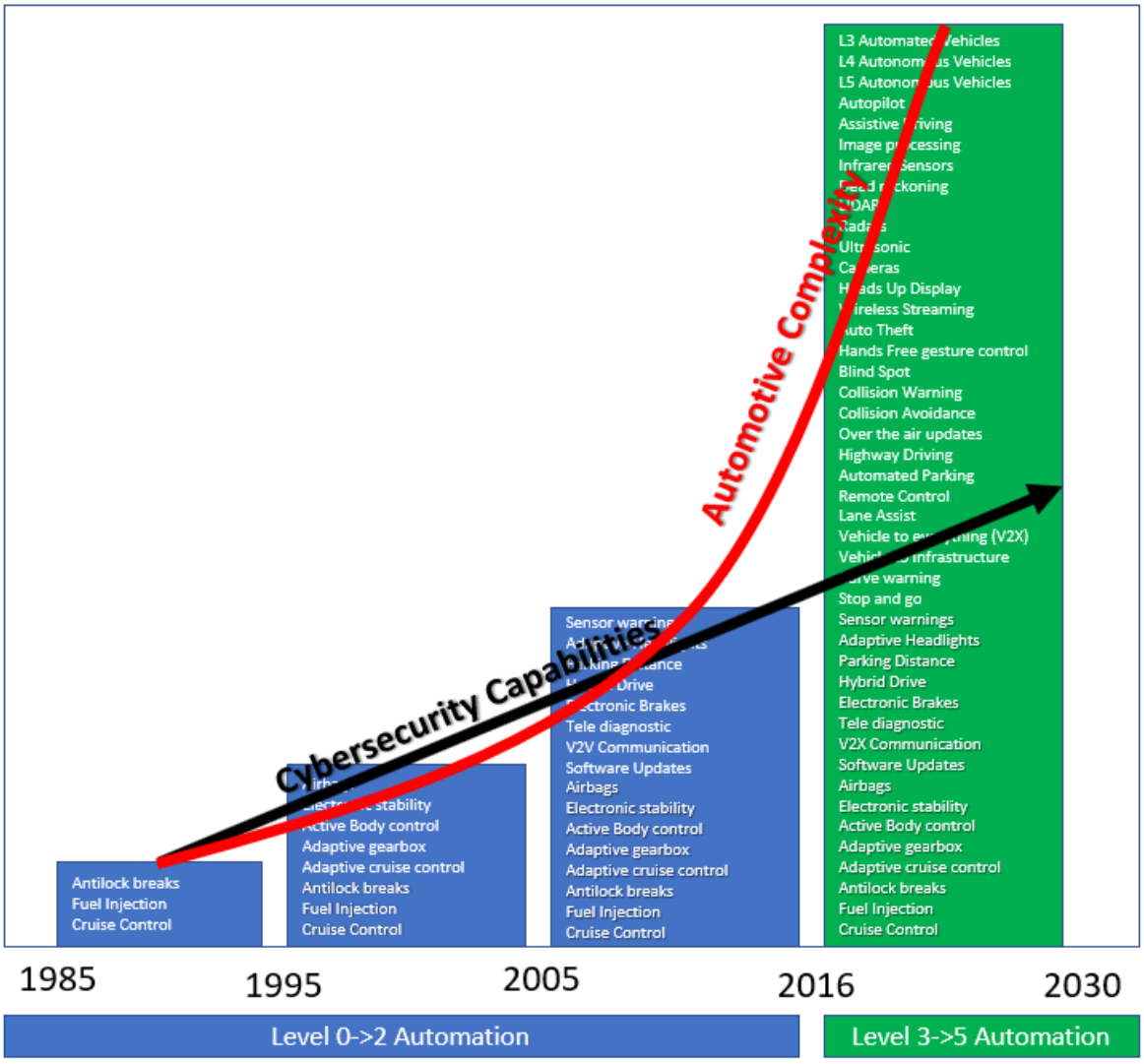

Figure 2.4: Automotive Complexity vs. Cybersecurity Capabilities

Through computing, communication, and sensors, many intruders can lunch malicious attacks into a vehicle. At the same time, a lot of research has been conducted in the area to find cybersecurity solutions. Figure 2.5 provides a high-level overview of where these solutions are focused and what each of them means.

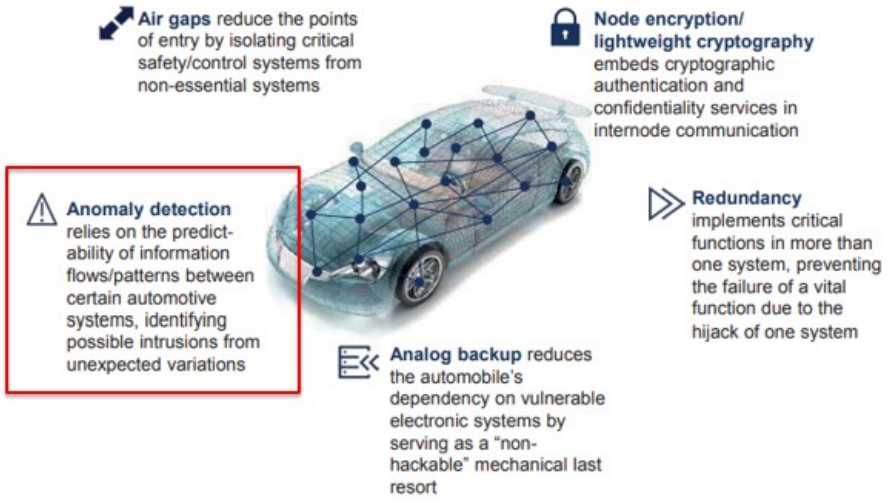

Figure 2.5: Design Solution Patterns [68]

This study primarily focuses on Anomaly Detection type of solutions.

2.4. Connected Vehicle: V2X

Vehicle to everything (V2X) is a communication protocol designed to enable intra vehicle communication. It provides a way for the vehicle (V) to communicate with other (X) systems. It is currently based on two competing standards: DSRC and Cellular V2X. These are two-way wireless communications with short to medium range which allows for high data transmissions in vehicle to infrastructure type of communications [2]. These protocols have an emphasis on active safety application and are designed to have different capabilities for both critical communications for safety systems and service-based communications for added functionalities.

According to FCC-03-324 [69], the FCC has reserved 75 MHz range in the 5.9 GHz band to be used for vehicle communication and mobility applications [70]. The purpose of these initiatives has been to ensure safe and secure connectivity between vehicles and other infrastructure components. These protocols promise to prevent a significant number of crashes and accidents in transportation, which account for around more than 35,000 deaths in the USA

alone [71]. Another main reason that V2X has been studied and implemented is to enhance mobility services within the transportation landscape. This will increase driver and passenger mobility along with new revenue streams and profitable business models from OEMs and other stakeholders. The subsection below provides more details on the DSRC protocol that is used for V2X.

2.4.1. DSRC – Dedicated Short-Range Communication

The basic idea behind the deployment of Dedicated Short-Range Communication (DSRC) is to enable applications for collision prevention. These applications are dependent on frequent data and message exchange between vehicles and other vehicles or infrastructure. Therefore, the DSRC standard defines a set of rules on how vehicles can exchange messages in a certain time frame for time-critical applications.

Figure 2.6 shows an example of a working DSRC-based collision avoidance system.

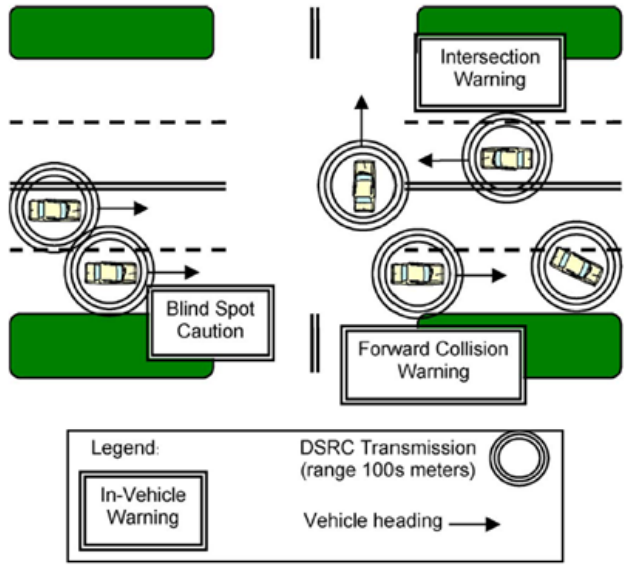

Figure 2.6: DSRC-based Collision Avoidance System from [70]

Each vehicle equipped with DSRC creates a 360-degree situational awareness around itself. Through this "cloud," it broadcasts information of its state in the form of Basic Safety Messages (BSM), which include speed, acceleration, location, control information, path tracking, etc. This information is usually sent out 10 times per second over a range of 100 meters. Every other vehicle within the range will also receive these messages from other vehicles equipped with DSRC. After receiving the message, the vehicle will estimate the trajectory motion of its neighbor, compare it with its own path, and compute if any of its surrounding vehicles poses any threat of collision [70]. In Figure 2.6, vehicles can prevent a collision in front of them, provide a warning when there is a vehicle at the "blind spot" or alert the driver of vehicles approaching the intersection. In addition to the V2V communication, vehicles can also communicate via roadside units using BSM messages. If a vehicle calculates that there is a potential collision approaching, the DSRC system will warn or assist the driver in controlling the vehicle.

To explain this further, consider the following use-case scenario: The driver of a highway car does not stop after seeing a stopped car within a certain distance. This is a typical scenario in highway accidents. In this case, other cars that do not have a safe distance or the driver is distracted will result in rear-ending the vehicle and causing a chain accident. If these cars are equipped with DSRC, a warning can be issued to the car and breaks could be applied automatically avoiding the collision. To enable this use case, DSRC has a protocol and standard that specifically defines how the vehicles will communicate and interact with each other. This communication is based on a standard allowing interoperability among devices.

Figure 2.7 shows a block diagram of the protocol stack for DSRC communication. Starting with the physical layer, DSRC uses IEEE 802.11p, the 11p is mainly for the PHY and the MAC

layer. Going a layer higher, DSRC uses a set of standards defined by IEEE 1609. IEEE 1609 includes 1609.1 for applications, 1609.2 for security services, 1609.3 for network services and 1609.4 for channel switching.

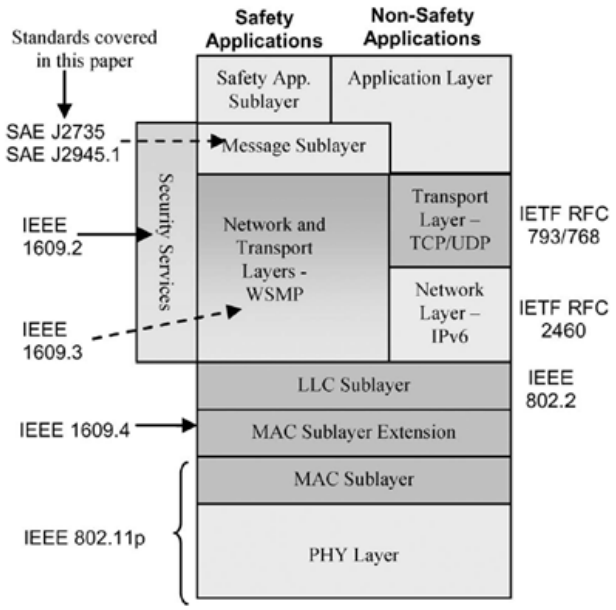

Figure 2.7: Architecture of DSRC Communication in the US from [70]

As a collision preventing mechanism, 802.11p leverages the Carrier Sense Multiple Access with Collision Avoidance (CSMA/ CA), and for modulation technique, it uses Orthogonal-Frequency Division Multiplexing (OFDM). MAC sublayer establishes rules to access the common medium which can be shared fairly and effectively among various sets of stations. The Logical Link Layer (LLC) is based on the IEEE 802.2. Going further up, the protocol defines the type of network layer mechanism, that is either IPv4 or IPv6. This can be used to control the MAC layer [70].

At the top, SAE J2735 specifies the message formats that supports many vehicles-based communication which includes the basic safety message (BSM), responsible for sending safety-critical vehicle state information. In this dissertation, BSMs are used to build an intrusion

detection system and inject attacks. In its competing standard, for C-V2X, these message sets are defined from CAM (cooperative awareness message) and DENM (decentralized environmental notification message. While this dissertation is focused on BSM / DSRC, the proposed IDS and other methodologies can be applied to C-V2X as well.

2.4.2. IEEE 1609.2 – Security for DSRC

IEEE 1609.2 is the standard which defines the security of message formats and their processing used by Wireless Access Vehicular Environments (WAVE) devices. This part describes methods to secure messages and applications within DSRC. As it will be described later in Chapter III, there are multiple kinds of attacks that can hit WAVE, and due to the critical nature of this technology, it is crucially important to detail the operations and have a standard to carefully unify those services [69].

Clauses 5 through 9 in [72] provide in detail the protocols, methods, specifications, and rules for all security services in WAVE. It is redundant to explain the whole standard in this dissertation but below some possible drawbacks are identified that might lead to attacks on this technology.

The latest version of this standard was released in March of 2016, and it includes some updates for pseudonymous data and CRL (Certificate Revocation Lists) format. One of the things that IEEE 1609.2 defines is a signed protocol data unit (PDU). These signed PDU security managements contain a provider service ID with permissions headers, validity checks, security management fields, a reference to certificates, etc. And as the name states, they should also be signed. The signature is contained within the message, and the message is signed by a certificate.

The certificate is issued by a CA (certificate authority), which is chained to the root certificate itself [72]. This is similar to web-based security and introduces the risk of CA compromising.

In addition, different from web-based security, vehicles are still behind on their connectivity, specifically if WAVE is retrofitted. Another drawback that raises here is how do you efficiently distribute CRLs in a vehicle environment when a CA is compromised?

2.5. Artificial Intelligence and Machine Learning

Artificial Intelligence (AI) is a growing field in the area of computer science. It focuses on using certain algorithms to train or build machines to perform tasks which mimic human intelligence. The term was used first in 1956 at Dartmouth College where computer scientists from Massachusetts Institute of Technology, Carnegie Mellon University, and IBM presented methods where computers would learn certain strategies [73].

The field of AI started to become popular claiming success in problem-solving with symbolic methods, rule-based systems, expert systems, etc. Defense Advanced Research Projects Agency (DARPA) completed multiple projects, and years 1956 through 1974 were known as the golden years of AI. During this time, research was focused on reasoning, optimization, natural language, etc. After a long period of funding but with no major breakthroughs, multiple agencies started cutting their research funds, and criticism started looming. Limited computational power, limited processing, and storage capabilities made AI enter into its first “winter” (1974–1980)[73].

The technology evolved, big data became a trend, Moore’s law continued its progression, processing capabilities became abundant, and so Artificial Intelligence started to shine back up.

This is also known as the rebirth of AI. Lately, Artificial Intelligence is used in areas like customer experience, human resources, fraud detection, predictive analytics, robotics, vision, gaming, medical science, etc. AI is starting to become pervasive across different industries. Learning to reason and make a decision based on data is a powerful "prediction tool" and it can be used in almost any area where human reasoning or decision making is required. While it is impossible to go into detail for all its uses or branches in this dissertation, one of the most predominant branches of AI is Machine Learning (ML).

Machine Learning (ML) is simply a set of algorithms which can train machines to do certain tasks without explicitly programming these tasks. This ability is what makes ML attractive in many areas, including cybersecurity. In this dissertation, supervised and unsupervised learning methods are used to build an Intrusion Detection System as shown in Chapter V. In addition, Fuzzy Logic as another branch of Artificial Intelligence is used to build the Fuzzy based Threat Model as it will be explained in Chapter IV. The sub-paragraphs below provide some review of all these three Artificial Intelligence branches.

2.5.1. Supervised Learning

In order to build a Machine Learning model, the algorithm has to "learn" in a set of data which is called training data. Supervised learning is a method used in Machine Learning where the training data contains labels. In the case of supervised learning, this data is labeled, which means that the data contains input variables (x) and output variables (y). The simplistic view of this method, is that the algorithm learns a mapping function from the input to the output $y = f(x)$ [74]. These types of algorithms are used in stage two of the proposed Intrusion Detection System

to classify the type of attacks. Some of the best-performing and most well-known algorithms in this space are:

- Neural networks
- Decision Trees
- Support Vector Machines
- Random Forest
- AdaBoost

2.5.2. Unsupervised Learning

Contrary to supervised learning, the algorithm "learns" on unlabeled data in unsupervised learning. This means that the training data contains only input variables with no output variables. The objective of these algorithms (also referred as clustering algorithms) is to learn by modeling and understanding the structure, behavior or distribution of the data [74]. These types of algorithms are used in stage one of the proposed Intrusion Detection System. Different algorithms are tested in this stage such as:

- Canopy
- Density Based K-Means
- Filtered Cluster
- K-Means
- FarthestFirst

2.5.3. Fuzzy Logic

Fuzzy Logic by its definition refers to all the methodologies used to categorize classes with undefined boundaries. This theory of methodology was developed from Lofti A. Zadeh in 1964

at the University of California, Berkeley. The inspiration came due to a need to solve real-world, complex problems, which often did not have binary or discrete representations. In a similar fashion, threat models and their characterizations are indeed cloudy or fuzzy rather than discrete. This is also the reason why this dissertation attempt to use Fuzzy Logic in Threat Assessment. Since 1964, Fuzzy Logic has been used in a variety of applications, but according to the literature survey in this work, there is no substantial research that attempts to use Fuzzy Logic in threat assessment.

Fuzzy Systems work on the premises of the following foundational elements:

- Fuzzy Sets: Data sets with smooth boundaries. Majority of the elements described in the previous threat assessment models can be considered Fuzzy sets because they do not have strict or describe boundaries.
- Fuzzy Rules: A rule used for knowledge representation describing the relationship between two linguistic variables.
- Linguistic Variables: Variables or properties as describes by Fuzzy sets.

As was mentioned earlier, Fuzzy Logic has had tremendous progress since its discovery. In [75], Yen and Langari do a thorough job at explaining Fuzzy Logic and providing its relationship with other areas. Let us look briefly at where Fuzzy Logic has been able to make an impact.

- *Approximate Reasoning*: One of the first and most predominant papers in this area is the one from Mamdani [76]. In this dissertation, Mamdani introduces his method and uses it as a basis for modeling and decision making. If-Then rules and propositional calculus allowed Fuzzy to be used as a predominant algorithm in a variety of such applications. The usage of Fuzzy Logic in Threat Assessment falls into this category.

- *Probability Theory*: Probability theory and Fuzzy Logic are often seen as two competing methodologies. But Chain, Zhu, and Bazzi in [77] provide solid research about the relationship of those two. They use applications such as signal detection in the presence of noise or image segmentation to illustrate the value of hybrid Fuzzy-probability techniques.
- *Control Engineering*: Fuzzy Control is essentially a control engineering technique based on the Fuzzy set theory. Many applications use Fuzzy Logic Controllers, which are a way to convert linguistic control methods into automatic control methods. Lee provides a good overview of such control techniques in [78].
- *Intelligent Controls*: As a control algorithm, Fuzzy Logic is also being used in intelligent control systems by representing its knowledge in an organizational or hierarchal structure. Currently, there are many applications in this area such as vision-based systems, autonomous systems, and industrial controls. Silva gives a variety of examples specifically focused in this area of his book [79].
- *Analytics*: Data science and analytics is an emerging field of today's technology. The abundance of data creation along with advancements in computing and algorithms have made this field an important one for discovering and solving a variety of analytical problems. Classic machine learning algorithms are often used, but they are not effective for model interpretation. Fuzzy Rule-based systems have proven to be a successful method for model and knowledge interpretation in big data applications around cybersecurity, finance, medicine, etc. Fernandez and others provide a relationship of big data models and Fuzzy representation of knowledge in their publication [80]. In addition,

Herrera gives a good view of Fuzzy systems in data science and big data in his lecture [81].

- *Pattern Recognition and Machine Learning*: Pattern recognition algorithms are mainly categorized in supervised and unsupervised techniques. Fuzzy Logic has been mainly used in unsupervised learning methods to improve accuracy and handle uncertainty or outliers. For example, the Fuzzy C-Means algorithm (FCM) improves the traditional c-means algorithm allowing a data point to partially belong in multiple clusters at the same time. A similar method also exists for the K-Nearest neighbor algorithm and is usually referred to as Fuzzy-KNN. Friedman and Kandel's book [82] explains the usage of Fuzzy models in pattern recognition.
- *Cybersecurity*: One of the chapters in [83] describes the usage of Fuzzy in computer security. An area where Fuzzy has been successful in Fuzzy logic is the intrusion detection system due to its ability to predefine and discover attack models. Fuzzy logic is mostly used in signature-based Intrusion Detection Systems. In addition, anti-virus companies have used Fuzzy methods to detect viruses. In this aspect, Fuzzy is used more for pattern matching as described in the paragraph above. This dissertation work uses Fuzzy Logic for threat assessment analysis rather than intrusion detection systems.

2.6. Intrusion Detection Systems

As seen in Figure 2.5, anomaly detection is one of the areas for vehicle cybersecurity solutions. These solutions are often called Intrusion Detection Systems. IDS are software applications that usually monitor network traffic for suspicious activity or any malicious attacks. They are evolving in their nature as well to keep up with the progress. They usually work by modeling the normal behavior of a system and then comparing it with the current monitored state

of the system. When an action significantly deviates from normal behavior, then the system analyses such behavior and categorizes the actions into a certain anomaly [31]. This is often called anomaly-based detection, and it works on the premise of real-time data. In most general scenarios, attacks start with probes and sweeps against the network hardware. Intrusion Detection Systems (or Intrusion Prevention Systems – IPS) look at the data to identify these sweeps and probes and give an alert against an expected attack behavior. Generally, IDS systems have the architecture shown in Figure 2.8.

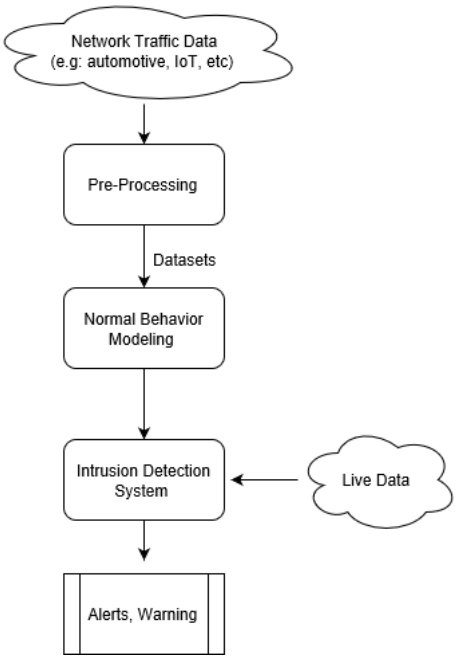

Figure 2.8: Intrusion Detection System Architecture

IDS are separated into the following two main categories: signature-based systems and anomaly-based systems [84] [85]–[88]. Paragraphs below describe each of these categories.

2.6.1. Signature-Based IDS

Signature-based IDS monitor network traffic for probes and sweeps. Probes and sweeps are preconfigured, and predetermined attack patterns are known as signatures. These types of IDS are efficient for pre-known signatures but are inefficient when the IDS has no prior signature

knowledge [89] [90]. Therefore, the main drawback of signature-based systems is the requirement to continuously update the signature database. In addition, signature-based systems are also not efficient towards attacks with self-modifying behavior.

2.6.2. Anomaly-based

Anomaly-based Intrusion Detection Systems work on the principle of modeling the normal behavior of the system. They continuously monitor the data to create the baseline model, then look for deviations from that. The benefit of these systems from the signature ones would be the detection of malicious attacks even if those attacks were unknown to the system prior. These types of IDS have problems with high false alarm rates, the bias in the data, and high computational requirements [86], [91]–[94]. The IDS proposed in Chapter V falls under this category.

2.6.3. IDS State-of-the-art

When researching the automotive intrusion detection systems, there are several papers that have designed, built, and validated vehicle intrusion detection systems in a variety of approaches. As part of the literature, over 30 of these papers were considered to evaluate the state-of-the-art. On October 2018, the highly rated Journal of Ad Hoc Networks published a survey paper from researchers in the United Kingdom and Greece providing a great summary of these methods [95]. [95] has provided an overview of IDS in many environments ranging from aircraft, land vehicles, onboard systems, VANETs, etc. IDS is evaluated in this dissertation as state-of-the-art are the ones focused in VANETs. Table 2.7 gives a summary of these IDSs. All of the IDS given in this table are models with a TRL maturity level of 3. TRL 3 is described as “IDS have been evaluated the inaccurate simulation of vehicle states and attack mechanisms, possibly using data

from real vehicles" according to [95]. Below is a description of the elements provided in Table 2.7.

- IDS Reference: Reference of the IDS considered
- Type: Whether the IDS is a signature based or anomaly based as explained previously.
- Description: one sentence summary of the IDS.
- Evaluation: Whether an analytical, simulation or experimental evaluation approach was used for the references IDS.
- Technique: What technique does the IDS use.
- Performance: performance results on the IDS

Table 2.7: Vehicle Network IDS Based on [95] and Individual Papers

IDS	Type	Description	Evaluation	Technique	Performance
[96]	Anomaly	Use SVM and NN to detect grey hole and rushing attacks.	Simulation	SVM, feedforward NN	FP:1.21% FN: 0.23%
[97]	Anomaly	Extension of [96] with sensor data and different algorithms.	Simulation	k-nearest neighbor, FFNN, SVM	FN:0.12% FP: 0.22%
[98]	Anomaly	Represent anomalous and normal behavior using entropy and detect outliers using K-means clustering.	Simulation	k-means clustering	varies
[99]	Anomaly	Detect threats based on data collected in a collaborative fashion (speed, flow, density, location).	Simulation	Statistical t-test	FP: 2%
[100]	Signature and Anomaly	Trust-based mechanism geared towards denial of service attacks.	Simulation	Rule-based system	Accuracy 88%
[101]	Signature	Detect DoS based on # of TCP SYN that has not been acknowledged within a certain time.	Simulation	Rule-based	FP: 4-25%
[31]	Anomaly	Using Deep Learning to detect CAN message intrusions.	Simulation	Deep Learning	Acc: 99.91%-99.97% FP: 0.018-0.09%

Chapter 3. Threat Analysis

The primary purpose of designing V2X connectivity is to improve road safety, decrease the number of accident fatalities and increase traffic efficiency. However, the desired functionality could be impacted by several threats, and attacks as this technology heavily rely on wireless communications. The main V2X threats are more likely to affect the following properties: availability, authenticity, confidentiality, and integrity. This section explores in detail the main threats and attacks that affect V2X systems. Threats below are primarily based out of the analysis done in [18], [102], [103], and [104], according to the STRIDE model. STRIDE is a threat modeling developed from Microsoft. Each of the STRIDE categories will be analyzed as follow: Spoofing in section 3.1, Tampering in 3.2, Repudiation in 3.3, Information Disclosure in 3.4, Denial of Service in 3.5, and Elevation of Privilege in 3.6 [27].

3.1. Spoofing

Spoofing attacks attempt to spoof the authentication layer. Authentication layer is used to protect legitimate nodes from rogue insiders and outsiders. A good authentication mechanism generally prevents against several attacks, including black holes, spoofing GPS signals and replay attacks. The objective of this mechanism is that the resources and services should be accessed only by authenticated users.

An example of spoofing type of attacks are masquerading attacks. In order to impact the decision of a surrounding vehicle, this attack uses a valid entity known as a mask. This may appear as a legitimate node, and it is hidden. It tries to fabricate false messages and sends them to

surrounding vehicles in order to impact the vehicle's decisions. A malicious vehicle node can try to act as an emergency vehicle, and thus deceive other vehicles by broadcasting false basic safety messages into the V2X network, Other surrounding vehicles are deceived into believing that another vehicle is responsible for this attack [67].

When malicious nodes fail or refuse to forward messages, then they create a black hole attack [16]. The black hole attack can be injected into any ad hoc network and is a common attack against the authentication mechanism. Black hole attack means that a legitimate vehicle never receives messages because of the malicious vehicle which pretends to be part of the network. The malicious vehicle is not a legitimate node in the network. As a result, legitimate vehicles become vulnerable to such attacks from vehicles. These attacks are usually performed by insider actors in the network.

Another spoofing attack is GPS spoofing. In vehicular systems, GPS position information is an important variable for the vehicle and should be accurate. This information usually comes from Global Navigation Satellite Systems (GNSS). In spoofing attacks, a GPS satellite simulator is used to generate radio signals or messages that overwrite the signals from the accurate GPS satellite. This way, an attacker can spoof the vehicle to receive and process a different location than the one that they actually are. This can cause serious consequences for a car.

3.2. Tampering

Tampering attacks attempt to modify or inject malicious code or messages in the execution of the program. This has the potential to disrupt the operations of the vehicular network, OBUs, and RSUs because they receive periodic updates. As an example, we can look at [103], which describes a misconfiguration attack using country string field.

Another example of tampering attacks is when they fabricate and broadcast false basic safety messages (BSM). This is often done to deceive other cars and get other vehicles to behave in a certain manner. In addition to broadcast tempering, an attacker can also tamper transaction messages in flight. Tampering attacks are categorized as active attacks [105].

3.3. Repudiation

This attack happens when a vehicle refuses to accept the message causing the sender node to resend the message. Usually, this happens when the receiver does not verify the sender's authenticity or freshness. Due to the broadcast nature of DSRC, these attacks are not feasible to happen in this environment, that is why they are out of scope for this dissertation.

3.4. Information Disclosure

Information Disclosure attacks attempt to violate the confidentiality of messages. These attacks are often used to track or record certain confidential information and have privacy consequences. A common example of these is an eavesdropping attack. These attacks only impact one vehicle and attempt to collect user or other information about that vehicle, such as payment information, identity information, etc.

Another common example related to information disclosure is when attackers try to exploit vehicle tracking information. In general, an OBU sends out a BSM to inform other surrounding vehicles for traffic or safety situations. This message is signed with the OBU's certificate and other identifiers. If the attacker is able to track this piece of information across time, then it is able to track vehicle location.

3.5. Denial of Service

In a DoS scenario, the attacker attempts to bring down or overload the communication medium either by jamming signals at the physical layer thereby causing channel jamming or by flooding the nodes, so the vehicle nodes are prevented from accessing the network. The main purpose of the attacker in a denial of service attack is to prevent legitimate vehicles from accessing the V2X network and exchanging messages with other vehicles. A DoS intruder may attack either the individual vehicles (OBUs) or network units, i.e., RSUs[103].

An implementation of Denial of Service attack is flooding. These attacks flood the network with many false messages generated by malicious vehicle nodes. This causes the OBUs and RSUs to be flooded and unable to communicate with each other over the V2X channel. As a result of this attack, important basic safety messages are lost, and collision or other warnings are not delivered by the legitimate vehicle nodes. Spamming attacks are another type of DoS attack, and they occur when an intruder sends a series of messages to simply consume the network resources. The control of this attack is difficult in V2X as there is no centralized infrastructure.

Lastly, jamming attacks disrupt the communication channel at the physical layer by injecting noisy signals in order to halt message transmission delivery. As a result, the communication channel goes down, and the vehicles are unable to communicate with each other or infrastructure services. Jamming is also used to hide the identity of the attacker. [18]

3.6. Elevation of Privilege

This attack happens when messages attempt to obtain higher privileges. For example, fake high priority messages which attempt to flash malicious software would consist of this type of attack. Because this dissertation looks at BSM messages, the elevation of privilege attacks has

similar properties to the tempering attacks. For this purpose, this type of attack is not injected or categorized separately from the tempering attacks on the dataset used in Chapter V.

Table 3.1: STRIDE Threat Summary

Threat Type	Description	Property Targeted	Discoverable in BSM
Spoofing	Fake identity	Authentication	Yes
Tempering	Temper data	Integrity	Yes
Repudiation	Refuse to have done an action	Non-repudiation	No
Information Disclosure	Eavesdrop information	Confidentiality	Yes
Denial of Service	Overload the network with many messages	Availability	Yes
Elevation of Privilege	Unauthorized actions	Authorization	No

Table 3.1 gives a summary of the STRIDE threats.

Chapter 4. Assess: Fuzzy-based Threat Assessment Model (FTAM) [1]

The second stage of this solution is to perform a threat assessment for automotive cybersecurity threats. There are many models for threat assessment and characterizations which are used in automotive and other applications such as EVITA[13], NHTSA[12], HEAVENS[106], OCTAVE [107] [108] and others as references in [12] [66] [109] [110] [111] [112] [113] [114]. Many such models provide a threat assessment and are key to organizations attempting to define the severity of a threat. All these available threats or risk assessments are done in a manual and linear fashion, using table look-ups, or scoring mechanisms. They are often assessed based on users' perspectives using linguistic variables. Due to the ambiguous nature of the existing process, these threat assessment methods often lead to inefficient scoring and inaccurate assessments. The hypothesis in this dissertation is that the use of Fuzzy Logic improves the efficiency and accuracy of the threat assessment process.

This research is the first research, to our knowledge, that uses Fuzzy Logic to build a Fuzzy-Based Threat Assessment Model (FTAM). EVITA, NHTSA, HEAVENS, and OCTAVE were used to benchmark and drive the design of the new model. The research methodology attempted to close gaps and eliminate or mitigate drawbacks identified from the literature survey for these models. Although a new model is proposed, all the threat characterization levels are based on established models or references frameworks. From a Fuzzy perspective, a multistage methodology proposed from [28] is used. Methodologies from Mamdani [29] and Sugeno [30]

[1] This chapter was submitted for publication at the *Journal of Applied Intelligence*

were considered for use in the proposed method. V2X STRIDE threats as described in Chapter III were used in the design of FTAM [27].

The organization of this Chapter is as follow: Section 4.1 will discuss the Threat Assessment Models used to design FTAM, section 4.2 will explain a Fuzzy multistage architecture. Section 4.3 will introduce FTAM and sections 4.4 through 4.7 will provide the details of FTAM. The last section in this chapter (4.8), will give details on the integration and implementation of FTAM.

4.1. Threat Assessment Models in Automotive Cybersecurity

Threat assessment is the process of identifying and characterizing a cybersecurity threat by evaluating and assessing its properties. This is a tool used by many organizations around the globe in order to evaluate their systems and perform a risk analysis against common threats. This process has been studied and researched from multiple angles. Currently, there are a variety of threat models, and they can evaluate and characterize various threats. SAE J3061 guidelines provide a set of principles for automotive cybersecurity which includes threat identification, assessment, and analysis [112]. This standard defines threat assessment as “an analysis technique that is applied in the concept phase to help identify potential threats to a feature and to assess the risk associated with the identified threats” [115]. J3061 also does not specify a certain threat model as a standard, but rather leaves this choice to the individual organizations and allows them to determine risk levels. Paragraphs below will analyze the major threat assessment models and provide some of their characteristics, advantages, and drawbacks.

4.1.1. EVITA

EVITA is a European funded research project to design an automotive architecture for in-vehicle networks. The objective of the EVITA project was to protect vehicle components and sensitive data from cybersecurity threats. The project started in July 2008 and concluded in December 2011. Among many of its deliverables, they also established a threat assessment model [13]. EVITA threat assessment combines attack severity, attack potential, and (un)controllability.

To assess the attack potential, EVITA looks at the elapsed time, the expertise of the attacker, the knowledge that is available for the system, the window of opportunity available to perform the attack, and the specialty of the equipment used from the attacker. The severity of the attack is determined from the impact it has on safety, privacy, financial loss, and operational disruption. The third component measures how you can avoid (or control) the situation in an attack scenario[13]. EVITA is also known as a threat and operability analysis (THROP) because it focuses on the functional perspective for a feature when considering threats. This model is designed for the concept phase, and it requires a considerable amount of effort during the attack classification. EVITA also has issues with determining the accuracy for attack potentials due to using subjective operations. On the other hand, EVITA performs well in terms of threat assessment because it makes a clear categorization between different aspects of consequences in the severity step. EVITA characteristics are considered and used in FTAM. Table 4.1 provides elements of the EVITA model.

Table 4.1: EVITA Threat Elements

Severity	Probability	Controllability
Safety (S0-S4)	Elapsed Time	
Privacy (S0-S4)	Window of Opportunity	
Financial Loss (S0-S4)	Expertise of Attacker	
Operational Disruption (S0-S4)	Knowledge of the System	
	Specialized Equipment	

Figure 4.1 provides an overview of how EVITA elements come together to determine a risk level.

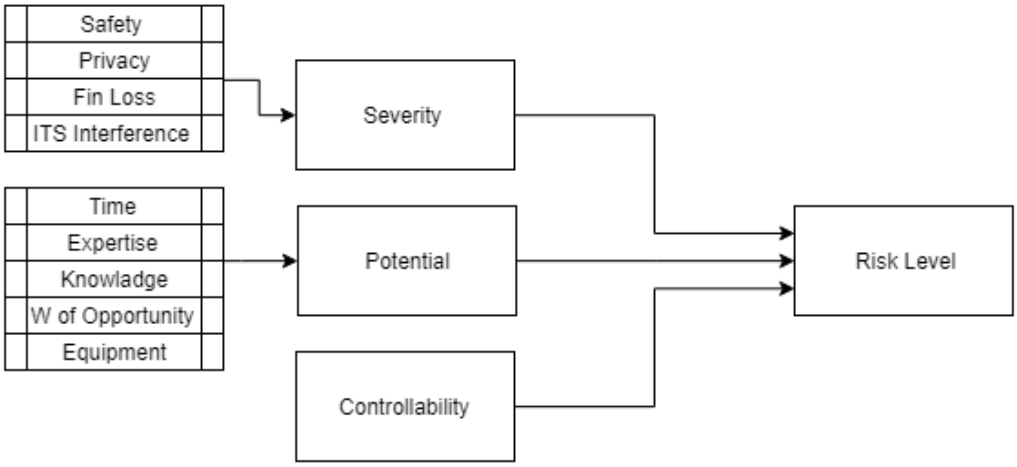

Figure 4.1: EVITA Architecture

4.1.2. NHTSA

National Highway Traffic Safety Administration (NHTSA) has defined a composite threat assessment model for identifying and classifying potential threats in the automotive space [12]. This model is also referred as the “Threat Matrix” due to using a matrix nature when assessing cybersecurity threats. NHTSA model attempts to identify potential threats, provide information regarding existing attack surfaces and defenses, and categorize or assess the threats so organizations can develop attack mitigations.

NHTSA's model shares common elements and is based on the following: STRIDE [27] , Trike [116] and ASF [117]. This composite threat model takes the following steps during its process:

- Identify applications or systems which are critical for the vehicle's operations. If these applications or systems are attacked, then they would have an impact on the safety of the vehicle.
- After critical systems are identified, then the model would seek to decompose them and understand the interconnections between their composites. A diagram of the interconnections and a data flow diagram is created during this step.
- Do threat identification. This is a continuous process that takes into consideration research and other cross-functional areas.
- Threat Analysis is based on STRIDE and is a step which identified and categorizes a threat.

Table 4.2 is used during this process.

Table 4.2: NHTSA Threat Matrix [12]

Matrix Category	Category Description	Options
ID Number	Identification number for the attack	
Attacked Safety and Non- Safety Zone Groups /Attack Support Zone Groups	Groups of various like categories of components and systems that are targeted by the attack or that are used to support the attack	Communications: o Internal communications paths (e.g., CAN, FlexRay, IDb-1394, MOST) • Vehicle Operations: o Powertrain - Engine control, hybrid drive systems, transmission, misc. power train sensors (e.g., torque converter lockup) o Chassis and Safety - Brake control, steering, environmental sensors, airbag sensors, tire pressure sensors, misc. chassis sensors (e.g., steering angle) o Body Electronics - Instruments, door modules (e.g., remote locks, light control, seat control) • Comfort Systems:

		o Climate control, air vent positions, remote start • Infotainment: o Audio, display/video, navigation, embedded telephonic communications • External interfaces: o GPS, diagnostic ports, USB, Bluetooth, key fob
Attacked Zone Safety Related	Whether or not the attacked zone contains safety-related functions	• Yes • No
Component/System	The component or system that is under attack	E.g., the electronic braking system as opposed to an electronic brake actuator
Vulnerability That Could Be Exploited	Protocols/applications that could be used/corrupted in order to achieve the outcome of the attack	E.g., lack of firewalls, easy diagnostic access
Attack Vector	Entry point of the potential attack	E.g., OBD-II input, USB port, Bluetooth, GPS, audio system, etc.
Attack Method	The transport mechanism that could be used to launch the attack	E.g., OBD-II input, USB port, Bluetooth, GPS, audio system, etc.
Attack Type	Type of attack that could be used	• Spoofing identity • Tampering with data • Repudiation • Information disclosure • Denial of service • Elevation of privilege
Attack Name/Scenario	A compressed narrative of the potential attack derived from the use cases	The narrative contains: • Name of the attack (title) • Who the attacker may be • What the targeted component/system may be • How the attacker may gain access to the component/system • How the attack may be launched
Resources Required	Resources that may be needed to carry out the attack	E.g., hardware, software, access to vehicle (physical or remote), skill level
Casualty Severity	Projected outcome severity due to the potential attack	• High: High likelihood of severe injury or loss of life; loss of control of vehicle • Medium: Potential to cause injury; experienced operator may be able to maintain control of vehicle • Low: No injury; no loss of vehicle control during the attack; attack motive was for theft, nuisance, or publicity only
Financial Severity	The outcome severity in terms of direct or indirect financial loss to the owner, OEM,	• High: Could cause major financial loss to vehicle, business, or product reputation • Medium: Potential to cause moderate financial loss to vehicle, business, or product reputation • Low: Minimal loss to vehicle, business, or product reputation; attack motive was for nuisance or publicity only
Trip Phase	The vehicle's movement category at the instance of the potential attack	One or more of the following may be used: • Parked- not moving, engine shut off • Idling- not moving, engine running • Stop-and-go- i.e., heavy traffic • City driving- typical city limits speed • Urban driving- typical urban space speeds

		• Highway driving- typical highway speeds • Any- Not speed-dependent
Loss of Privacy	Whether or not items such as onboard address books, vehicle location, or passwords may have been compromised and shared with un-trusted parties	• Yes • No
Outcome	Ramifications of a successful potential attack, assuming no mitigations were bypassed	
Operator Override	What an average operator may be able to do to override or avoid the ramifications of the potential attack	• High: Extremely complex code; may attack multiple components/systems; may use zero-day exploits; may have multiple triggers; hard to detect and remove; may be persistent (launching attack payload more than once), and may erase itself after the attack is executed • Medium: Moderately complex code; may contain remote trigger; may be persistent; may use zero-day exploits • Low: Non-persistent; easy to detect; makes use of potential vulnerabilities
Difficulty of Implementation	How difficult is it to implement the potential attack	• High: Extremely complex to implement; may require prolonged and advanced physical access to the vehicle; may need specialized tools and/or knowledge to launch. • Medium: Moderately complex to implement; may require some physical access to vehicle; may need some and/or specialized knowledge to launch • Low: Easy to implement; requires minimal/no physical access to vehicle; requires no specialized knowledge to launch
Likelihood	The likelihood of a potential attack to be carried out	• High: Well-known attack; very easy to perform; canned malware available for the attack • Medium: Some knowledge of system needed; access to entry point more difficult; some custom code needed • Low: Expert knowledge of component/system required; entry point difficult to access/unexpected; high level of custom coding involved

As can be observed, the table is large and confusing in some respects. This threat matrix is not recommended in SAE J3061 [112].

4.1.3. HEAVENS

HEAVENS leverages Microsoft's STRIDE model to perform threat assessment [106]. The objective of this project is to define security requirements similar to ISO26262 functional safety requirements. HEAVENS assesses threats according to several element. The three elements that HEAVENS will look at when doing risk assessment are: Threat Level (TL), Impact Level (IL) and Security Level (SL). As seen from Table 4.3, TL focuses on determining the likelihood of the threat and takes into consideration four elements. Impact level assesses the impact of the threat in the following four categories: Safety, Financial, Privacy, and Operational legislation. TL and IL are combined to derive the Security Level (SL), which is the output of the HEAVENS assessment. This model has an advantage over other assessment models because it benefits from the structured and systematic STRIDE approach, a property which is used in FTAM. From the other side, HEAVENS has a disadvantage because it requires an extensive amount of work to analyze and determine the safety level for each individual threat. This usually results in inconsistent assessment for different threats.

Table 4.3: HEAVENS

Security Level	**Threat Level**	**Impact Level**
Combination of TL and IL	Expertise of the attacker (0-3)	Safety
	Window of opportunity (0-3)	Financial
	Equipment (0-3)	Privacy
	Knowledge of the system (0-3)	Operational
		Legislation

4.1.4. UM Risk Assessment Model

Researchers at the University of Michigan have also produced a risk assessment framework for automated driving applications. This dissertation refers to this framework as The UM risk assessment model [114]. This model looks at impact, attack potential, and motivation to determine a threat assessment vector. Figure 4.2 shows the elements of this model.

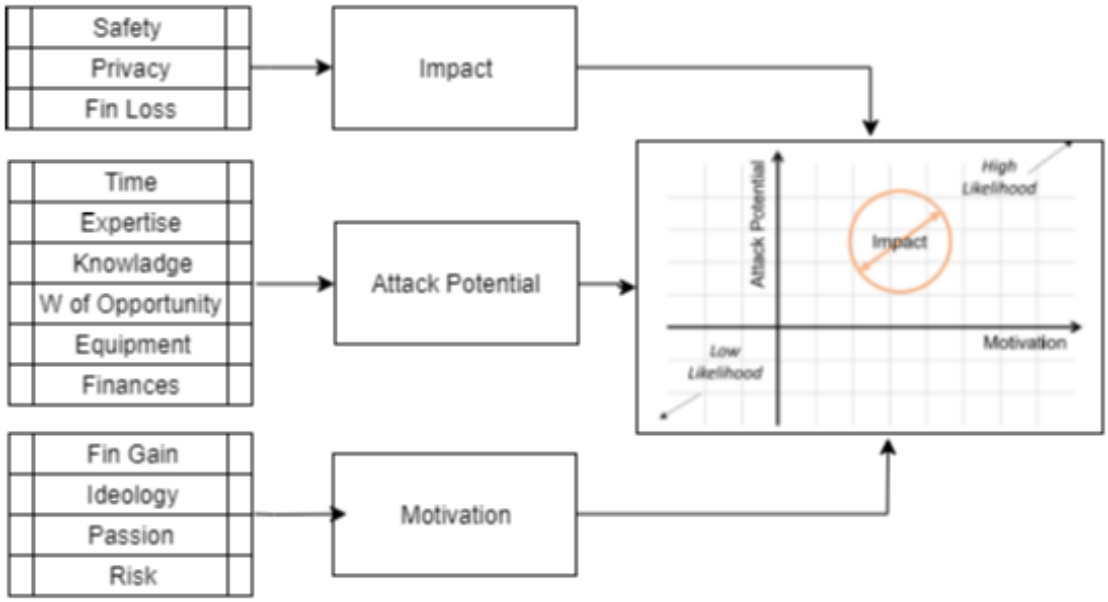

Figure 4.2: UM Risk Assessment Model

As it can be observed, The UM model is mainly based on EVITA but provides a vector for threat assessment rather than a level. This model is good at analyzing threats and improves on EVITA's assessment due to its focus on automated driving. The analysis is well defined, and threat characterization is in line with other models. The UM model adds additional elements such as passion, ideology, etc., to further evaluate the motivation of risk. But from the other side, this model is not able to efficiently quantify the threat level using a discrete method. The output of such a model is a vector represented in three dimensions.

4.1.5. OCTAVE

Operationally Critical Threat, Assess and Vulnerability Evaluation (OCTAVE) is another threat assessment model which focuses on organizational threats and practical issues. OCTAVE is different from the other threat assessment models because it focuses on the organizational risks and security practices followed by the organization and is not applicable to cyber-physical systems. OCTAVE has three phases. The first one is about the organizational view and takes in

considerations threats, assets, current practices, vulnerabilities, and security requirements. The second one looks at it from a technological view and considers the key technological components and vulnerabilities. The third phase looks at risks, protection strategies, and mitigation plans. This model is focused on the process rather than requirements or assessments. It is mostly used for information security risk assessments across the enterprise and is not applicable for cyber-physical systems such as automotive systems. The reason why this is considered in this study is due to its layered approach, flexibility to analyze, and in-depth capabilities [102] [103].

Table 4.4 gives an overview of the previously described methods, along with their major characterizations, advantages, and disadvantages.

Table 4.4: Threat Assessment Models

Model	Design Phase Applicability	Major Characterizations	Advantages	Disadvantages
EVITA	Concept	Severity, Potential, Controllability	widely used, automotive-focused,	not conforming ISO26262, in-balance between safety & non-safety threats, accuracy of measures
NHTSA	System	List format	FMEA focused,	Spreadsheet and text-based, not favorable in the concept phase,
HEAVENS	System	Threat Level, Impact Level, Security Level	Uses STRIDE approach,	Extensive effort for analysis, ambiguous
UM*	Concept	Impact, Attack Potential, Motivation	More comprehensive coverage	Produces a three-dimensional vector, not a level

4.2. Multistage Fuzzy Architecture

Fuzzy Logic helps with uncertainty and has many properties which help in threat assessment. In a threat assessment model such EVITA, for example, if a Fuzzy inference system is used, there are 11 threat characteristics, each of them with about 5 levels of assessment. If a

classic Fuzzy inference system is applied in this model, then there are about 115 = 161,051 Fuzzy rules to be defined. Such a large number of rules is simply impossible and not efficient to design or maintain.

Due to this problem, a Fuzzy multistage architecture proposed by Shaout and Trivedi in [28] is considered. In this paper, a performance appraisal system was designed using Fuzzy Logic. Using the traditional Fuzzy approach, their system would look like Figure 4.3:

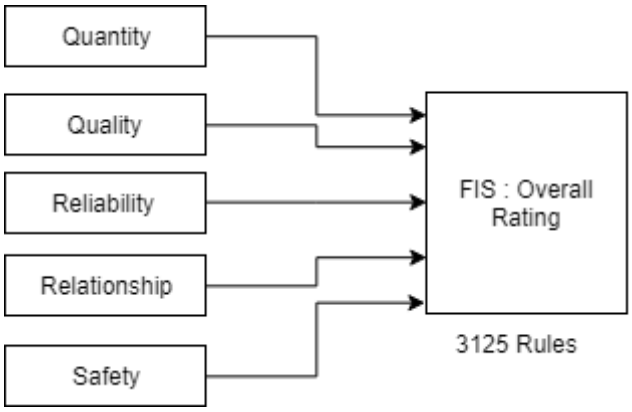

Figure 4.3: Traditional Fuzzy Approach in Performance Appraisal System [28]

This system has five characteristics, and each of them has five linguistic variables. Therefore, this would result in 5^5 = 3,125 rules. To cope with this problem and reduce the number of rules, a multistage architecture was proposed. Figure 4.4 gives an overview of this architecture.

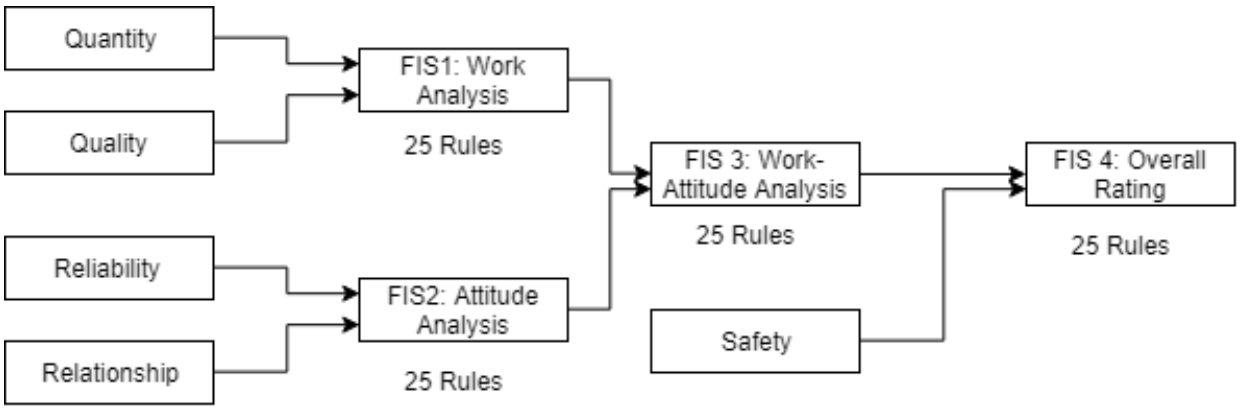

Figure 4.4: Multistage Fuzzy Architecture Proposed in [28]

This architecture is able to group the relevant characteristics and create a multistage Fuzzy inference system. The referenced paper designs four different Fuzzy Inference Systems (FIS) and can reduce the number of Fuzzy rules from 3,125 to only 100 rules. The simulation of their methodology results proves to be valuable in reducing the number of rules and creating better relationships among Fuzzy elements. This method was used when designing the proposed threat assessment model.

4.3. Fuzzy-Based Threat Assessment Model (FTAM)

The Fuzzy-Based Threat Assessment Model is a new and innovative approach to threat assessment. Based on literature survey research, no other models use Fuzzy Logic for threat assessment. The objective of this dissertation is to produce a threat assessment model focused on vehicle communication systems and aimed to be a functional model. For that reason, the model proposed is based on other common threat assessment models. The following are some of the reasons why a new model is needed in the first place:

- After a careful analysis of the literature survey, disadvantages and drawbacks are discovered for each of the existing threat models. The proposed threat model attempts to cope with some of those disadvantages, especially the ones related to vehicle communication. FTAM advantages are given after its analysis.
- SAE J3061 as one of the defined guidelines for cybersecurity allows and encourages individual organizations to select their threat assessment of choice. This standard leaves room for new and additional threat models according to specific applications.
- Threat modeling is a subjective process dependent on linguistic categorizations but often delivers a discrete level of risk. Dealing with subjective notations is difficult due to a lack of mathematical quantifications. Previously, it was explained that Fuzzy Logic performs

well in those situations. FTAM is the first model according to our research that leverages Fuzzy Logic in order to assess risks with subjective classifications.

- SAE J3061 also leaves it up to the individual organizations to determine the acceptance of their risk levels. The proposed model is flexible enough to allow organizations to perform their threat assessment and determine risk acceptance levels.

FTAM attempts to assess threats and characterizes them based on the following three categories.

- Attack: Attack (A) captures the impact or the severity that an attack could do on Privacy (A_p), Safety (A_s), and Financials (A_f), It is measured by the following:

$$A = \sum_{i \in p,s,f} w * A_i \quad (1)$$

Where w is the weight for each component and $0 \leq w \leq 1$.

- Attacker: Attacker (T) assesses the capabilities of the attacker and its motivation. It centers this assessment on the following elements: expertise (T_e), resources (T_r), and financial gain (T_g) . The following equation measures the assessment:

$$T = \sum_{i \in e,r,g} w * T_i \quad (2)$$

- Withstand Potential (P) measures the system potential to withstand an attack. It does so by looking at the system's ability to control (P_c) an attack after it happens and the difficulty (P_d) to accomplish an attack. Its respective equation is:

$$P = \sum_{i \in c,d} w * P_i \quad (3)$$

The high-level architecture of such threat modeling is shown in Figure 4.5:

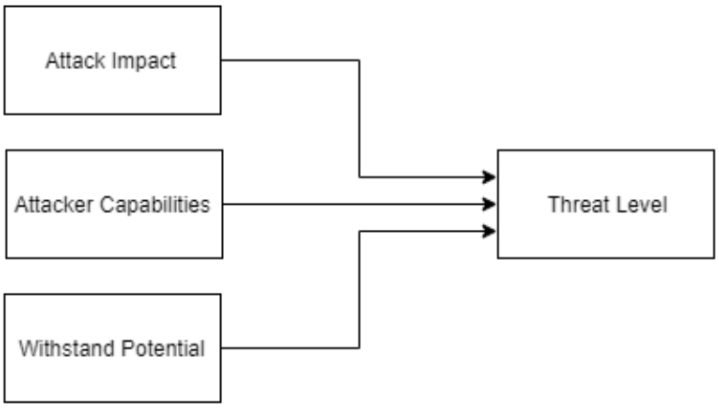

Figure 4.5: Fuzzy-based Threat Assessment Model (FTAM)

Equation 4 gives a conceptual mathematical view of the proposed system. FTAM is not a real-time system and its Threat Level (TL) is determined by adding Attack Severity with Attacker and subtracting Withstand Potential:

$$TL = \sum_{i \in p,s,f} w * Ai + \sum_{i \in e,r,g} w * Ti - \sum_{i \in c,d} w * Pi \qquad (4)$$

Sections below provide a careful analysis of FTAM components.

4.4. Attack Impact

The objective of the attack impact is to measure the severity of the impact when an attack occurs. As described previously, this element is measured in terms of Privacy, Safety, and Financial loss. If this is compared with the EVITA Architecture, it is noticed that it lacks ITS interference. This is done intentionally because V2X threats usually do not have interference with ITS. This element does align with the UM model's Severity element [114].

In terms of Fuzzy Logic, this branch is designed using two Fuzzy Inference Sets. The first one combines Privacy and Safety, while the second one adds Financial Loss. The block diagram for this branch is shown in Figure 4.6:

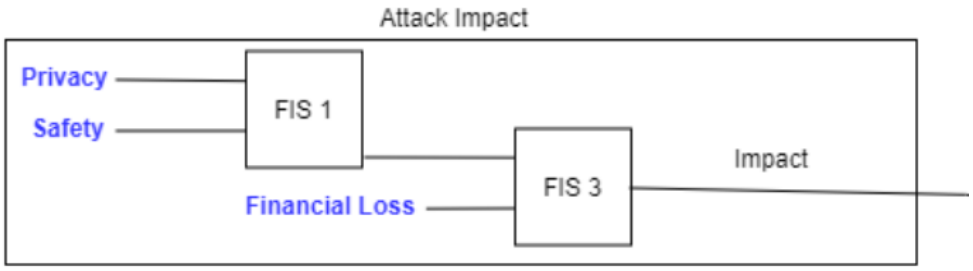

Figure 4.6: Attack Impact

4.4.1. FIS1: Privacy and Safety

The first Fuzzy Inference System (FIS) in this FTAM is the combination of Privacy and Safety. This is under the attack branch and attempts to measure the impact or the severity of a Threat. Figure 4.7: Fuzzy Inference System Model gives a generic view of how this FIS works.

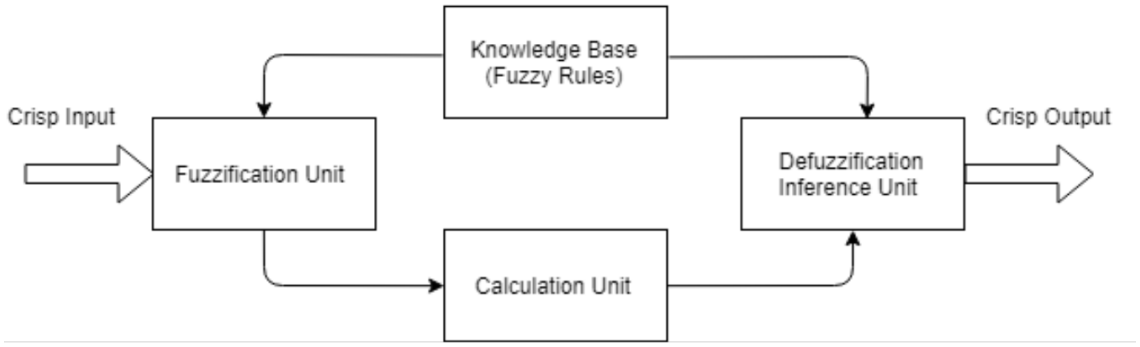

Figure 4.7: Fuzzy Inference System Model

The fuzzification unit takes the input and converts it into Fuzzy quantities. The knowledge base holds the relationship rules between the inputs and outputs. The calculation unit calculates or “fires up” the operations based on each rule.

FIS1 diagram for FTAM is shown in Figure 4.8: FTAM FIS1 and is comprised of Privacy and Safety. Levels for the Safety element come from ISO26262 and are defined as follow: None, Low, Medium, and High. Meanwhile, the Privacy element references the NHTSA model, which has assigned binary values: Yes, for an impact on privacy and No for no impact

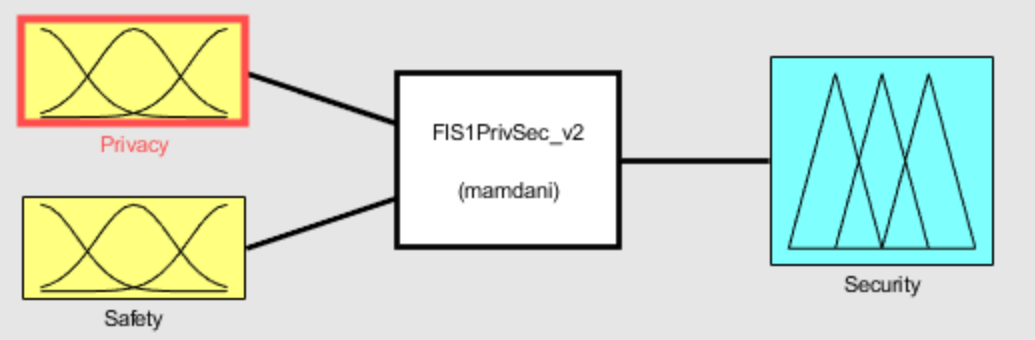

Figure 4.8: FTAM FIS1

The Safety Fuzzy linguistic input variable (FLIV) is characterized by a set of Fuzzy linguistic values. Their membership functions are represented by equation 5:

$$f(x;a,b,c) = \begin{cases} 0, x \leq a \\ \frac{x-a}{b-a}, a \leq x \leq b \\ \frac{c-x}{c-b}, b \leq x \leq c \\ 0, c \leq x \end{cases} \quad (5)$$

Where a and c locate the basis of the triangle and b is the peak. Figure 4.9 describes Safety element transitions.

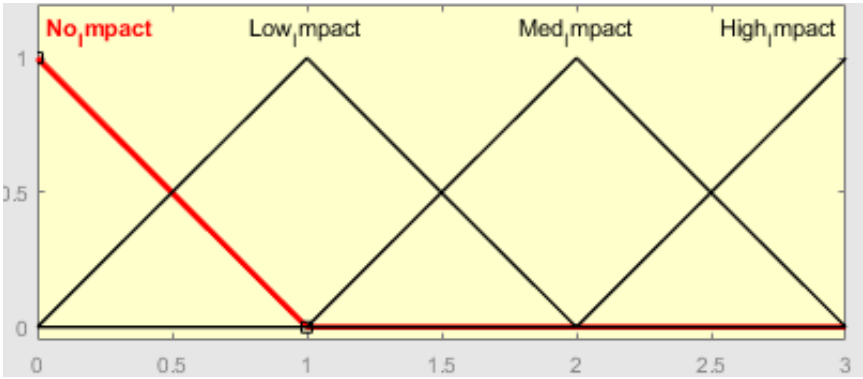

Figure 4.9: FIS1 Safety

Privacy FLIV is represented with two linguistic variables (Yes and No) and its membership functions are represented by equation 6:

$$f(x;a,b) = \begin{cases} 0, x \le a \\ 2\left(\frac{x-a}{b-a}\right)^2, a \le x \le \frac{a+b}{2} \\ 1-2\left(\frac{x-a}{b-a}\right)^2, \frac{a+b}{2} \le x \le b \\ 1, x \ge b \end{cases} \quad (6)$$

Equation 6 creates a spline-based curve with parameters a and b showing the extremes of the slope.

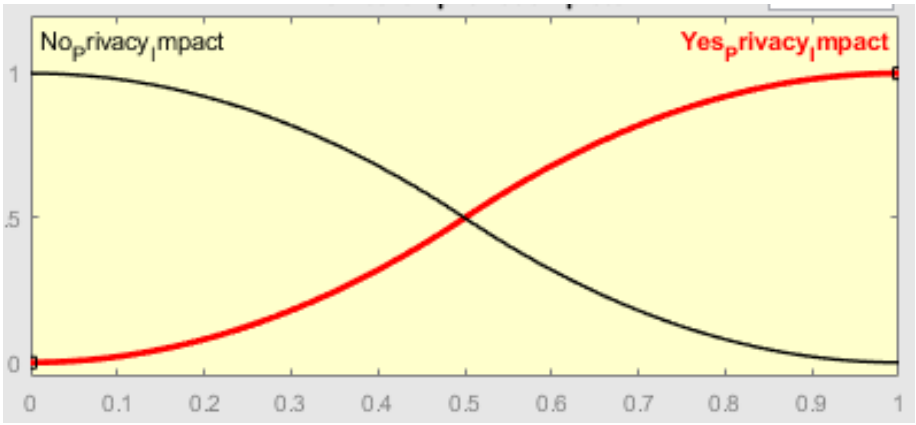

Figure 4.10: FIS1 Privacy

The output for FIS is a transitionary output and named security. Its respective values are also from ISO26262: No, Low, Medium, High. For the first (no) function, in order to maintain a no-impact state, a trapezoidal shaped membership function is used. This function is represented by equation 7.

$$f(x;a,b,c,d) = \begin{cases} 0, x \le a \\ \left(\frac{x-a}{b-a}\right), a \le x \le b \\ 1, b \le x \le c \\ \left(\frac{d-x}{d-c}\right), c \le x \le d \\ 0, d \le x \end{cases} \quad (7)$$

For all the other membership functions in FIS1 output, a Gaussian-shaped function centered around the linguistic value and with σ = 0.5 is used. The reason Gaussian is used in this FIS and others throughout the paper is due not only to its ability to provide a smooth output and be non-

zero at all points, but also to the fact that it can be formed from univariate sets. Its representative equation is as follows:

$$f(x;\ \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \qquad (8)$$

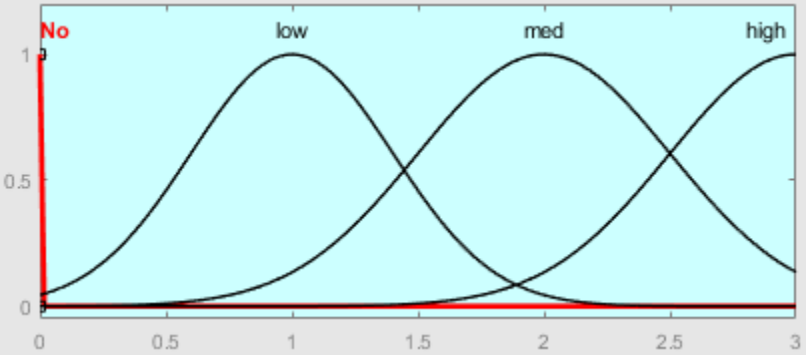

Figure 4.11: FIS1 Security

Rules for this FIS1 are shown in Figure 4.12. It is important to mention that the rules generated throughout this dissertation were through assessing and benchmarking STRIDE attacks with EVITA and HEAVENS models. In addition, FIS1 and other subsequent FISs account for all of the rules for each variable. It is possible that, if this model is commercialized or used for other applications, these rules can be changed based on further testing, validation or scope of implementation. Due to the nature of the individual assessments, all the rules are defined since it is possible that they can be triggered in different scenarios:

1. (Privacy==No_Privacy_Impact) & (Safety==No_Impact) => (Security=No) (1)
2. (Privacy==No_Privacy_Impact) & (Safety==Low_Impact) => (Security=low) (1)
3. (Privacy==No_Privacy_Impact) & (Safety==Med_Impact) => (Security=med) (1)
4. (Privacy==Yes_Privacy_Impact) & (Safety==No_Impact) => (Security=low) (1)
5. (Privacy==Yes_Privacy_Impact) & (Safety==Low_Impact) => (Security=med) (1)
6. (Privacy==No_Privacy_Impact) & (Safety==High_Impact) => (Security=high) (1)
7. (Privacy==Yes_Privacy_Impact) & (Safety==Med_Impact) => (Security=high) (1)
8. (Privacy==Yes_Privacy_Impact) & (Safety==High_Impact) => (Security=high) (1)

Figure 4.12: FIS1 Rules

All the rules have a weight of 1 and establish the 3-dimensional surface shown in Figure 4.13:

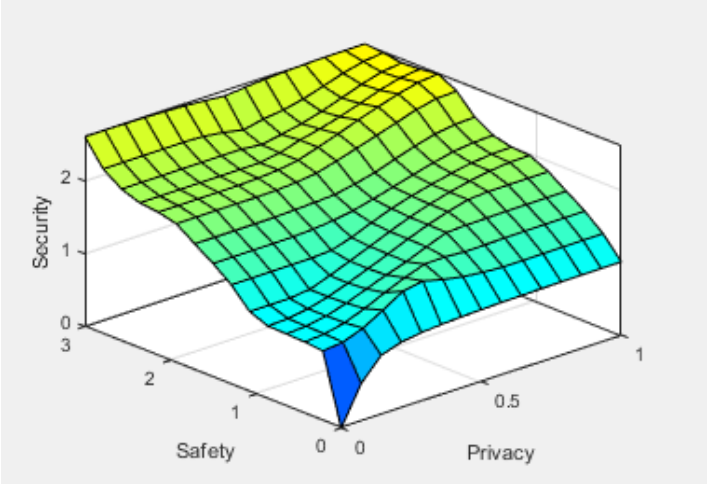

Figure 4.13: FIS1 Output

4.4.2. FIS3: Security and Financial Loss

The next Fuzzy Inference System (FIS) is the combination of previous output from Privacy and Safety with Financial Loss. Since this is still under the attack element, the purpose is to continue evaluating how severe is an attack if it happens. FIS1 evaluated the impact on privacy and safety, while this FIS adds financial loss as one of the elements. Until this point, FTAM is aligned with EVITA and the UM model from a functional perspective [13] [114].

FIS3 diagram is shown in Figure 4.14 . The Security input in this FIS is the same as the output of FIS1, while the Financial Loss is the additional input. The output of this FIS is attack severity.

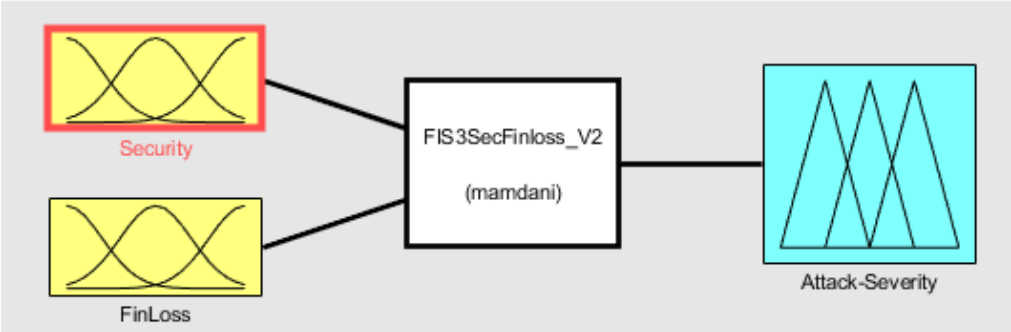

Figure 4.14: FTAM FIS3

Security input and its mathematical representation are already described in the previous section. While the financial loss is based on the German standard of the Federal Office for Information Security (BSI-Standard 100-4) [118]. There are essentially four levels for this input which are described as follows:

- No Impact means that there is no financial implication when such an attack occurs. These Fuzzy linguistic values are represented by membership functions with a trapezoidal graph. Trapezoidal was selected due to its ability to preserve the zero state where: a=0; b=0.

$$f(x;a,b) = \begin{cases} 1, x \leq a \\ 2\left(\frac{x-a}{b-a}\right)^2, a \leq x \leq \frac{a+b}{2} \\ 1-2\left(\frac{x-b}{b-a}\right)^2, \frac{a+b}{2} \leq x \leq b \\ 0, x \geq b \end{cases} \quad (9)$$

- Low value translates to a financial loss caused by the attack which is tolerable for the system. This and the rest of the membership functions under Financial loss are represented with a triangular equation for simplicity:

$$f(x;a,b) = \max(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0) \quad (10)$$

For LOW parameters are as follow: a=0, b=1, c=2.

- Medium means that a successful attack would result in substantial financial losses, but the system would still be able to stand. It is represented by a triangular graph with parameters: a = 1, b = 2, c = 3.
- Lastly, the “High” level says that an attack would result in a high financial loss with severe damage to the system. Its membership functions are represented by equation 10 with parameters: a = 2, b = 3, c =4.

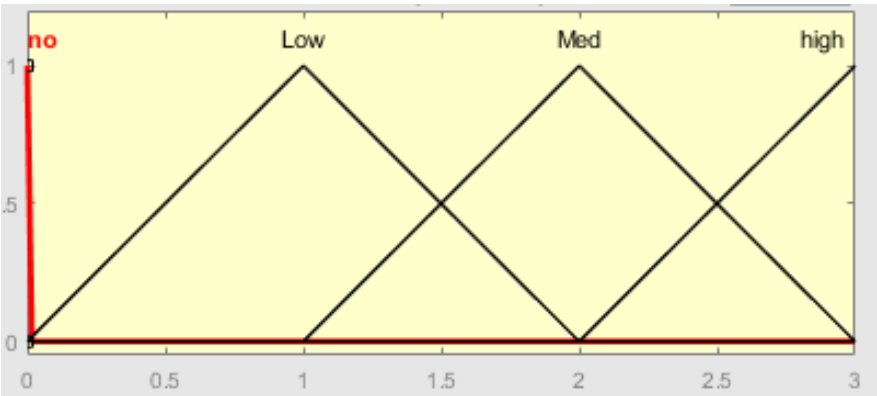

Figure 4.15: FIS3 Financial Loss

The output of FIS3 is Attack Severity, which is aimed at combined Financial Loss, Privacy and Safety impacts. EVITA, UM, NHTSA, HEAVENS, and others also include these or similar variables to measure attack impact (or severity). The output levels leverage a standard 0-4 scale, which is used in ISO26262, EVITA, and others. Its levels are as follows:

- No – translated to no injuries from a safety perspective; no unauthorized access to data from a privacy perspective and no financial loss.
- Low – means a relatively low impact of an attack occurs with light injuries from a safety perspective, with no Personally Identifiable Information (PII) being leaked and low financial losses.
- Medium means that an attack would result in a severe impact with severe injuries from a safety perspective, with leaks of PII and a moderate financial loss.
- High is the level where if an attack occurs would have significant damages, including life-threatening injuries from a safety perspective, privacy of data being impacted, and there being a significant financial loss for one or more vehicles.
- Very High would be the highest impact that an attack can cause, e.g., fatal injuries, breached data, and a heavy financial loss from multiple vehicles.

- All those levels are described with a Gaussian membership function. Gaussian was chosen due to its ability to maintain a non-zero state at all points. Equation 11 is used for the membership function as follow:

$$f(x;\ \sigma, c) = e^{\frac{-(x-c)^2}{2*0.3^2}} \qquad (11)$$

The output of FIS3 is shown in Figure 4.16: FIS3 Attack Severity

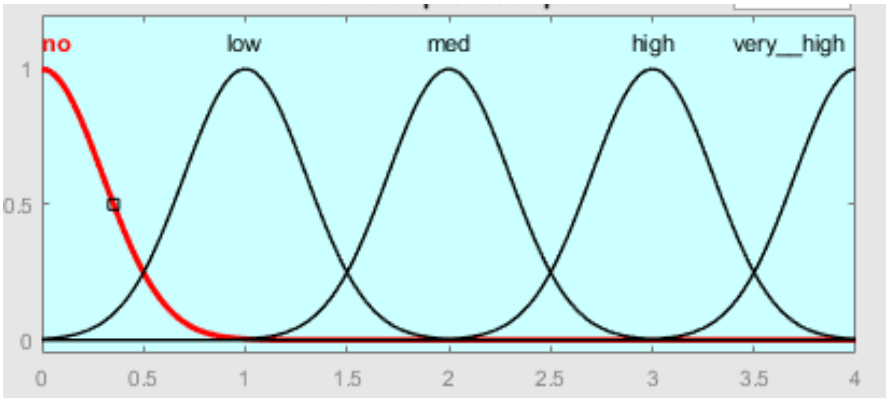

Figure 4.16: FIS3 Attack Severity

The representation of knowledge between input/outputs in this FIS is shown by the rules in Figure 4.17: FIS3 Rules. It is important to mention that these rules are generated by assessing and comparing STRIDE attacks in the following assessment methodologies: CVSS, OWASP, EVITA, and HEAVENS [119].

1. (Security==no) & (FinLoss==no) => (Attack-Severity=no) (1)
2. (Security==no) & (FinLoss==Low) => (Attack-Severity=low) (1)
3. (Security==no) & (FinLoss==Med) => (Attack-Severity=med) (1)
4. (Security==no) & (FinLoss==high) => (Attack-Severity=med) (1)
5. (Security==Low) & (FinLoss==Low) => (Attack-Severity=low) (1)
6. (Security==Low) & (FinLoss==no) => (Attack-Severity=low) (1)
7. (Security==Low) & (FinLoss==Med) => (Attack-Severity=med) (1)
8. (Security==Low) & (FinLoss==high) => (Attack-Severity=high) (1)
9. (Security==Medium) & (FinLoss==no) => (Attack-Severity=med) (1)
10. (Security==Medium) & (FinLoss==Low) => (Attack-Severity=med) (1)
11. (Security==Medium) & (FinLoss==Med) => (Attack-Severity=high) (1)
12. (Security==Medium) & (FinLoss==high) => (Attack-Severity=high) (1)
13. (Security==High) & (FinLoss==no) => (Attack-Severity=high) (1)
14. (Security==High) & (FinLoss==Low) => (Attack-Severity=high) (1)
15. (Security==High) & (FinLoss==Med) => (Attack-Severity=very__high) (1)
16. (Security==High) & (FinLoss==high) => (Attack-Severity=very__high) (1)

Figure 4.17: FIS3 Rules

These rules generate the surface shown in Figure 4.17. As stated previously, all rule combinations are accounted for, and they are generated based on assessment benchmarking.

As it can be observed, when there is no impact on security, privacy, or financial loss, then the attack severity is 0. When these impacts increase, then the attack severity increases proportionally according to the knowledge rules:

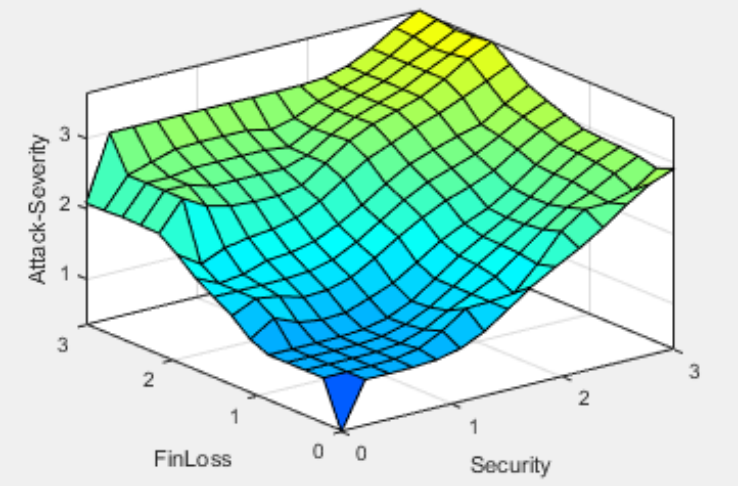

Figure 4.18: FIS3 Surface

4.5. Attacker

The next element to evaluate is the attacker capabilities. Similar to other Threat Assessment models, this is considered the attack potential. The reference models use a variety of characteristics to assess this element such as: Time required to perform the attack, expertise of the attacker, knowledge of the system, window of opportunity, equipment's available to use and financial resources available. In FTAM, these are categorized into the following three elements:

- Expertise – This is used in assessment from all of the Threat Models (Common Criteria [120], TVRA [121], OWASP , EVITA[13], SECTRA, and HEAVENS [106]). This parameter evaluates the level of knowledge that the attacker has in order to perform the

attack. Therefore, in FTAM expertise also includes the element “knowledge of the system” found in other systems.

- Resources is another category which is found in different names across all the other threat models. It is the parameter which evaluates the resources available to an attacker in order to perform the attack. This is sometimes referred to as “equipment” as well, but in FTAM it also includes financial or other resources available to the attacker.
- “Financial and other gains” helps FTAM to determine the motivation that an attacker has in order to perform the attack. Other threat models include this element in one shape or the other. Note that in FTAM, the word “other” is included which captures other gains that an attacker might have (i.e., ideology, etc.)

The model used for Fuzzy is shown in Figure 4.19.

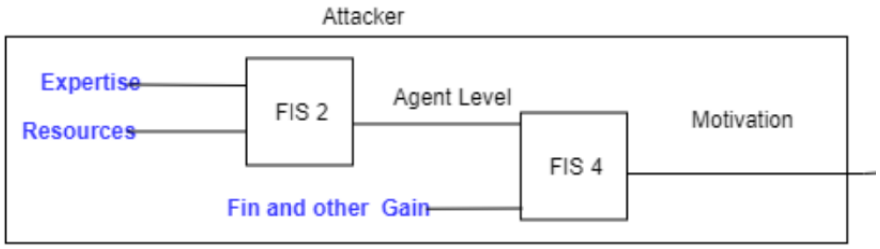

Figure 4.19: FTAM Attacker

4.5.1. FIS2: Expertise and Resources

The first Fuzzy Inference System (FIS) in this branch of FTAM is the evaluation of expertise and resources available. As described previously, those elements provide a good assessment of the agent (attacker) level. The reasoning behind this FIS follows: If an attacker is

an expert, has the knowledge and the resources to perform an attack, then he is considered high risk. FIS2 model uses the "Mamdani" method and is shown in Figure 4.20: FIS2.

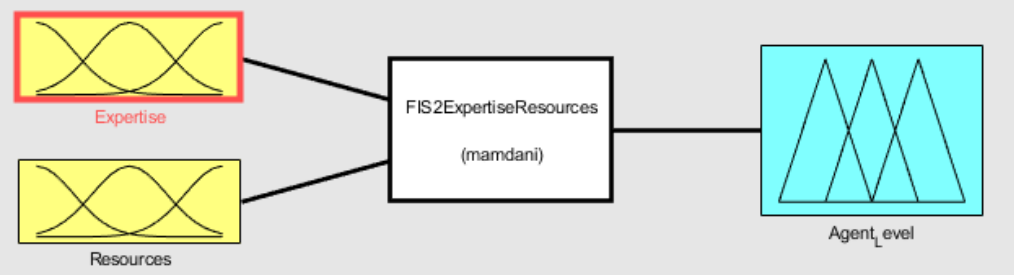

Figure 4.20: FIS2

The expertise element has four membership functions and is modeled primarily after the EVITA model:

- Normal – this level means that the attacker does not have specialized expertise to perform the attack. They are able to follow simple instructors in order to mount a threat, but they are unable to succeed when these instructions are not clear enough. To represent this function, a pi-shaped equation is used. This function is generally used when the linguistic variables are more discrete in nature and can be clearly separated from each other. In this case, the attacker is clearly categorized into one of those areas, and that is why such a function is used. This curve is used for other levels as well and is described in equation 12. For normal level, a = b = 0 and c = d = 1.

$$f(x;a,b,c,d) = \begin{cases} 0, x \le a \\ 2\left(\frac{x-b}{b-a}\right)^2, a \le x \le \frac{a+b}{2} \\ 1-2\left(\frac{x-b}{b-a}\right)^2, \frac{a+b}{2} \le x \le b \\ 1,\ b \le x \le c \\ 1-2\left(\frac{x-c}{d-c}\right)^2, c \le x \le \frac{c+d}{2} \\ 2\left(\frac{x-c}{d-c}\right)^2, \frac{c+d}{2} \le x \le d \\ 0, x \ge d \end{cases} \quad (12)$$

For the normal MF we have a=b=0 and c=d=1.

- Proficient – at this level, the attacker has some general knowledge about the security domain. This attacker is proficient in generally known attacks and is able to improvise many of them. This level also uses equation 12 and has a = b = 1 and c = d =2
- Expert is someone who has expert knowledge in the security domain. They are usually familiar with all the pieces of the system, including algorithms, operations, hardware, and software. This MF uses the same equation as the other levels with a=b=2 and c=d=3.
- Multiple-Experts is the last level of expertise. This level is similar to 3, but there is a team of experts which is attempting to mount an attack. Equation 12 is used to represent this MF with a=b=3 and c=d=4. Figure 4.21 shows the membership functions for the linguistic values of the Fuzzy variable Expertise.

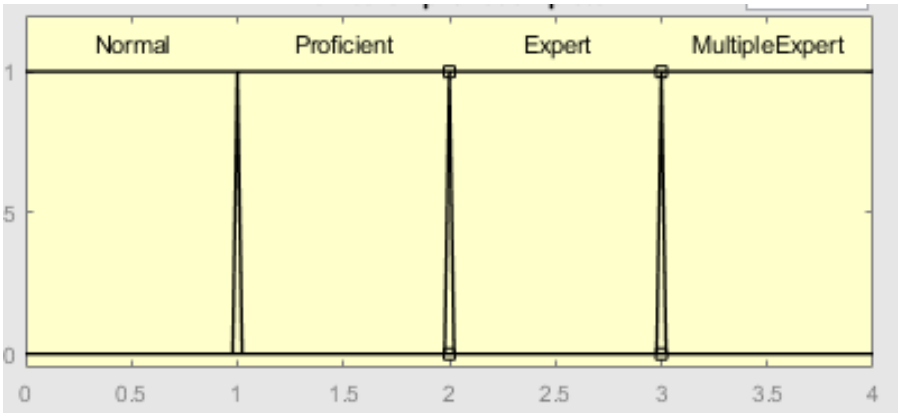

Figure 4.21: FIS2 Expertise

The other element in evaluating the agent level is resources. This property attempts to assess the resources available to mount an attack. The levels of this element align with the majority of the other threat models (also commonly known as equipment) and are as follows:

- Standard resources available means that devices required to perform an attack are widely available and the financial cost is relatively low. Example of this might be OBD cables,

laptops, or debuggers. The separation between these levels is somehow Fuzzy, so these are represented with equation 13 where a = −1 and c =1.

$$f(x;a,b,c) = \begin{cases} 0, x \le a \\ \frac{x-a}{-a}, a \le x \le b \\ \frac{c-x}{c}, b \le x \le c \\ 0, c \le x \end{cases} \quad (13)$$

- Advanced – at this level, the resources are not widely available to the attacker. Although not readily available, these devices or resources can be purchased with a moderate financial cost and be able to mount an attack. Examples of such resources would be a group of PCs, access to cloud computing instances, etc. This level is represented with equation 14 where b = 1 and c=2.

$$f(x;a,b,c) = \begin{cases} 0, x \le a \\ \frac{x}{b}, a \le x \le b \\ \frac{c-x}{c-b}, b \le x \le c \\ 0, c \le x \end{cases} \quad (14)$$

- Specialized resources are not available to the public or cannot be purchased directly at a reasonable cost. They often include resources which need to be manufactured or created separately and might include resources that are restricted. The financial cost to obtain such resources is also a burden for an individual attacker. This uses equation 5 with a = 1, b = 2 and c=3.
- Highly Specialized – Resources required for this level are not available, needs to be designed and manufactured and often require multiple of them. This is the highest level of resources or financial cost required to mount an attack. This also uses equation 5 with a = 2, b = 3 and c = 4.

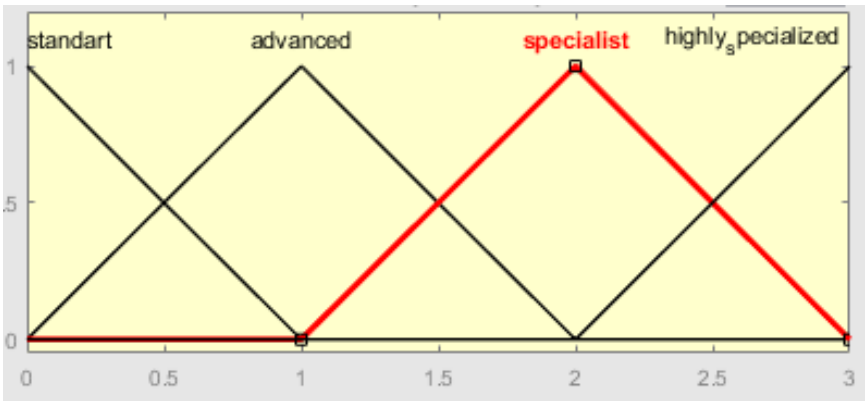

Figure 4.22: FIS2 Resources

The knowledge in this FIS is also generated with the same methodology as the other FIS' and is described in Figure 4.23 as follow:

1. (Expertise==Normal) & (Resources==standart) => (Agent_Level=basic) (1)
2. (Expertise==Normal) & (Resources==advanced) => (Agent_Level=enhanced-basic) (1)
3. (Expertise==Normal) & (Resources==specialist) => (Agent_Level=moderate) (1)
4. (Expertise==Normal) & (Resources==highly_specialized) => (Agent_Level=moderate) (1)
5. (Expertise==Proficient) & (Resources==standart) => (Agent_Level=basic) (1)
6. (Expertise==Proficient) & (Resources==advanced) => (Agent_Level=enhanced-basic) (1)
7. (Expertise==Proficient) & (Resources==specialist) => (Agent_Level=moderate) (1)
8. (Expertise==Proficient) & (Resources==highly_specialized) => (Agent_Level=high) (1)
9. (Expertise==Expert) & (Resources==standart) => (Agent_Level=enhanced-basic) (1)
10. (Expertise==Expert) & (Resources==advanced) => (Agent_Level=moderate) (1)
11. (Expertise==Expert) & (Resources==specialist) => (Agent_Level=high) (1)
12. (Expertise==Expert) & (Resources==highly_specialized) => (Agent_Level=very_high) (1)
13. (Expertise==MultipleExpert) & (Resources==standart) => (Agent_Level=moderate) (1)
14. (Expertise==MultipleExpert) & (Resources==advanced) => (Agent_Level=high) (1)
15. (Expertise==MultipleExpert) & (Resources==specialist) => (Agent_Level=very_high) (1)
16. (Expertise==MultipleExpert) & (Resources==highly_specialized) => (Agent_Level=very_high) (1)

Figure 4.23: FIS2 Rules

The output of the agent level is captured in the following levels. These layers are also used from EVITA and CC:

- Basic: this level essentially describes an attack where the level of expertise is minimal, and resources are standard. A Gaussian function with a standard deviation of 0.5 is used in this case as shown in equation 15.

$$f(x;\ c) = e^{-(x-c)^2} \qquad (15)$$

- Enhanced Basic: This is the level where resources or expertise became more advanced. This agent level is able to mount an attack with some level of expertise of access to certain resources. This graph also uses equation 15 with a different epicenter.
- Moderate: This is the level where the agent has a good understanding of the system or has access to specialized equipment. It uses the same equation as the others
- High: Specialized resources are needed in this level and expertise is also needed. The agent level is high, and the risk is also high
- Very High: The highest level that an agent can be is very high. In this instance, resources are almost unlimited, and expertise is very high with the potential of multiple agents as well.

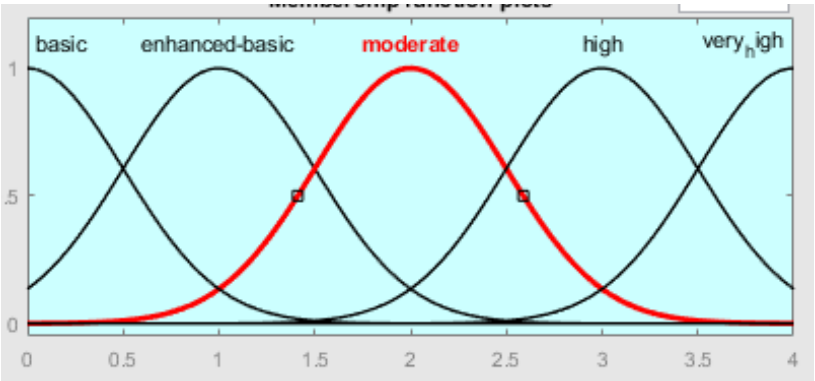

Figure 4.24: FIS2 Agent Level

Their representation and knowledge enable the output surface shown in Figure 4.25.

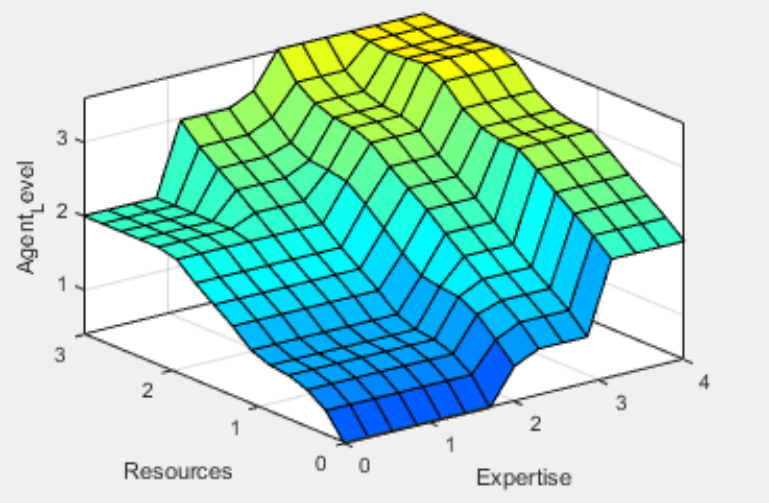

Figure 4.25: FIS2 Surface

As we can observe, the agent level is dependent on the expertise levels and resources.

4.5.2. FIS4: Agent Level and Financial or Other Gains

In order to determine the Attacker's motivation or its full potential to mount a threat, FTAM combines the agent level output from FIS2 with the financial or other gains. Figure 4.26 gives an overview of FIS2.

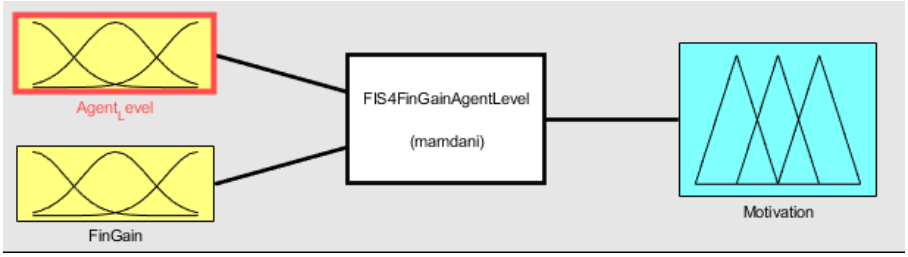

Figure 4.26: FIS4

The agent level input is already described from FIS2. While the Financial and other gain output is a new threat characteristic. Upon research, many of the referenced threat models evaluate this in terms of financial gains only, while others also add other characteristics such as hobby, ideologies, etc. In FTAM, this characteristic is all inclusive, and the users of such a model can assess its level according to their own requirements. Whatever the motivation of the

threat agent is, it is captured in four levels. This means that the end user is allowed to evaluate or assess this characteristic depending on the use case. This FIS uses the Mamdani method. The following membership functions make up Financial and other gain function.

- None – this means that the agent does not have any incentive (whether financial, ideological or others) to perform an attack. In this level FTAM uses a step equation (equation 16) due to a clear separation between the values:

$$f(x;) = \begin{cases} 1, x \leq 0 \\ 0, x \geq 0 \end{cases} \quad (16)$$

- Low – means that the attacker has some sort of gain into performing this attack. The gain in this category is usually for accessing a subsystem of the car. Mechanics modifying certain components might be included in this category. In this category, FTAM uses a Gaussian equation to describe the distribution. The standard deviation used is 0.25. Equation 17 is used in this case.

$$f(x;c) = e^{-4(x-c)^2} \quad (17)$$

- Medium: In this category, the attacker has a considerable gain if the attack is successful. In these type of attacks, the attacker is usually attempting to gain access to the full vehicle in order to accomplish its objective. Equation 17 is also used in this case.
- High: This is the highest level of gain that an attacker could have when mounting an attack. Usually, these types of attacks are due to state-sponsored agencies or terrorist organization where the attacker is highly motivated to perform the attack. This level also uses equation 17.

The following rules are used to describe their relationship and generate the motivation of the attack. As mentioned previously these rules are generated by benchmarking STRIDE attacks

with the other referenced threat models. Figure 4.27 shows the membership functions for the linguistic values of the Fuzzy variable Financial and other gains.

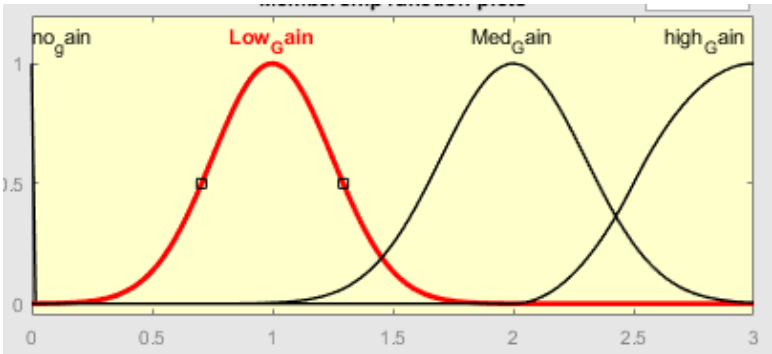

Figure 4.27: FIS4 Financial and Other Gains

Rules shown in Figure 4.28 are used to describe their relationship and generate the motivation of the attack. As mentioned previously these rules are generated by benchmarking STRIDE attacks with the other referenced threat models.

1. (Agent_Level==basic) & (FinGain==no_gain) => (Motivation=no) (1)
2. (Agent_Level==enhanced-basic) & (FinGain==no_gain) => (Motivation=no) (1)
3. (Agent_Level==moderate) & (FinGain==no_gain) => (Motivation=no) (1)
4. (Agent_Level==high) & (FinGain==no_gain) => (Motivation=low) (1)
5. (Agent_Level==very_high) & (FinGain==no_gain) => (Motivation=med) (1)
6. (Agent_Level==basic) & (FinGain==Low_Gain) => (Motivation=low) (1)
7. (Agent_Level==enhanced-basic) & (FinGain==Low_Gain) => (Motivation=low) (1)
8. (Agent_Level==moderate) & (FinGain==Low_Gain) => (Motivation=med) (1)
9. (Agent_Level==high) & (FinGain==Low_Gain) => (Motivation=med) (1)
10. (Agent_Level==very_high) & (FinGain==Low_Gain) => (Motivation=high) (1)
11. (Agent_Level==basic) & (FinGain==Med_Gain) => (Motivation=med) (1)
12. (Agent_Level==enhanced-basic) & (FinGain==Med_Gain) => (Motivation=med) (1)
13. (Agent_Level==moderate) & (FinGain==Med_Gain) => (Motivation=med) (1)
14. (Agent_Level==high) & (FinGain==Med_Gain) => (Motivation=high) (1)
15. (Agent_Level==very_high) & (FinGain==Med_Gain) => (Motivation=high) (1)
16. (Agent_Level==basic) & (FinGain==high_Gain) => (Motivation=med) (1)
17. (Agent_Level==enhanced-basic) & (FinGain==high_Gain) => (Motivation=high) (1)
18. (Agent_Level==moderate) & (FinGain==high_Gain) => (Motivation=high) (1)
19. (Agent_Level==high) & (FinGain==high_Gain) => (Motivation=high) (1)
20. (Agent_Level==very_high) & (FinGain==high_Gain) => (Motivation=high) (1)

Figure 4.28: FIS4 Rules

The output of FIS2 and FIS4 is captured as “motivation.” This is a combination of attackers’ expertise, resources, and gains; therefore, it describes how capable and motivated an attacker is to mount a threat. This is the only branch which uses non-unified membership function. To align

with the ISO26262, EVITA, HEAVENS, and UM models, four levels are used to describe this capability. The following are:

- None – essentially means that if there are no resources, no expertise or no financial or other gains, the attacker is not capable and neither motivated to mount an attack. This level uses the same step function from equation 16 and essentially is able to propagate or maintain an attack with zero level motivation.
- Low – means that an attacker has at least one of the factors (resources, expertise, or gain) to mount an attack. This level is described from a Gaussian equation with standard deviation = 0.4.
- Moderate – means that the attacker has at least two of the factors to mount an attack.
- High – this level describes an attacker where it has all of the elements to mount an attack. They have the knowledge to perform it, they have the resources to describe it, and they are also motivated to pursue it. Equation 18 is used but has a lower standard deviation of 0.1 in order to emphasize the level.

$$f(x;\ \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \qquad (18)$$

Figure 4.29 shows the membership functions for the linguistic values of the Fuzzy variable Motivation.

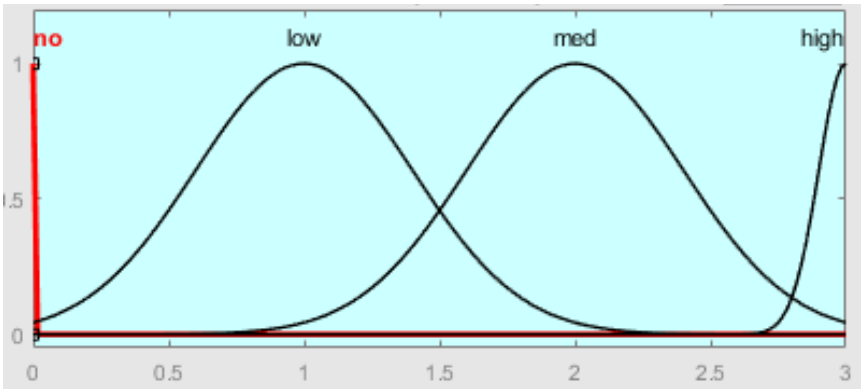

Figure 4.29: FIS4 Motivation

Referring to knowledge created from the rules, membership function properties and following the Mamdani method then surface in Figure 4.30 is created.

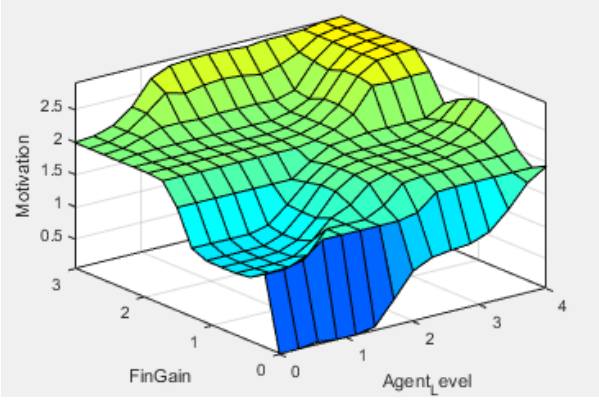

Figure 4.30: FIS4 Surface

4.6. Withstand Potential

The third arm of FTAM is withstand potential. As shown in equation 4, the final threat level is generated by the fuzzification of Attack Impact, Attacker Potential, and Withstand Potential. This last element attempts to measure the system capability for controlling or withstanding a certain attack. To do this, FTAM looks at Difficulty and Controllability. These are similar elements used across other models as well.

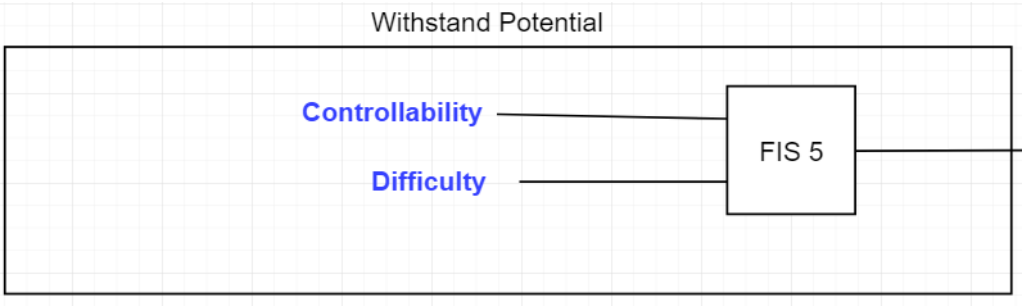

Figure 4.31: Withstand Potential

4.6.1. FIS5: Controllability and Withstand Potential

The fifth Fuzzy Inference System creates an intelligent system which measures Difficulty and Controllability. This FIS is the only one which is not proportionally directed. Therefore, some of the knowledge rules are negated from its usual form. Overview of FIS5 is given in Figure 4.32 .

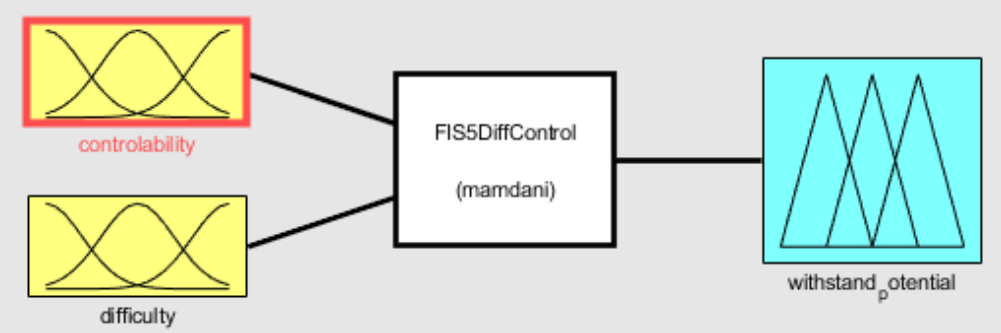

Figure 4.32: FIS5

Controllability is a functional safety element which measures the system capability to control an attack. This element is also included in ISO26262 with four levels (C1-C3). ISO26262 defines controllability as “the ability to avoid the damage through the timely reaction” [122]. In this proposed FIS, the same controllability levels as the ones used in ISO26262 are used

- Controllable in General – This category is used when an attack results in the unavailability of a feature which does not cause vehicle safe operation. For this purpose, FTAM uses a Gaussian equation 18 with a standard deviation of 0.1.

- Simply controllable – This also follows the ISO standard. In parallelism, this means that 99% of attacks can be controlled. Its membership functions are also designed using a Gaussian distribution equation 18 with a standard deviation of 0.475 in order to give the required transition.
- Normally controllable – In this level, more than 90% of attacks are able to be controlled and contained by the driver. It is represented with the same equation and standard distribution as above.
- Difficult to control – Lastly, difficult to control means that less than 90% of these types of attacks can be controlled or contained from the system. This corresponds with C3 level in ISO26262. The same equation 18 is also used at this level.

Figure 4.33 shows the membership functions for the linguistic values of the Fuzzy variable, Controllability.

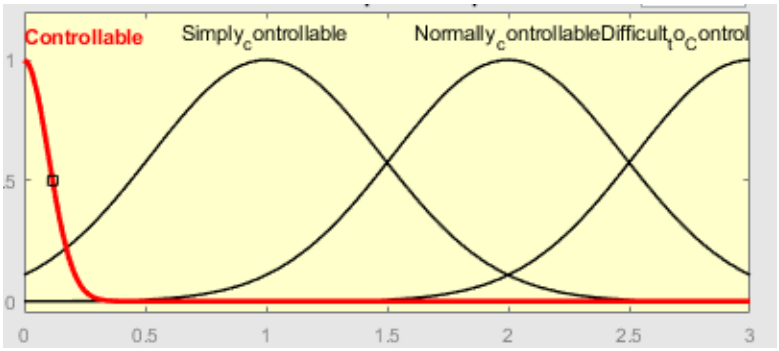

Figure 4.33: FIS5 Controllability

The other element to assess Withstand Potential is difficulty to mount an attack. This characteristic is also similar to the ones from other threat models and is represented using three

levels. Low, Medium, and High. Note the scale for this characteristic is from the NHTSA threat matrix. According to [12], these levels are defined as follows:

- Low – means that an attack is not difficult to implement and requires no specialized knowledge to be mounted. For this purpose, a triangular shaped curve with elements: a = 0, b = 1 and c = 2 is used. Its representative equation is 19.

$$f(x; c, b) = \max(\min\left(\frac{x}{b}, \frac{c-x}{c-b}\right), 0) \qquad (19)$$

- Medium – means that an attack requires some level of knowledge to be mounted and is moderately complex to be implemented. In addition, these kinds of attacks often require physical access to the vehicle.

$$f(x; a, b, c) = \max(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0) \qquad (20)$$

- High – the most difficult level, and it means that the attack requires a high amount of knowledge to be implemented and it is also complex in nature. These attacks require prolonged access to the physical vehicle. Equation 20 is used in this case as well.

Figure 4.34 shows the membership function used for difficulty.

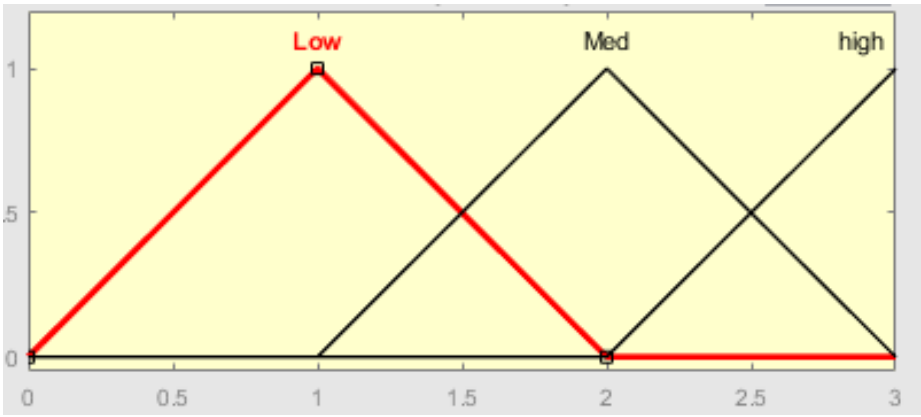

Figure 4.34: FIS5 Difficulty

The output of FIS5 is the actual withstand potential. This output is measured in the following three levels

- Low – means that the capability of the system to withstand an attack is low, and often attacks are successful. In all three of these membership functions, a Gaussian function is used. This has a low standard deviation = 0.2 in order to show the separation of the attacks.
- Medium – means that the system can withstand the majority of the attacks due to either being complex or being able to control them.
- High – means that the system can withstand most of the attacks.

Graph generated from these membership functions is given in Figure 4.35:

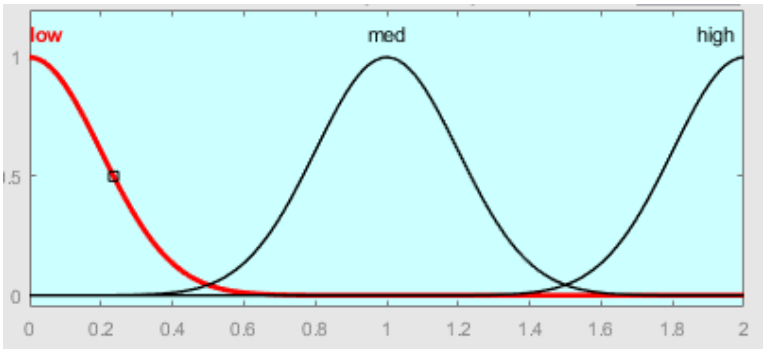

Figure 4.35: FIS5 Withstand Potential

FIS5 also has a knowledge base of rules generated in a similar manner like the other FIS. Figure 4.36 gives a list of these 12 rules.

1. (controlability==Controllable) & (difficulty==Low) => (withstand_potential=med) (1)
2. (controlability==Simply_controllable) & (difficulty==Low) => (withstand_potential=med) (1)
3. (controlability==Normally_controllable) & (difficulty==Low) => (withstand_potential=low) (1)
4. (controlability==Difficult_to_Control) & (difficulty==Low) => (withstand_potential=low) (1)
5. (controlability==Controllable) & (difficulty==Med) => (withstand_potential=high) (1)
6. (controlability==Simply_controllable) & (difficulty==Med) => (withstand_potential=med) (1)
7. (controlability==Normally_controllable) & (difficulty==Med) => (withstand_potential=med) (1)
8. (controlability==Difficult_to_Control) & (difficulty==Med) => (withstand_potential=low) (1)
9. (controlability==Controllable) & (difficulty==high) => (withstand_potential=high) (1)
10. (controlability==Simply_controllable) & (difficulty==high) => (withstand_potential=high) (1)
11. (controlability==Normally_controllable) & (difficulty==high) => (withstand_potential=med) (1)
12. (controlability==Difficult_to_Control) & (difficulty==high) => (withstand_potential=low) (1)

Figure 4.36: FIS5 Rules

After implementing the membership functions, and using the Mamdani method, the surface in Figure 4.37 is generated:

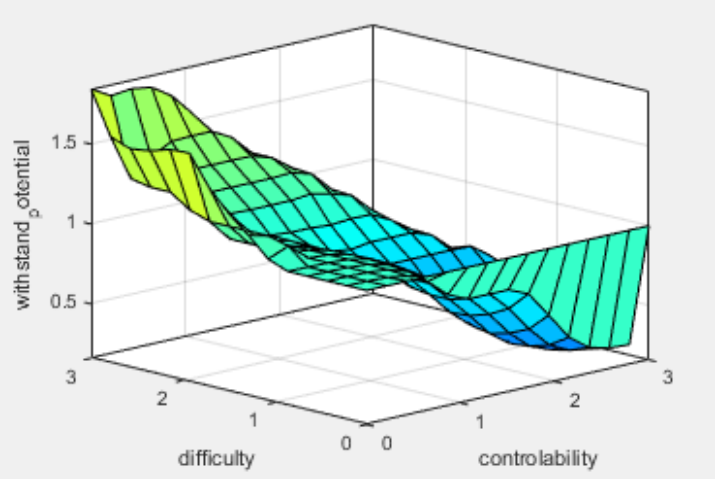

Figure 4.37: FIS5 Surface

As described previously, due to the negated nature or the inverse relationship of these elements, the graph surface shown in Figure 44 is different from the other FIS.

4.7. Threat Level

After evaluating or assessing the three different aspects of a Threat (Attack Potential, Attacker Capabilities and Withstand Potential), a final Fuzzy Inference System is used to determine the final Threat Level (TL) based on these inputs. The final diagram of FTAM is shown in Figure 4.38:

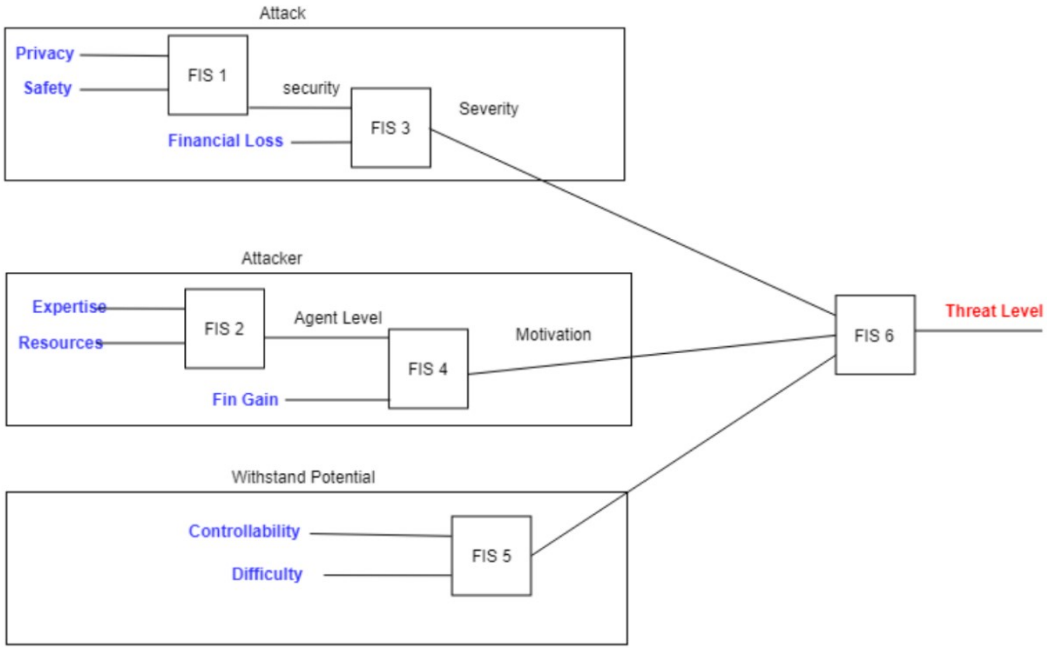

Figure 4.38: FTAM Architecture

As observed, this is a multistage architecture made up from six Fuzzy Inference Systems. So far, FIS1-5 were explained in detail, their knowledge systems, rules, and methodologies. The output of FIS3, FIS4, and FIS5 serve as the inputs for FIS6 in order to establish a threat level.

4.7.1. FIS 6: Severity, Motivation, and Likelihood

FIS6 has three inputs and uses the Mamdani method. Looking at ISO26262, MISRA Safety Analysis [109] guidelines or other functional safety standards it is observed that a risk is assessed based on severity and probability. In order to stay close to these functional safety standards, FTAM uses a four level Threat Level (TL) assignment in this model – named E0 to E3.

These levels are given in the following graph, and every Membership function is described using a Gaussian equation 11 with a calculated standard deviation of $\sigma = 0.2778$

The diagram of FIS6 is shown in Figure 4.39:

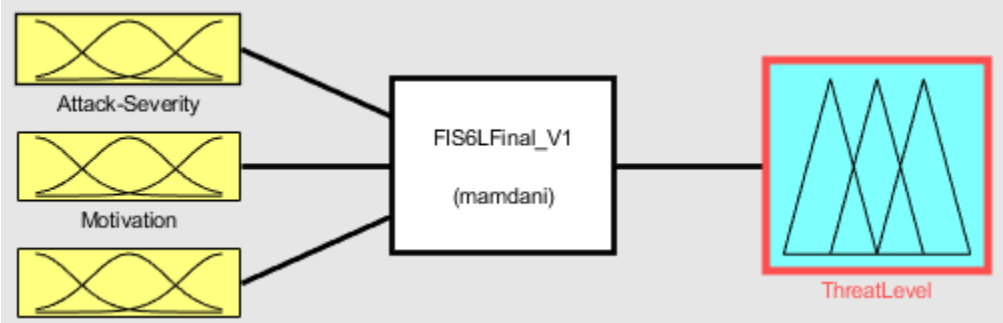

Figure 4.39: FIS6

As shown, FIS6 has three inputs and uses the Mamdani method. By looking at ISO26262, MISRA Safety Analysis guidelines or other functional safety standards, it is observed that risk is assessed based on severity and probability. In order to stay close to these functional safety standards, a four level Threat Level (TL) model is proposed – named E0 to E3. These levels are given in the following graph, and every Membership function is described using a Gaussian equation (11) with a calculated standard deviation of σ=0.2778.

Those levels are defined and mapped with ISO26262:

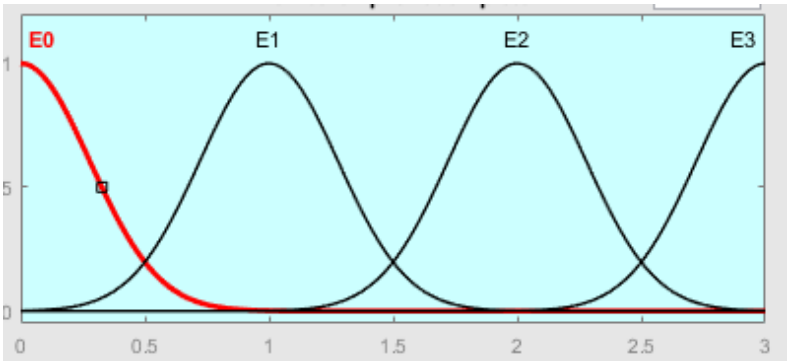

Figure 4.40: FIS6 Threat Level

Benchmarking STRIDE threats against OWASP, EVITA, and HEAVENS, we have built the following knowledge – represented by these rules. Since there are 3 inputs, the number of rules is higher than other FIS as shown in Figure 4.41,Figure 4.42, and Figure 4.43.

1. (Attack-Severity==no) & (Motivation==low) & (withstand_potential==high) => (ThreatLevel=E0) (1)
2. (Attack-Severity==no) & (Motivation==med) & (withstand_potential==high) => (ThreatLevel=E0) (1)
3. (Attack-Severity==no) & (Motivation==high) & (withstand_potential==high) => (ThreatLevel=E0) (1)
4. (Attack-Severity==no) & (Motivation==no) & (withstand_potential==high) => (ThreatLevel=E0) (1)
5. (Attack-Severity==low) & (Motivation==no) & (withstand_potential==high) => (ThreatLevel=E1) (1)
6. (Attack-Severity==low) & (Motivation==low) & (withstand_potential==high) => (ThreatLevel=E1) (1)
7. (Attack-Severity==low) & (Motivation==med) & (withstand_potential==high) => (ThreatLevel=E1) (1)
8. (Attack-Severity==low) & (Motivation==high) & (withstand_potential==high) => (ThreatLevel=E1) (1)
9. (Attack-Severity==med) & (Motivation==no) & (withstand_potential==high) => (ThreatLevel=E1) (1)
10. (Attack-Severity==med) & (Motivation==low) & (withstand_potential==high) => (ThreatLevel=E1) (1)
11. (Attack-Severity==med) & (Motivation==med) & (withstand_potential==high) => (ThreatLevel=E2) (1)
12. (Attack-Severity==med) & (Motivation==high) & (withstand_potential==high) => (ThreatLevel=E2) (1)
13. (Attack-Severity==high) & (Motivation==no) & (withstand_potential==high) => (ThreatLevel=E1) (1)
14. (Attack-Severity==high) & (Motivation==low) & (withstand_potential==high) => (ThreatLevel=E2) (1)
15. (Attack-Severity==high) & (Motivation==med) & (withstand_potential==high) => (ThreatLevel=E2) (1)
16. (Attack-Severity==high) & (Motivation==high) & (withstand_potential==high) => (ThreatLevel=E2) (1)
17. (Attack-Severity==very__high) & (Motivation==no) & (withstand_potential==high) => (ThreatLevel=E2) (1)
18. (Attack-Severity==very__high) & (Motivation==low) & (withstand_potential==high) => (ThreatLevel=E2) (1)
19. (Attack-Severity==very__high) & (Motivation==med) & (withstand_potential==high) => (ThreatLevel=E2) (1)
20. (Attack-Severity==very__high) & (Motivation==high) & (withstand_potential==high) => (ThreatLevel=E2) (1)

Figure 4.41: FIS6 Rules 1-20

20. (Attack-Severity==very__high) & (Motivation==high) & (withstand_potential==high) => (ThreatLevel=E2) (1)
21. (Attack-Severity==no) & (Motivation==no) & (withstand_potential==med) => (ThreatLevel=E0) (1)
22. (Attack-Severity==no) & (Motivation==low) & (withstand_potential==med) => (ThreatLevel=E0) (1)
23. (Attack-Severity==no) & (Motivation==med) & (withstand_potential==med) => (ThreatLevel=E0) (1)
24. (Attack-Severity==no) & (Motivation==high) & (withstand_potential==med) => (ThreatLevel=E0) (1)
25. (Attack-Severity==low) & (Motivation==no) & (withstand_potential==med) => (ThreatLevel=E1) (1)
26. (Attack-Severity==low) & (Motivation==low) & (withstand_potential==med) => (ThreatLevel=E1) (1)
27. (Attack-Severity==low) & (Motivation==med) & (withstand_potential==med) => (ThreatLevel=E1) (1)
28. (Attack-Severity==low) & (Motivation==high) & (withstand_potential==med) => (ThreatLevel=E1) (1)
29. (Attack-Severity==med) & (Motivation==no) & (withstand_potential==med) => (ThreatLevel=E2) (1)
30. (Attack-Severity==med) & (Motivation==low) & (withstand_potential==med) => (ThreatLevel=E2) (1)
31. (Attack-Severity==med) & (Motivation==med) & (withstand_potential==med) => (ThreatLevel=E2) (1)
32. (Attack-Severity==med) & (Motivation==high) & (withstand_potential==med) => (ThreatLevel=E2) (1)
33. (Attack-Severity==high) & (Motivation==no) & (withstand_potential==med) => (ThreatLevel=E2) (1)
34. (Attack-Severity==high) & (Motivation==low) & (withstand_potential==med) => (ThreatLevel=E2) (1)
35. (Attack-Severity==high) & (Motivation==med) & (withstand_potential==med) => (ThreatLevel=E2) (1)
36. (Attack-Severity==high) & (Motivation==high) & (withstand_potential==med) => (ThreatLevel=E3) (1)
37. (Attack-Severity==very__high) & (Motivation==no) & (withstand_potential==med) => (ThreatLevel=E3) (1)
38. (Attack-Severity==very__high) & (Motivation==low) & (withstand_potential==med) => (ThreatLevel=E3) (1)
39. (Attack-Severity==very__high) & (Motivation==med) & (withstand_potential==med) => (ThreatLevel=E3) (1)
40. (Attack-Severity==very__high) & (Motivation==high) & (withstand_potential==med) => (ThreatLevel=E3) (1)

Figure 4.42: FIS6 Rules 20-40

40. (Attack-Severity==very__high) & (Motivation==high) & (withstand_potential==med) => (ThreatLevel=E3) (1)
41. (Attack-Severity==no) & (Motivation==no) & (withstand_potential==low) => (ThreatLevel=E0) (1)
42. (Attack-Severity==no) & (Motivation==low) & (withstand_potential==low) => (ThreatLevel=E0) (1)
43. (Attack-Severity==no) & (Motivation==med) & (withstand_potential==low) => (ThreatLevel=E0) (1)
44. (Attack-Severity==no) & (Motivation==high) & (withstand_potential==low) => (ThreatLevel=E0) (1)
45. (Attack-Severity==low) & (Motivation==no) & (withstand_potential==low) => (ThreatLevel=E1) (1)
46. (Attack-Severity==low) & (Motivation==low) & (withstand_potential==low) => (ThreatLevel=E1) (1)
47. (Attack-Severity==low) & (Motivation==med) & (withstand_potential==low) => (ThreatLevel=E2) (1)
48. (Attack-Severity==low) & (Motivation==high) & (withstand_potential==low) => (ThreatLevel=E3) (1)
49. (Attack-Severity==med) & (Motivation==no) & (withstand_potential==low) => (ThreatLevel=E2) (1)
50. (Attack-Severity==med) & (Motivation==low) & (withstand_potential==low) => (ThreatLevel=E3) (1)
51. (Attack-Severity==med) & (Motivation==med) & (withstand_potential==low) => (ThreatLevel=E3) (1)
52. (Attack-Severity==med) & (Motivation==high) & (withstand_potential==low) => (ThreatLevel=E3) (1)
53. (Attack-Severity==high) & (Motivation==no) & (withstand_potential==low) => (ThreatLevel=E3) (1)
54. (Attack-Severity==high) & (Motivation==low) & (withstand_potential==low) => (ThreatLevel=E3) (1)
55. (Attack-Severity==high) & (Motivation==med) & (withstand_potential==low) => (ThreatLevel=E3) (1)
56. (Attack-Severity==high) & (Motivation==high) & (withstand_potential==low) => (ThreatLevel=E3) (1)
57. (Attack-Severity==very__high) & (Motivation==no) & (withstand_potential==low) => (ThreatLevel=E3) (1)
58. (Attack-Severity==very__high) & (Motivation==low) & (withstand_potential==low) => (ThreatLevel=E3) (1)
59. (Attack-Severity==very__high) & (Motivation==med) & (withstand_potential==low) => (ThreatLevel=E3) (1)
60. (Attack-Severity==very__high) & (Motivation==high) & (withstand_potential==low) => (ThreatLevel=E3) (1)

Figure 4.43: FIS6 Rules 40-60

4.8. Final Integration

After generating the knowledge base and FIS1-FIS6 individually, Simulink is used to integrate all of them together into a multistage architecture similar to [28]. Figure 4.44 provides a high-level diagram of the system.

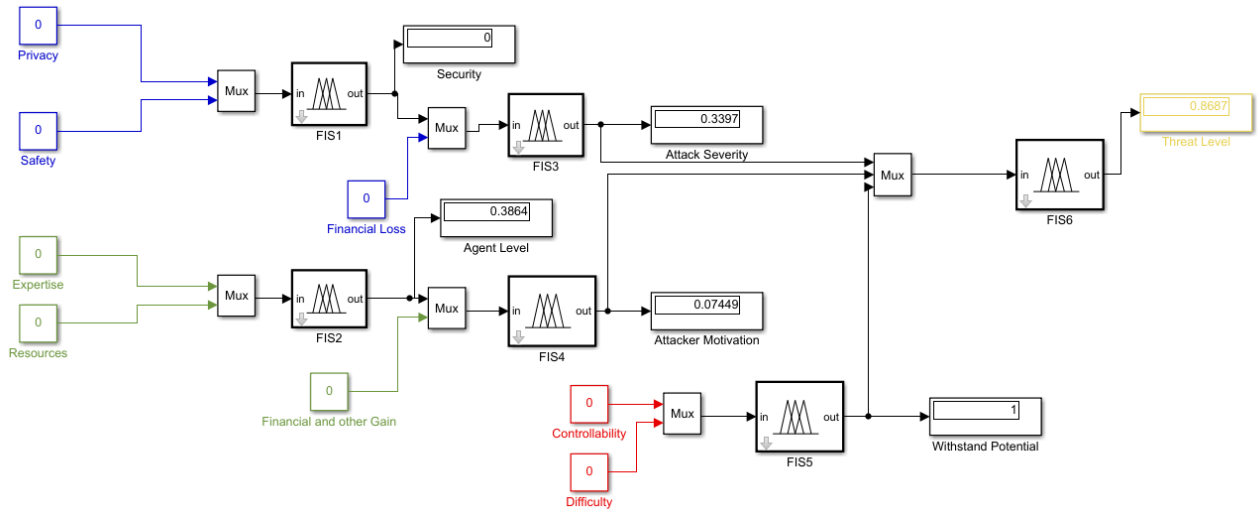

Figure 4.44: FTAM Simulink Model

Blue elements characterize the Attack potential, green shows the Attacker capabilities, and red shows the withstand potential. The output of this model is a Threat Level. The individual

Fuzzy Inference Systems are build using MATLAB, specifically the Fuzzy Logic Designer application. After designing and testing the individual FISs, Simulink is used to integrate the blocks together. Special consideration is given so that the inputs and outputs for each FIS align accordingly. All the rules are accounted for since it is expected that they all can be triggered. The Simulink model is designed to show the output after each combination or layer. Constants and MUX blocks are used for input. The code given in Appendix A can instantiate the individual FIS elements in the Simulink model so it can be used or integrated into other applications.

Chapter 5. Detect: Two Stage Intrusion Detection Intelligent System based on FTAM[2]

During this three-step framework targeted mainly at automotive cybersecurity, the "Detect" phase is defined as a mean to detect malicious threats. In this dissertation, an intrusion detection system (IDS) is proposed to detect malicious attacks in a regular network while taking into consideration the results from the previous assessment stage. In addition, performance evaluation along with computational requirements are analyzed. The proposed IDS consists of a two-stage Intrusion Detection System based on machine learning algorithms. The first stage labels (detects) whether there is an attack present, and the second stage classifies these attacks in a supervised learning methodology. The second stage also addresses and eliminates the number of false positives. The simulation of this approach results is an IDS able to detect and classify a 99.965% accuracy and lower the false positives rate to 0%. In addition, the algorithmic results are correlated with threat assessment levels to build a relationship. To build such a system, the dataset from Wyoming Connected Vehicle Deployment program is used [3].

The organization of this Chapter is as follow: Section 5.1 will give an overview of the dataset used, 5.2 will discuss the adversary models for the threats, 5.3 will describe some of the data pre-processing methodologies, and 5.4 will describe in detail the proposed two-stage IDS.

[2] A version of this chapter was previously published as an article in a peer-reviewed journal: N. Kaja, A. Shaout, and D. Ma, "*An Intelligent Intrusion Detection System*," *J. Appl. Intell.*, pp. 1–13, 2019

5.1. Data Set: Wyoming Connected Vehicle Pilot

5.1.1. Data Collection

As mentioned previously, V2X is a relatively new technology, and currently there are not a lot of vehicle manufacturers that have deployed the technology in production as of 2018. Based on a quick internet search of recent news, only General Motors and Audi have currently deployed this technology on the road. Even these efforts are limited in scope. This makes it difficult to access readily available datasets for research and find well-studied performance results in V2X security or intrusion detection models.

From a research perspective, both academia and industry have done an extensive amount of research regarding V2X performance and other aspects. The majority of these research studies have often been theoretical, analytical, or functional in nature. In 2015, the US Department of Transportation launched a Connected Vehicle pilot program sponsored by the Intelligent Transportation Systems Joint Program office. Through this program, three state agencies (New York City DOT Pilot, Tampa-Hillsborough Expressway Authority Pilot and Wyoming DOT Pilot) started pilot programs. All three pilots had different objectives, ranging from functionality, interoperability, security using industry standards. Publications and deliverables of these pilots are found on the DOT website: https://www.its.dot.gov/pilots/technical_assistance_events.htm (as of 03/28/2019).

This dissertation leverages the data produced from the Wyoming DOT Pilot since that is one of the first pilots that made its data publicly available[17]. Data consists of actual Basic Safety Messages (BSM) and was gathered during February–March of 2018 and published in the same year at www.data.gov. To put the timeline and the freshness of data in perspective: Wyoming

CV Pilot Conceptual Development Phase goes from 09/2015–08/2016, Design/Build/Test Phase goes from September 2016 – Spring 2019 and Operate/Maintenance Phase goes from Fall 2018–Fall 2020. Research for this dissertation was done in 2018 and early 2019.

Wyoming DOT Pilot data was gathered from a difficult highway segment (I80) with a significant number of accidents, mainly due to weather conditions. According to DOT, in four years, Interstate 80 in Wyoming (where data was gathered) produced over 200 trucks rollovers from high winds, 86 road closures and roughly $12Million loss due to each closure. Figure 5.1 gives a better view of the map where data was gathered:

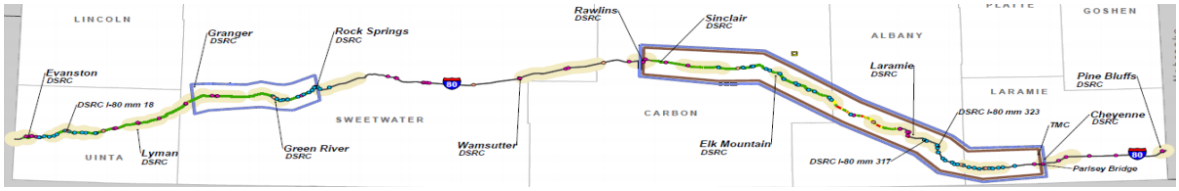

Figure 5.1: Wyoming I80 Corridor - Connected Vehicle Map [3]

Based on the current literature survey, this study is one of the first to study and publish results out of this dataset.

5.1.2. Data Description

The dataset used for proposing the IDS contains core and non-core BSM elements. The standard for BSM is described in SAE-J2735 standard [123]. Basic Safety Messages are used to exchange vehicle safety status data according to the DSRC standard. Messages are broadcasted 10 times per second. The file used for this experiment was downloaded from the official Federal Government repository (www.data.gov) and contains over 4 Million basic safety messages. To facilitate handling, this large file was split into 4 smaller files and attacks were injected according to the adversary models defined in 3.1-3.6. Data used for training was gathered in the segment shown in Figure 5.2.

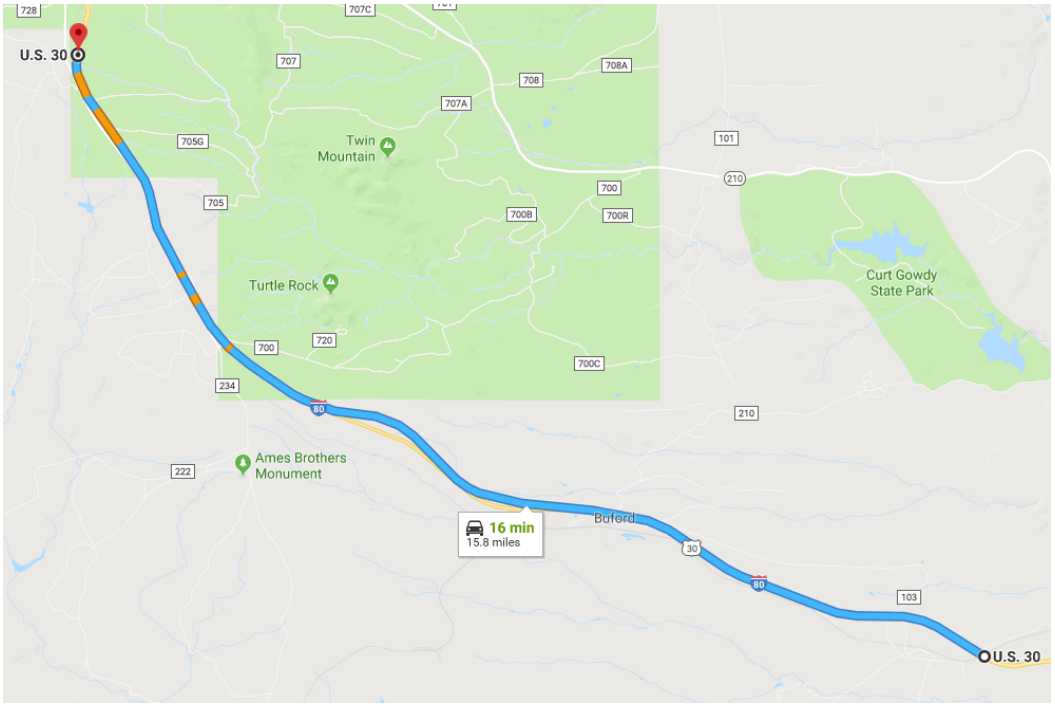

Figure 5.2: Google Map Generated from Training Data BSM Points

Start and end points are generated from the start BSM Latitude/Longitude and end BSM Latitude/Longitude. Table 5.1 shows the core elements contained in Part 1 of BSM and used as features for the proposed Intrusion Detection System:

Table 5.1: BSM Data Dictionary According to [123]

Data Element	Description
coreData_msgCnt	MsgCount data element is used to provide a sequence number within a stream of messages with the same DSRCmsgID and from the same sender
coreData_id	This is the 4-octet random device identifier, called the TemporaryID. When used for a mobile OBU device, this value will change periodically to ensure the overall anonymity of the vehicle, unlike typical wireless or wired 802 device ID.
coreData_secMark	The DSRC second expressed in this data element consists of integer values from zero to 60999, representing the milliseconds within a minute.
coreData_position_lat	The geographic latitude of an object, expressed in 1/10th integer microdegrees, as a 31-bit value, and with reference to the horizontal datum then in use

coreData_position_long	The geographic longitude of an object, expressed in 1/10th integer microdegrees, as a 32-bit value, and with reference to the horizontal datum then in use.
coreData_elevation	The DE_Elevation data element represents the geographic position above or below the reference ellipsoid (typically WGS-84).
coreData_accelset_accelYaw	This data frame is a set of acceleration values in 3 orthogonal directions of the vehicle and with yaw rotation rates, expressed as a structure.
coreData_accuracy_semiMajor	The DE_SemiMajorAxisAccuracy data element is used to express the radius (length) of the semi-major axis of an ellipsoid representing the accuracy which can be expected from a GNSS system in 5cm steps, typically at a one sigma level of confidence.
coreData_accuracy_semiMinor	The DE_SemiMinorAxisAccuracy data element is used to express the radius of the semi-minor axis of an ellipsoid representing the accuracy which can be expected from a GNSS system in 5cm steps, typically at a one sigma level of confidence
coreData_transmission	The DE_TransmissionState data element is used to provide the current state of the vehicle transmission
coreData_speed	This data element represents the vehicle speed expressed in unsigned units of 0.02 meters per second.
coreData_heading	The DE_Heading data element provides the current heading of the sending device, expressed in unsigned units of 0.0125 degrees from North such that 28799 such degrees represent 359.9875 degrees
coreData_brakes_wheelBrakes_leftFront	The Brake System Status data frame conveys a variety of information about the current brake and system control activity of the vehicle. The structure consists of a sequence of items which provide status flags for any active brakes per wheel, the traction control system, the anti-lock brake system, the stability control system, the brake boost system, and the auxiliary brake system.
coreData_brakes_wheelBrakes_rightFront	
coreData_brakes_wheelBrakes_unavailable	
coreData_brakes_wheelBrakes_leftRear	
coreData_brakes_wheelBrakes_rightRear	
coreData_brakes_traction	
coreData_brakes_abs	
coreData_brakes_scs	
coreData_brakes_brakeBoost	
coreData_brakes_auxBrakes	
coreData_size	The DF_VehicleSize is a data frame representing the vehicle length and vehicle width in a single data concept.
coreData_SteeringWheelAngle	The angle of the driver's steering wheel, expressed in a signed (to the right being positive) value with LSB units of 1.5 degrees.

5.2. Adversary Model

Attacks injected in the dataset are defined according to the STRIDE model. Not all of the STRIDE attacks are injected, but rather the focus is on the ones that were assessed at different levels according to FTAM and are BSM level attacks. Injected attacks are described below and give an overview of the adversary model used.

- Spoofing (S) – this attack is assessed as E2 from FTAM – The attack is injected by masquerading certain BSM data, so it looks like the messages are coming from a certain vehicle. The dataset contains 95 spoofing packages, from a specific OBU ID with spoofed vehicle safety elements. The spoofed vehicle attempts to lower the speed immediately.
- Tempering (T) – this attack is assessed as E3 from FTAM – The dataset contains 1,967 BSM messages with tempered speed and gear values. In a real scenario, this would look like a reckless driving car.
- Information Disclosure (I) – this attack is assessed as E1 from FTAM – This attack is realized by eavesdropping or trying to listen in the communication channel in order to intercept communication. 1,648 BSM messages are injected with a specific coreData_id, which correspondents with a device identifier.
- Denial of Service (D) – this attack is assessed as E3 from FTAM – This attack attempts to disrupt the V2X network by jamming the communication and resending a massive number of packages. In our scenario, we have resent the same package 220 times in a short period of time.

5.3. Feature Engineering

After performing preliminary tests on the dataset, it is discovered there are a significant number of redundancies and similarities in the dataset. From these initial tests, it is observed that training leads to a bias towards the more frequent records. The data itself is also too large to process, and often initial results lead to overfitting. This causes the model to perform well on the training set, but not as well on the test data. To pre-process the data, correlated features are identified and removed to avoid overfitting. These features are closely correlated with each other and are unable to allow the system to infer much knowledge from them [26] .

Since the data comes with 24 features, then dimensionality reduction is needed. First, the variance of feature values is calculated. Variance measures the spread between features in the dataset. Features with low variances usually do not give much knowledge. Therefore, they were filtered out during training. Equation (21) shows the variance.

$$\sigma^2 = \frac{\sum_{k=0}^{n} x^2}{N} - \mu^2 \qquad (21)$$

Another method to reduce the number of features is to look at their correlation coefficient. Correlation coefficient shows how much the variances of two variables are associated with each other. Equation (22) is used to calculate the correlation coefficient among the given features.

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \ \sum(y - \bar{y})^2}} \qquad (22)$$

In addition to variance and correlation coefficient, “Least Square Regression Error” (LSQE) and “Maximal Information Compression Index” (MICI) are used to minimize the features’ similarity and maximize dimensionality reduction.

LSQE or residual variance is the error of predicting y from y = bx + a while a and b are the regression coefficients which can be calculated by minimizing e(x, y)2 in (23). (24), (25) and (25) show the derivation of them,

$$e(x,y)^2 = \frac{1}{n}\sum e(x,y)_i^2 \quad (23)$$

$$e(x,y)^2 = y_i - a - bx_i \quad (24)$$

$$a = \bar{y} \quad (25)$$

$$b = \frac{cov\ (x,y)}{var(x)} \quad (26)$$

Once a and b are calculated, the mean square error e(x, y) can then be calculated by (27)

$$e(x,y) = var(x) \times (1 - relation(x,y)^2) \quad (27)$$

(27) essentially measures the relationship between two features x and y. If these features have a linear relationship, then e(x, y) will be 0, and if they do not have a relationship, then e(x, y) will be equal to var(x).

MICI, denoted by λ2(x, y), is calculated using (28)

$$\lambda_2(x,y) = min(eigv(\Sigma)) \quad (28)$$

Where $\sum$ provides the covariance of x and y while eigv is a vector of Eigenvalues of that covariance. When MICI is 0 the features have a linear relationship. A higher MICI means that the relationship is non-existent.

From all of the four of the methods described above, 12 features are identified to be removed from the data. These features are similar enough, and they do not provide distinct

knowledge to the training model of the IDS. Therefore, the number of features in the dataset is reduced from 24 to 12.

As an additional clarification note, while analyzing the data, the value coreData_id is a numerical value, but it contained a “string” value due to a random “B” added at the end of one of the values. This value was being treated as string and was causing an error in the programming. After some validation, it was concluded that this might be a potential error from a data collection method, so “3132544B” is replaced with “31325440” to keep the dataset consistent.

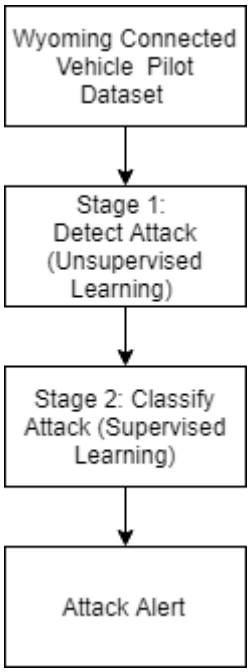

Figure 5.3: Intrusion Detection System Architecture

5.4. Two Stage IDS

After “feature engineering” or data pre-processing, the next step is to design a two-stage, intrusion detection system architecture based on machine learning algorithms. The contribution and novelty of this architecture is the ability to use a two-stage machine learning approach in the design of IDS. This is done in order to avoid false positives and accurately detect/prevent an attack. The objective of the first phase (stage) is to simply detect whether there is an attack

present or not. The second phase (stage) is to classify that attack and provide an alert for it. Figure 5.3 shows the architecture block diagram of the proposed system.

5.4.1. First stage: Detect!

The first stage uses an unsupervised, clustering algorithm to detect an attack. Clustering in this phase simply classifies the data into two categories: “attack” connections or “normal” connections. The design objective for this stage is to simply detect if there is an attack or not. To accomplish this classification, two clusters with low intercluster similarity and good intra-cluster similarity are built.

One of the tasks in this part of the study was to select the right algorithm for clustering. The dataset was unknown, and unfortunately, there are no other papers available to benchmark it against. Being the first work in this area, the data were tested, analyzed, and benchmarked against a number of clustering algorithms such as Canopy Clustering [124], Density-Based K-Means, Filtered Cluster, K-Means [26], and FarthestFirst [125]. Performance and time to train/predict were used as decisive factors. K-Means and its variants were outputting homogenous clusters, which is not feasible for our data since only 3.42% of the messages are malicious. A less known algorithm, FarthestFirst [126], [125], which is modeled on K-Means, performed significantly better than all the rest as shown in Table 5.2.

Table 5.2: Clustering Algorithm Performances

Algorithm	Cluster 1 (Attack)	Cluster 2 (Normal)
Canopy	10%	90%
Density Based K-Means	49%	51%
FilteredCluster	49%	51%
K-Means	49%	51%
FarthestFirst	2%	98%
Ground Truth	*3.42%*	*96.58%*

FarthestFirst is a version of K-means algorithm and based on the farthest-first traversal k-center. This algorithm is substantially faster than a regular K-Means. In this experiment, FarthestFirst was 10.8 times faster than a simple K-Means. FarthestFirst works in the following manner. There are X(1), ... ,X(n) BSM messages on the data (D). This data is described by 12 features and $f(x_{i,j}D)$ describes the frequency count of feature value xi,j in the dataset [127].

A scoring function is used to evaluate each point:

$$Score(X_i) = \sum_{j=1}^{12} f(x_{i,j}|D) \quad (29)$$

The following steps are used to in the algorithm [127]:

- Randomly select the first cluster1 center
    - For every other point calculate the distance with the cluster1 center
- Select the point with the maximum distance as the cluster2 center
    - For every other point calculate and assign points with either of the cluster centers according to the minimal distance.

The objective of the algorithm is to separate this in a few clusters – in this case, the objective is to split it into two clusters (attack and normal).

Figure 5.4 below shows the visual results of data clustering for attack detection based on the FarthestFirst algorithm.

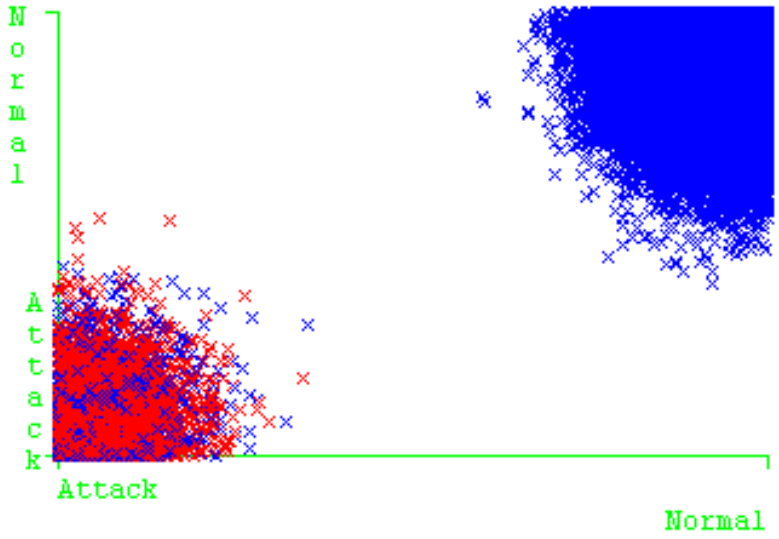

Figure 5.4: FarthestFirst Clustering

As shown in Figure 5.4, there are two well-separated clusters. The blue cluster shows the normal messages and the red cluster shows the malicious messages. It is important to note that the blue cluster (normal) is clear from red points (malicious messages). The objective in this stage is for the blue cluster to be as "clean" as possible from attacks to reduce the number of messages to be processed in the second stage. It is OK if the red cluster includes some blue (normal) messages because the second stage will help with reducing the number of false positives. Results are given in the table below:

Table 5.3: Stage 1 Results (Detect)

	Clustering	True Labels
Normal	98%	96.58%
Attack	2%	3.42%
Time to Build Model	0.14s	
Normal Cluster Center	105.0 3.031E7 15600.0 41.2336304 -105.43659 2616.5 0.0 3.25 8.0 NEUTRAL 20.44 3.3 Normal Normal	
Attack Cluster Center	1.0 3.032E7 58100.0 41.2391481 -105.4375635 2631.8 0.0 2.35 2.95 forwardGears 1655.7 348.85 Tempering Attack	
Stage 1 Accuracy	98.58%	

After building the model, the detection accuracy for the first stage is 98.58%.

5.4.2. Second Stage: Classify!

After detecting an attack in phase one, the next step is to classify that attacks in stage two. Please note that although some performance results are given in this Chapter, the performance analyzation is done in Chapter VII. Subsections below describe the four machine learning algorithms used for stage two.

5.4.2.1. J48

J48 is a decision tree-based algorithm which classifies data. This algorithm builds decision trees by using information entropy and is based on the C4.5 decision tree. This algorithm is often referred to as a statistical classifier because it bases its decision tree on labeled input data. When building the tree, J48 chooses the attributes based on the information gain, or whichever attribute results in the most efficient split of the data. The steps below describe how the algorithm works in detail:

1- Calculate the entropy using equation (30)

$$Entropy = \sum_{i=1}^{C} - p_i * \log_2(p_i) \quad (30)$$

2- Calculate information gain rate using equation (31). The gain is basically the difference of prior entropy (T) and the entropy of the selected branch (X).

$$Gain(T, X) = Entropy(T) - Entropy(X) \quad (31)$$

3- After calculating the Gain for each candidate attribute, then the data is split based on the attribute with the highest information gain.

Repeat steps 1-3 until the leaf level [128].

After applying the provided data to the first stage and using J48 as the second stage, the following performance results were obtained as shown in Table 5.4.

Table 5.4: J48 Performance Results

Name	**Value**
Number of Leaves	8
Size of the tree	14
Correctly classified	99.9965%
True Positive Rate	1.0
False Positive Rate	0.0
F-Measure	1.0
Time to Build Model	1.5 (s)

Performance variables shown in Table 5.4 are described below:

- *Classification accuracy*: % of connections that are classified correctly
- *True positives*: proportion of instances predicted positive that are actually positive

- *False positives*: proportion of instances predicted positive but are actually negative
- *F-Measure*: measure of test's accuracy
- *Time to Build Model* – time it takes to build the model based on the BSM data

As observed from the results table, stage two was able to decrease the number of false positives (FP) to 0 and increase the accuracy results to 99.9965%. This is a significant improvement from stage one results.

Confusion Matrix is given in Table 5.5:

Table 5.5: J48 Confusion Matrix

Spoofing	Normal	Information Disclosure	Tempering	DOS	Classified as ←
95	0	0	0	0	Spoofing
0	111064	0	0	4	Normal
0	0	1648	0	0	Information Disclosure
0	0	0	1967	0	Tempering
0	0	0	0	220	DOS

- *Advantages*: J48 is an easy algorithm to implement and visualize. Since it is based on C4.5, it performs well in discrete data with more than two classes, which is also one of the main reasons why J48 is chosen as one of the candidate algorithms in stage two. In addition, computational requirements for decision making are low compared with other algorithms used in this dissertation. Looking at the literature survey available, J48 is used commonly in medical and clinical applications, weather prediction, and banking data.
- *Disadvantages*: When looking at the training computational requirements, generally J48 takes more time and memory to be trained. If a J48 decision tree is not able to be configured properly, it results in a large tree, and the algorithm denigrates easy. If J48

outputs a complex tree, it gives off a poor performance and requires high computational power. That is why it is recommended to apply tree pruning, which helps with complexity and sometimes avoids overfitting and other classification errors. J48 and decisions trees, in general, have limits when dealing with continuous data, or decisions which require more than one output per attribute

5.4.2.2. Random Forest

Random Forest uses an ensemble learning method to combine decision trees, similar to those explained previously. This algorithm is similar to a technique known as bagging. Bagging is a machine learning ensemble meta-algorithm aimed at improving accuracy, reducing variance, and avoiding over-fitting. In a single decision tree, the predictions are sensitive due to certain data characteristics or noise. Bagging takes the average performance of multiple trees, so it eliminates such sensitivity and gives a more accurate performance. Random Forest improves bagging with a multitude of decision trees [129]. The mode output among the decision trees is the output of the Random Forest. Table 5.6 provides the IDS performance results while using Random Forest in the second stage.

Table 5.6: Random Forest Performance Results

Name	Value
Number of Iterations	100
Correctly classified	99.9983%
True Positive Rate	1.000
False Positive Rate	0.000
F-Measure	1.000
Time to Build Model	31.27s

Table 5.7: Random Forest Confusion Matrix

Spoofing	Normal	Information Disclosure	Tempering	DOS	Classified as ←
95	0	0	0	0	Spoofing
0	111066	0	0	2	Normal
0	0	1648	0	0	Information Disclosure
0	0	0	1967	0	Tempering
0	0	0	0	220	DOS

Only 2 BSM messages are misclassified as an attack in this algorithm.

- *Advantages:* Random Forest does a great job at correctly classifying and identifying malicious attacks from the data with only two normal packages misclassified as an attack. Although the dataset is large, the results show that errors are countable, and the accuracy is improved to 99.9983%. Based on literature survey, Random Forest is widely known to be one of the most accurate learning algorithms, and that is also the reason why it was selected as one of the algorithm candidates for stage two [129]. Random Forest in the second stage gives the best accuracy in this dissertation. This accuracy does not denigrate even with a large dataset or even at times when a large portion of the messages are missing. Once trained, this algorithm can also be reused in other models with similar data which is another feature making the Random Forest a predominant algorithm to be used for classification problems. Random Forest is commonly used in areas of medicine, e-commerce, and stock market application.
- *Disadvantages*: A high performance in Random Forest also comes with an increase in computational cost. A small improvement over J48 comes with 20 times longer time to build. Another disadvantage of Random Forest is the fact that it is hard to interpret it. In

addition, a careful analysis is needed in deciding its configuration parameters according to the dataset used as otherwise the performance accuracy will suffer.

5.4.2.3. AdaBoost

Adaptive Boosting is another ensemble of machine learning algorithm developed by Yoav Freund and Robert Schapiro[128]. In Adaptive Boosting (AdaBoost), the ensemble is built in such a way that prediction errors are improved at every layer. The subsequent models focus on fixing errors made by prior models. This is similar to regular boosting algorithms. Adaptive Boosting adds short decision trees in series until the performance is not subsequently improved. Performance results for IDS with Adaptive Boosting are shown in Table 5.8 and Table 5.9

Table 5.8: AdaBoost Performance Results

Name	Value
Correctly classified	98.293%
True Positive Rate	0.983
False Positive Rate	0.459
F-Measure	0.974
Time to Build Model	45.65s

Table 5.9: AdaBoost Confusion Matrix

Spoofing	Normal	Information Disclosure	Tempering	DOS	Classified as ←
0	0	0	95	0	Spoofing
0	111068	0	0	0	Normal
0	0	1648	0	0	Information Disclosure
0	0	0	1967	0	Tempering
0	220	0	0	0	DOS

- *Advantages*: AdaBoost is not a complicated algorithm to implement. Generally, the algorithm is known to give a good generalization and is used in many classification problems. AdaBoost is usually not prone to over-fitting due to its "boosting" technique. This algorithm is used in many classification problems, and it is known to improve

classification errors through boosting. In our case, AdaBoost does not give the best performance in terms of accuracy. Its implementation efficiency and over-fitting avoidance are the reasons why AdaBoost was selected as the third algorithm candidate. There are a few papers which have evaluated the use of AdaBoost in different applications such as [130], [131] and consider it as one of the best "off the shelf" algorithms. Applications where AdaBoost has been implemented successfully mainly focus on optical character recognition, pedestrian detection systems, speech, and facial recognition, etc.

- *Disadvantages*: One of the main disadvantages of AdaBoost is its sensitivity due to noise in the dataset and potential outliers. This property is also shown when applied to our data. When additional unknown attacks or outlier packages are injected, AdaBoost suffers in their classification. As it will be analyzed below, this algorithm is not the most optimal solution for the given problem, this is also often the case for other complex classification problems as well.

5.4.2.4. Naive Bayes

Naive Bayes is another algorithm selected for use in the second stage of the intrusion detection system. This algorithm uses a probabilistic classification and is based on Bayes theorem. The objective of this algorithm is to determine the probability of the features happening in every class and return the highest probable class.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \qquad (32)$$

Equation (32) calculates the probability of an event (A) considering the prior probability of conditions (B) that might be related to the event (A).

Table 5.10 and Table 5.11 provide the performance results of the IDS with Naive Bayes as the algorithm in the second stage

Table 5.10: Naive Bayes Performance Results

Name	**Value**
Correctly classified	99.9878%
True Positive Rate	1.000
False Positive Rate	0.003
F-Measure	1.000
Time to Build Model	0.3s

Table 5.11: Naive Bayes Confusion Matrix

Spoofing	Normal	Information Disclosure	Tempering	DOS	Classified as ←
95	0	0	0	0	Spoofing
0	111065	0	0	3	Normal
0	3	1645	0	0	Information Disclosure
0	8	0	1959	0	Tempering
0	0	0	0	220	DOS

- *Advantages*: Naive Bayes is primarily based on the conditional independence assumption as shown by equation (32). When the dataset holds the conditional independence assumption as true, then the algorithm converges quickly, making it more efficient than other logistic regression algorithms. In this scenario, the algorithm can also be trained with less data. One of the most common applications that uses Naive Bayes is email spam detection and news categorization. Its ability to infer based on the independence assumption is also the reason why Naive Bayes was selected in this dissertation. This property makes this algorithm to be suitable with other applications such as sentiment analysis, digit recognition, etc.

- *Disadvantages*: Naive Bayes is an algorithm that does not perform well in datasets where features are not independent of each other. This can be observed when some of the messages were not independent, and Naive Bayes failed, just in those instances.

Chapter 6. Performance Evaluation and Conclusions

This dissertation proposes a three-step framework. First, through the threat analysis, this work was able to identify and analyze vehicle communication threats within V2X. During a normal cybersecurity analysis process, the first step is to perform threat analysis and determine the landscape. The second step described in chapter IV assesses such threats using a new proposed model and assigns them a scale from E0 to E3 accordingly. Lastly, Chapter V proposes a new intrusion detection system which can detect threats using a BSM dataset.

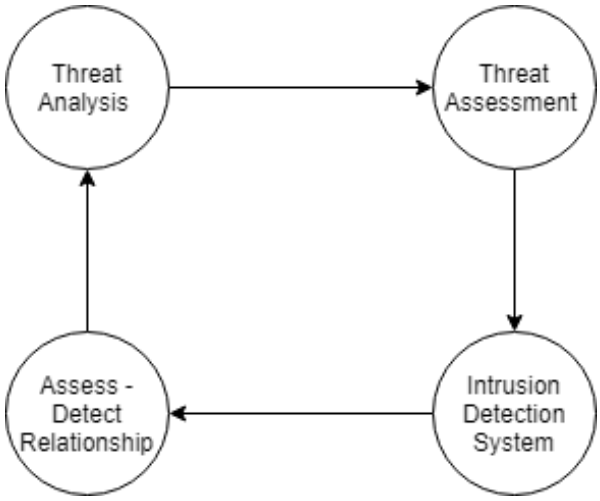

Figure 6.1: Complete Framework

This framework defines a continuous process as shown in Figure 6.1. This process analyzes threats, assesses them, detects threats, determines the assess-detect correlation to select the best algorithm for protection, and continues analyzing threats to restart the process all over again.

This chapter is organized as follows: Section 6.1 describes the correlation between Assess and Detect; 6.2 analyzes the performance of FTAM and IDS; 6.3 gives summary and conclusions; and 6.4 closes with recommendations for future work.

6.1. Assess and Detect – Their Correlation

In order to complete this framework, results of FTAM are correlated with the IDS results in order to establish a relationship between the different algorithms used in IDS and the output of FTAM. Table 6.1 essentially shows how each of the algorithms performs against Spoofing, Tempering, Information Disclosure, and DoS. For the purpose of this step, both Decision Tree-based algorithms are considered in one column since their performances are similar.

Table 6.1: Best Performing IDS algorithms for each FTAM level

STERIDE	FTAM Level	Decision Trees	AdaBoost	NaiveBayes
Information Disclosure	E1	✓	✓	×
Spoofing	E2	✓	×	✓
Tempering	E3	✓	✓	×
DoS	E3	✓	×	✓

Based on the preliminary results of this dissertation as shown in Table 6.1 and Figure 6.2 this dissertation establishes the following hypothesis:

"There is a correlative relationship between the Fuzzy-Based Threat Model levels and the machine learning algorithms used in stage two of the Intrusion Detection System."

Table 6.2: Assess-Detect Correlation

FTAM Level	IDS Algorithm
E0	All
E1	Decision Trees + AdaBoost
E2	Decision Trees + Naïve Bayes
E3	Decision Trees + AdaBoost

From the literature review, this is the first time such a concept of correlation is introduced. This relationship is the basis for another research study. To be fully proved out, it needs further research, more validation, a robust and standard dataset, and testing more algorithms. For the purpose of this paragraph, it proves out the concept that there is a correlation between the two, and this relationship can prove out to be useful in protecting from threats with a certain assessment level.

6.2. Performance Summary

Subsections below provide a performance summary for all of the elements in this dissertation.

6.2.1. FTAM Performance Benchmarking and Analysis

To validate FTAM, its performance is analyzed by benchmarking results with other known, established, and accepted models. In order to test this model, some generic attacks are defined first. These attacks are defined according to the STRIDE model, and a V2X environment is used in their definition. The list below provides some quick definitions for the attacks used.

- Spoofing (S) – The objective of a spoofing attack is to bypass the authentication mechanism by spoofing the sender's ID. In a V2X environment, this is done by masquerading certain BSM data, so that it looks like the messages are coming from a certain vehicle.
- Tempering (T) – This threat attempts to temper values of data in a memory location or temper BSM data. This is often done by tempering BSM messages in order to allow data to pass through.

- Repudiation (R) – This attack tries to replay an old message and resent it. Usually, this happens when the receiver does not verify the sender's authenticity or freshness. This is often done by replaying BSM messages repeatedly
- Information Disclosure (I) – This attack is realized by eavesdropping or trying to listen in the communication channel in order to intercept communication.
- Denial of Service (D) – This attack attempts to take down the V2X network by jamming the communication and resending a massive number of packages. This can be accomplished by sending a high amount of BSM messages in a short time to be processed by the network.
- Elevation of Privileges (E) – This attack happens when messages that are sent attempt to obtain higher privileges. For example, fake high priority messages which attempt to flash malicious software would consist of an E type of attack.

Based on this definition of STRIDE, threats are analyzed and run through FTAM to determine a threat level. Please note that in a normal implementation, their characterization would depend on internal organization assessment. For this exercise, data is derived from the levels on some of the examples shown in [13] [66] [112] [106] [119] and [111]. Since FTAM characterizations are aligned with EVITA and others, it is easy to extract the inputs for validation. Last column on

Table 6.3 below shows the results of the FTAM for each of the threats defined above.

Table 6.3: FTAM - STRIDE Threat Levels

Threat	Attack Impact			Attacker Capability			Withstand Potential		TL
	Privacy	**Safety**	**Fin Loss**	**Expertise**	**Resources**	**F&O gain**	**Controllability**	**Difficulty**	
S	1	1	2	2	2	2	1	1	E2
T	1	3	3	3	2	3	3	2	E3
R	0	1	2	2	1	2	1	2	E2
I	1	0	1	1	0	1	0	2	E1
D	1	3	3	3	2	2	3	2	E3
E	1	2	2	3	2	2	1	3	E2

As observed, threat levels in this scenario range from E1 to E3. In this case, there is no E0 level which is a low threat. To verify that FTAM produces accurate levels, these same threats are run with the same characteristics through two other accepted models (EVITA and HEAVENS) to compare and benchmark. This test is done to compare FTAM with the other models based on the same data. Results given in Table 6.4 show an accurate proportional relationship between FTAM, EVITA, and HEAVENS.

Table 6.4: STRIDE Threat Levels for FTAM, EVITA, and HEAVENS

Threat	FTAM	EVITA	HEAVENS (TL Score)	HEAVENS (TL Level)
S	E2	R4	6	Medium
T	E3	R5	2	High
R	E2	R3	5	Medium
I	E1	R2	8	Low
D	E3	R6	1	Critical
E	E2	R3	3	High

6.2.2. FTAM Advantage

Results in Table 6.4 and additional experimentation are used to build a relationship between FTAM, EVITA, HEAVENS, and functional safety standards (EAL, SIL, and ASIL). These results are shown in Table 6.5. At the beginning of this dissertation, a set of reasons was provided on why a new Threat Assessment model was proposed. The bullet points below

reiterate those reasons while explaining how FTAM has closed those gaps and has an advantage compared with other threats models.

- FTAM is designed with safety in mind. Through the knowledge embedded into FIS, the important elements such as safety, privacy are always protected, and their importance is reflected at the final Threat Level output.
- Compared with EVITA, NHTSA and HEAVENS, FTAM does not require a significant effort to assess a threat. The final integration is offered as a package and organizations can configure it for their own use. In addition, FTAM does not use a subjective assessment; therefore, it does not have any issues with ambiguous assessments like some of the other models.
- As mentioned previously, SAE J3061 allows and encourages individual organizations to select their threat assessment of choice. FTAM is designed with flexibility in mind. This framework can be adapted according to organizational needs and adapted quickly. FTAM allows that organizations can also define their own risk acceptance levels. This is an advantage compared to other models which have embedded table lookups or rigid assessments.
- Most importantly, FTAM is not subjective in its assessment. It uses a clearly defined Fuzzy architecture to perform threat assessment. Fuzzy Logic in FTAM can deal with subjective notations or characterizations.
- Other threat assessments often result in inconsistent categorization. For example: Threats 35 and 54 presented in [119] are both Denial of Service attacks (D type of attacks). FTAM classifies both of these threats as E3 ,consistent with its categorization in STRIDE where D is categorized as E3. EVITA categorizes threat 35 as R3 and HEAVENS

categorize it as Medium. While for Threat 54, EVITA categorizes it as R2 and HEAVENS as "Low." Both of these threats are Denial of Service and assuming that its characteristics are the same they should be categorized as Medium from HEAVENS or R3/R4 from EVITA, which is not the case. Meanwhile, FTAM is consistent and categorizes them both as E3.

Based on these experimental results, Table 6.5 determines the relationship between FTAM, EVITA, HEAVENS, and some of the functional safety standards.

Table 6.5: FTAM Levels

FTAM	EVITA	HEAVENS	EAL	SIL	ASIL
E0	R0	None	0	N/A	QM
E1	R1	Low	1	1	A
E1	R2	Low	2	1	A
E2	R3	Medium	3	2	B
E2	R4	Medium	4	2	B
E3	R5	High	5	3	C
E3	R6	Critical	6	3	D
E3	R7	Critical	7	4	N/A
N/A	R7+	Risk beyond acceptable levels			

6.2.3. IDS Performance Evaluation

Performance evaluation tables for the individual algorithms in the proposed Intrusion Detection System are given in 5.4. Figure 6.2 shows the performance summary results from the four algorithms tested in the second stage

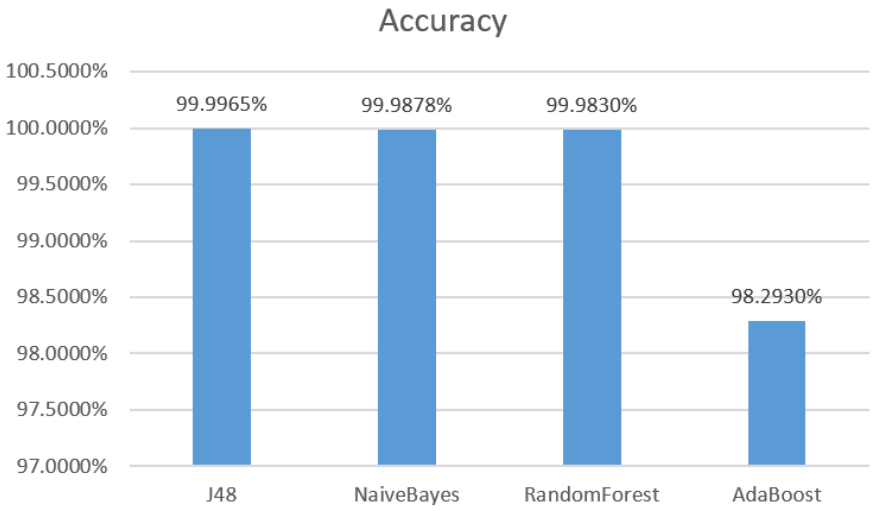

Figure 6.2: Attacks Correctly Classified

As shown, the J48 (99.9965%) and Naive Bayes (99.9878%) have the best classification accuracies when it comes to performance. Both of these algorithms—assisted by FarthestFirst in stage one—are able to eliminate false positives. As mentioned previously, false positives are a syndrome of anomaly based intrusions, and this proposed IDS architecture in this dissertation is able to cope with them in an easy and efficient way

6.2.4. IDS Computational Performance

In order to evaluate the performance of the algorithms, the computational time required to train and test the algorithms is used. The models were built and tested using a Windows HP Desktop with an Intel Core i7-6700 CPU @ 3.40 GHz. Table 6.6 shows computational times versus classification performance.

Table 6.6: Summarized Results

Algorithm	Classification Performance	Time to train
J48	**99.9965%**	1.5 (s)
NaiveBayes	99.9878%	**0.3 (s)**
Random Forest	99.9830%	31.27 (s)
AdaBoost	98.2930%	45.65 (s)

When it comes to classification performance, J48 is the best algorithm to perform. Looking at the computational time required, Naive Bayes delivers similar performance results (within 0.01% of J48) but with 5 times better computational time requirement. The limited data shows that there is a correlation between the threat level and the computational time required, the higher the threat level, more computational time is required to detect. This relationship is subjective considering the limited data points considered in this dissertation.

6.2.5. State-of-the-art Comparison

Another additional factor in IDS performance evaluation is to compare it with state-of-the-art. Unfortunately, due to using a newly released dataset, there is no published work available with the same dataset to make an “apple-to-apple” comparison for the performances. Due to this factor, the same algorithms with the same architecture were tested on a standard Knowledge Discovery Dataset. This dataset is one of the most widely used datasets in the evaluation of Intrusion Detection Systems. It was built by the Massachusetts Institute of Technology (MIT) and used in the International Knowledge Discovery and Data Mining Tool Competition [132].

Table 6.7: KDD IDS Performance Results

Research Paper	**Algorithm**	**Accuracy (%)**	**F-Measure**
2017: Meena, Choudhary [133]	Bayes	92.72	0.916
2017: Meena, Choudhary [133]	J48	99.45	0.993
2016: Subba et al. [134]	C4.5 (best)	98.74	Nor reported
2017: Kushwaha et al. [86]	SVM	99.63	0.99
Proposed Algorithm	**Two Stage**	**99.95**	**0.999**

As it can be observed from Table 6.7, the proposed IDS in this dissertation, used in another dataset outperforms other proposed IDS in that same dataset. Similar pre-processing was used in this experiment as well. These results are published in [26]

6.3. Summary

This dissertation provided a three-step framework for V2X cybersecurity using Machine Learning methods. Chapter I gives an introduction of the area and makes the case for the need of this research. Chapter II focuses on the literature survey and reviews some of the main aspects worked on this dissertation. Chapter III is the first step of the framework and does a threat analyzation based on the STRIDE framework[27].

In chapter IV, the case for automotive threat assessment models and their role in defining cybersecurity requirements is presented first. In addition, this chapter does analysis for some of the existing threat assessment models. Based on this analysis, a new, innovative Fuzzy based threat assessment model (FTAM) is proposed. The use of Fuzzy Logic in threat assessment makes it possible to mitigate or eliminate the drawbacks identified from the other existing models and improve the threat assessment process. This system is the first of its type and is designed based on some elements from existing models such as EVITA[13], NHTSA[12], HEAVENS[106], UM[114] and OCTAVE[107]. The Fuzzy Logic makes FTAM a flexible

framework which is able to produce threat levels based on different assessments with no ambiguous characterizations. FTAM uses six Fuzzy inference systems in a multistage architecture similar to [28]. This proposed model was also benchmarked against EVITA and HEAVENS for validation purposes. Based on this benchmark, a level relationship between FTAM, EVITA, HEAVENS, and other functional safety standards such as EAL, SIL, ASIL was established. The results of FTAM are used to drive the design of the next Detect phase.

The Detect phase is described in Chapter V. This chapter initially provides a background in intrusion detection systems and their importance in the cybersecurity space. Then the dataset from Wyoming DOT Connected Vehicle Pilot is used and analyzed[17]. This dissertation is the first to publish IDS results on this dataset. Before usage, the data is pre-processed using variance, correlation coefficients, Least Square Regression Error, and Maximal Information Compression Index. Pre-processing the data is necessary to reduce the number of features and help with avoiding bias and overfitting. After such feature engineering, the data is used to build an intelligent, two-stage intrusion detection system. The first stage uses the FurthestFirst unsupervised learning algorithm to detect an attack while the second stage tests four different supervised learning algorithms (J48, Random Forest, Nave Bayes, and Adaptive Boosting) to classify the attacks. After building and testing the two-stage model, a high accuracy in performance results was achieved. In addition, the proposed IDS was able to fully eliminate the number of false positives which are usually a syndrome of anomaly-based intrusion detection systems.

The last chapter in this dissertation does a performance evaluation of FTAM, IDS, and their relationship. From an FTAM perspective, the model is compared against other models using the

same threats and advantages of this work are given. From an IDS perspective, the performance results are analyzed regarding accuracy and computational requirements. This chapter also sets the ground for a relationship between the detect and assess steps. Due to being the first study using such methodology or dataset, there are some limitations on the state-of-the-art comparison, but some comparisons and benchmarking with other works is given.

6.4. Conclusions and Future Work Recommendations

Based on the results obtained from this dissertation, below is a list of conclusions and future work recommendations.

- Fuzzy Logic improves threat assessment processes and can cope with limitations when problems include linguistic variables. This method should be used in other risk or threat assessment models to build flexible frameworks.
- Threat analyzation is an important and required step in understanding and solving cybersecurity problems. Using commonly known, standardized characterizations and open source frameworks is recommended to cope with cybersecurity challenges.
- Threat characterizations should be standardized, but the methodologies on how you determine the degree of the threat should be flexible enough to allow for needed customizations.
- A multistage architecture in Fuzzy Logic reduces the number of rules required to build the system and enables a stronger relationship between the layers for a more manageable framework.
- There is a correlation or relationship between the threat assessment methods and methods used to detect threats or protect from them. Further work needs to be done in establishing or proving out this relationship.

- Cybersecurity solutions should evaluate and approach problems holistically rather than in individual silos. Intruders' ultimate objective is to "hack" the system, not its individual components.
- The proposed two-stage architecture which combines unsupervised and supervised learning algorithm produces better accuracy, lowers the number of false positives, and improves computational requirements. This architecture can be used in a variety of machine learning and artificial intelligence problems.
- Feature engineering focused on dimensionality reduction is a recommended step to reduce bias and improve performance results for classical machine learning algorithms.
- Algorithm selection in machine learning or data science should be based on factors such as computational requirements, dataset characteristics, type of problems and others rather then generalized performances or stereotypes for certain algorithms.
- Additional research on other standardized datasets are needed to build and analyze the use of intrusion detection systems using BSM messages.

APPENDIX

```
%% MATLAB script to instantiate the FIS models
clear;
FIS1 = readfis('FIS1PrivSec_v2');
FIS2 = readfis('FIS2ExpertiseResources');
FIS3 = readfis('FIS3SecFinloss_V2');
FIS4 = readfis('FIS4FinGainAgentLevel');
FIS5 = readfis('FIS5DiffControl');

%% create final FIS6 from the FIS1-5
FIS6 = mamfis;
FIS6 = mamfis("NumInputs",3,"NumOutputs",1);
FIS6 = mamfis("NumInputs",3);
FIS6.Inputs(1)=FIS3.Output(1);
FIS6.Inputs(2)=FIS4.Output(1);
FIS6.Inputs(3)=FIS5.Output(1);
FIS6 = addOutput(FIS6, [0 3],'NumMFs' ,4,'MFType',"gaussmf");
FIS6.Outputs(1).Name = "ThreatLevel";
fuzzyLogicDesigner(FIS6)

%% initialize FUZZY constants as needed
set_param('SimulinkModelV3/Privacy','Value','1')
set_param('SimulinkModelV3/Safety','Value','2')
set_param('SimulinkModelV3/Financial Loss','Value','2')
set_param('SimulinkModelV3/Expertise','Value','2')
set_param('SimulinkModelV3/Resources','Value','3')
set_param('SimulinkModelV3/Financial Gain','Value','2')
set_param('SimulinkModelV3/Controllability','Value','1')
set_param('SimulinkModelV3/Difficulty','Value','3')
```

BIBLIOGRAPHY

[1] K. Bimbraw, "Autonomous Cars : Past , Present and Future," *2015 12th Int. Conf. Informatics Control. Autom. Robot.*, pp. 191–198, 2015.

[2] K. Abboud, H. A. Omar, and W. Zhuang, "Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9457–9470, 2016.

[3] K. Hartman, "Wyoming Connected Vehicle Pilot Deployment Program," *DOT*, 2018. [Online]. Available: https://www.its.dot.gov/pilots/pilots_overview.htm.

[4] K. Farrell, "The rapid urban growth Triad: A new conceptual framework for examining the urban transition in developing countries," *Sustain.*, vol. 9, no. 8, pp. 1–19, 2017.

[5] Atkins, "Autonomous Vehicles," p. 2015, 2015.

[6] C. Yan, W. Xu, J. Liu, and Q. 360, "Can You Trust Autonomous Vehicles: Contactless Attacks against Sensors of Self-Driving Vehicles," *Def Con 24*, 2016.

[7] D. K. Nilsson, U. E. Larson, F. Picasso, and E. Jonsson, "A first simulation of attacks in the automotive network communications protocol flexRay," *Adv. Soft Comput.*, vol. 53, pp. 84–91, 2009.

[8] S. Checkoway *et al.*, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," *System*, pp. 6–6, 2011.

[9] C. Miller and C. Valasek, "A Survey of Remote Automotive Attack Surfaces," *Defcon 22*, pp. 1–90, 2014.

[10] C. Valasek and C. Miller, "Adventures in Automotive Networks and Control Units," *Tech. White Pap.*, p. 99, 2013.

[11] N. H. T. S. Administration, "Cybersecurity Best Practices for Modern Vehicles," p. 22p, 2016.

[12] C. McCarthy, K. Harnett, and A. Carter, "Characterization of Potential Security Threats in Modern Automobiles A Composite Modeling Approach," *Nhtsa*, no. September, 2014.

[13] H. Seudié, "Vehicular On-board Security : EVITA Project Project," *Forum Am. Bar Assoc.*, no. November, 2009.

[14] P. Jin, M. Walton, G. Zhang, J. Xiaowen, and A. Singh, "No Title," in *Analyzing the Impact of False-Accident Cyber Attacks on Traffic Flow Stability in Connected Vehicle Environment*, 2013.

[15] Daniel J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles:

opportunities, barriers and policy recommendations," *Transp. Res. Part A Policy Pract.*, vol. 77, pp. 167–181, 2015.

[16] T. Bryan, "Proposed rule would mandate vehicle-to-vehicle (V2V) communication on light vehicles, allowing cars to 'talk' to each other to avoid crashes," *NHTSA*, 2016.

[17] P. Overview, S. Stories, and F. Links, "Wyoming (WY) DOT Pilot," pp. 3–5, 2018.

[18] C. Laurendeau, C. Laurendeau, M. Barbeau, and M. Barbeau, "Threats to Security in DSRC/WAVE," *Security*, vol. 4104, p. 266, 2006.

[19] A. Shaout, N. Kaja, and S. Awad, "A smart traffic sign recognition system," in *11th International Computer Engineering Conference (ICENCO)*, 2015, pp. 157–162.

[20] N. Kaja, A. Shaout, and O. Dehzangi, "Two Stage Intelligent Automotive System to Detect and Classify a Traffic Light," in *International Conference on New Trends in Computing Sciences, ICTCS 2017*, 2017, pp. 30–35.

[21] N. Kaja, A. Shaout, and D. Ma, "A Two-Stage Intrusion Detection Intelligent System," ACIT 2017

[22] N. Kaja, A. Nasser, D. Ma, and A. Shaout, "Automotive Security," in *Encyclopedia of Wireless Networks*, 2019, pp. 1–6.

[23] L. Zhang, N. Kaja, D. Ma, and L. Shi, "A Two Stage Deep Learning Approach for CAN Intrusion Detection," in *NDIA Ground Vehicle Systems Engineering and Technology Symposium*, 2018.

[24] A. Shaout, N. Kaja, and M. Borovikov, "Security Solution for Cloud Computing Using a Hardware Implementation of AES," in *The International Arab Conference on Information Technology (ACIT2014)*, 2014, pp. 57–64.

[25] N. Kaja, A. Shaout, and D. Ma, "Fuzzy Based Threat Assessment Model," *J. Appl. Intell.*, pp. 1–23, 2019 (submitted).

[26] N. Kaja, A. Shaout, and D. Ma, "An Intelligent Intrusion Detection System," *J. Appl. Intell.*, pp. 0–9, 2019.

[27] Microsoft, "The STRIDE Threat Model," *Web*, 2009. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20).

[28] A. Shaout and J. Trivedi, "Performance Appraisal System-Using a Multistage Fuzzy Architecture," vol. 02, no. 03, pp. 405–411, 2013.

[29] O. Cordón, "A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: Designing interpretable genetic fuzzy systems," *Int. J. Approx. Reason.*, vol. 52, no. 6, pp. 894–913, 2011.

[30] C. H. Hyun, C. W. Park, and S. Kim, "Takagi-Sugeno fuzzy model based indirect adaptive fuzzy observer and controller design," *Inf. Sci. (Ny).*, vol. 180, no. 11, pp. 2314–2327, 2010.

[31] L. Zhang, N. Kaja, L. Shi, and D. Ma, "A Two-Stage Deep Learning Approach for Can

Intrusion Detection System," in *NDIA Ground Vehicle Systems Engineering and Technology Symposium*, 2018.

[32] E. Eckermann, *World History of the Automobile*. .

[33] S. D. A. Museum, "Automotive History." .

[34] A. Ismail and W. Jung, "Research trends in automotive functional safety," *QR2MSE 2013 - Proc. 2013 Int. Conf. Qual. Reliab. Risk, Maintenance, Saf. Eng.*, pp. 1–4, 2013.

[35] H. Ford, "Opening the Highways to All Mankind.".

[36] R. Viereckl, D. Ahlemann, A. Koster, and S. Jursch, *Racing Ahead with Autonomous Cars and Digital Innovation*, vol. 4, no. 12. 2015.

[37] Daniel J. Fagnant and K. Kockelman, "Transportation Research Part A 77: 167-181, 2015.," pp. 1–20, 2015.

[38] "Carnagie Melon | The Robotics Institute." [Online]. Available: https://www.ri.cmu.edu/.

[39] E. Auto, "Nissan, Tsinghua University opens Joint Research Center for Intelligent Mobility," *The Economic Times*, 2016.

[40] "Center for Automotive Research at Stanford." [Online]. Available: https://cars.stanford.edu/.

[41] "Mcity Test Facility." [Online]. Available: https://mcity.umich.edu/.

[42] "MIT Media Lab." [Online]. Available: https://www.media.mit.edu/.

[43] "Berkeley - DeepDrive." [Online]. Available: https://deepdrive.berkeley.edu/.

[44] L. Of, D. Automation, and A. R. E. Defined, "SAE Six Levels of Automation," 2014.

[45] Audi, "Audi Piloted driving." [Online]. Available: https://media.audiusa.com/models/piloted-driving.

[46] J. Stewart, "Tesla's Self Driving car plan seems insane, but it just might work," *Wired*.

[47] D. Etherington, "Ford outlines plan to build self-driving cars at scale to deploy with partners," *Techcrunch*.

[48] N. Bomey, "Daimler's Mercedes, Bosch to deliver self-driving car by 2021," *USA Today*.

[49] J. Walker, "The Self-Driving Car Timeline – Predictions from the Top 11 Global Automakers," *Tech emergence*, 2018.

[50] "Volvo Cars and Autoliv team up with NVIDIA to develop advanced systems for self-driving cars," *Volvo*.

[51] S. Byrford, "Honda reveals its plans for autonomous vehicles," *The Verge*.

[52] V. L. L. Thing and J. Wu, "Autonomous Vehicle Security: A Taxonomy of Attacks and Defences," *Proc. - 2016 IEEE Int. Conf. Internet Things; IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-Smart*

Data 2016, pp. 164–170, 2017.

[53] T. Armerding, “The 18 biggest data breaches of the 21st century,” *Cso*, pp. 1–14, 2018.

[54] O. August *et al.*, “Five Star Automotive Cyber Safety Framework,” *I Am Cavalry*, no. February, pp. 1–5, 2015.

[55] A. Hafeez, H. Malik, O. Avatefipour, P. R. Rongali, and S. Zehra, “Comparative Study of CAN-Bus and FlexRay Protocols for In-Vehicle Communication,” *SAE Tech. Pap.*, 2017.

[56] O. Avatefipour, A. Hafeez, M. Tayyab, and H. Malik, “Linking received packet to the transmitter through physical-fingerprinting of controller area network,” in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, 2017.

[57] M. Tayyab, A. Hafeez, and H. Malik, “Spoofing Attack on Clock Based Intrusion Detection System in Controller Area Networks,” *2018 Gr. Veh. Syst. Eng. Technol. Symp.*, 2018.

[58] A. Hafeez, M. Tayyab, C. Zolo, and S. Awad, “Finger Printing of Engine Control Units by Using Frequency Response for Secure In-Vehicle Communication,” in *2018 14th International Computer Engineering Conference*, 2018.

[59] E. Yağdereli, C. Gemci, and A. Z. Aktaş, “A study on cybersecurity of autonomous and unmanned vehicles,” *J. Def. Model. Simul.*, vol. 12, no. 4, pp. 369–381, 2015.

[60] J. Fahey, “How to hack your car?,” *Forbes*, 2002.

[61] M. Herfurt, “Car Whisperer,” *Trifinite*, 2005.

[62] J. Copping, “Hackers can take over car navigation system,” *The Telegraph*, 2007.

[63] K. Koscher *et al.*, “Experimental security analysis of a modern automobile,” *Proc. - IEEE Symp. Secur. Priv.*, pp. 447–462, 2010.

[64] C. Miller and C. Valasek, “CAN Message Injection,” 2016.

[65] “Car Hacking Research: Remote Attack Tesla Motors,” *Keen Security Lab of Tencent*, 2016. .

[66] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl, “Security requirements for automotive on-board networks,” *2009 9th Int. Conf. Intell. Transp. Syst. Telecommun. ITST 2009*, pp. 641–646, 2009.

[67] M. Staron, “Automotive Software Architectures,” *Automot. Softw. Archit.*, pp. 33–39, 2017.

[68] C. Bordonali, S. Ferraresi, and W. Richter, “Shifting gear s in cyber security for connected cars,” no. February, 2017.

[69] Y. L. Morgan, “Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics,” *IEEE Commun. Surv. Tutorials*, vol. 12, no. 4, pp. 504–518, 2010.

[70] J. B. Kenney, “Dedicated short-range communications (DSRC) standards in the United States,” *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[71] D. O. T. Hs, “Traffic Safety Facts 2016,” vol. 2018, no. March, pp. 1–10, 2018.

[72] Institute of Electical and Electronics Engineers, *IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages (IEEE Std 1609.2-2013)*, vol. 2013, no. April. 2013.

[73] D. Crevier, *AI*..

[74] A. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Commun. Surv. Tutorials*, vol. PP, no. 99, p. 1, 2015.

[75] J. Yen and R. Langari, *Fuzzy Logic : Intelligence, Control and Information*.

[76] E. . Mamdani, “APPLICATION OF FUZZY LOGIC TO APPROXIMATE REASONING USING LINGUISTIC SYNTHESIS,” vol. 91, 2017.

[77] E. Chan, H. Zhu, and W. Bazzi, “Fuzzy Logic and Probability theory,” pp. 1–7.

[78] C. C. Lee, “Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II.”

[79] S. Clarence, *Intelligent Control: Fuzzy Logic Applications (Mechatronics)*.

[80] A. Fernández, C. J. Carmona, M. J. del Jesus, and F. Herrera, “A View on Fuzzy Systems for Big Data: Progress and Opportunities,” *Int. J. Comput. Intell. Syst.*, vol. 9, pp. 69–80, 2016.

[81] F. Herrera, “Fuzzy Systems in Data Science and Big Data,” *8th Int. Work. Eval. Inf. Access (EVIA 2017)*, pp. 0–16, 2017.

[82] M. Friedman and A. Kandel, *Introductino to Pattern Recognition: Statistical, Structural, Neural and Fuzzy Logic Approaches.*

[83] S. Al Amro, F. Chiclana, and D. Elizondo, “Application of Fuzzy Logic in Computer Security and Forensics,” in *Computational Intelligence for Privacy and Security*, pp. 43–57.

[84] O. Al-Jarrah and A. Arafat, “Network Intrusion Detection System Using Neural Network Classification of Attack Behavior,” *J. Adv. Inf. Technol.*, vol. 6, no. 1, pp. 1–8, 2015.

[85] S. Ganapathy, K. Kulothungan, P. Yogesh, and A. Kannan, “An Intelligent Intrusion Detection System for Ad Hoc,” pp. 430–434, 2012.

[86] P. Kushwaha, H. Buckchash, and B. Raman, “Anomaly Based Intrusion Detection Using Filter Based Feature Selection on KDD-CUP 99,” pp. 839–844, 2017.

[87] N. SALMAN and M. BRESCH, “Design and implementation of an intrusion detection system (IDS) for in-vehicle networks,” *2007 IEEE Int. Conf. Electro/Information Technol.*, 2017.

[88] F. Li, L. Wang, and Y. Wu, “Research on CAN Network Security Aspects and Intrusion Detection Design,” *SAE Tech. Pap.*, vol. Part F1298, no. September, 2017.

[89] K. Hwang, M. Cai, Y. Chen, and M. Qin, “Hybrid intrusion detection with weighted

signature generation over anomalous internet episodes," *IEEE Trans. Dependable Secur. Comput.*, vol. 4, no. 1, pp. 41–55, 2007.

[90] P. Processing and M. Database, "Innovative Signature Based Intrusion Detection System," pp. 114–119, 2017.

[91] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 1577–1583, 2017.

[92] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 1110–1115, 2011.

[93] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," *2016 IEEE 2nd Int. Forum Res. Technol. Soc. Ind. Leveraging a Better Tomorrow, RTSI 2016*, pp. 0–5, 2016.

[94] Z. Li, W. Sun, and L. Wang, "A neural network based distributed intrusion detection system on cloud platform," *2012 IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst.*, vol. 1, no. May, pp. 75–79, 2012.

[95] G. Loukas, E. Karapistoli, E. Panaousis, P. Sarigiannidis, A. Bezemskij, and T. Vuong, "A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles," *Ad Hoc Networks*, vol. 84, pp. 124–147, 2019.

[96] K. M. A. Alheeti, A. Gruebler, and K. McDonald-Maier, "Using discriminant analysis to detect intrusions in external communication for self-driving vehicles," *Digit. Commun. Networks*, vol. 3, no. 3, pp. 180–187, 2017.

[97] K. Ali Alheeti, A. Gruebler, and K. McDonald-Maier, "Intelligent Intrusion Detection of Grey Hole and Rushing Attacks in Self-Driving Vehicular Networks," *Computers*, vol. 5, no. 3, p. 16, 2016.

[98] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. Hubaux, "Vehicular Networks," vol. 25, no. 8, pp. 1557–1568, 2007.

[99] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-Based Intrusion Detection for VANETs: A Statistical Approach to Rogue Node Detection," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6703–6714, 2016.

[100] C. A. Kerrache, N. Lagraa, C. T. Calafate, and A. Lakas, "TFDD: A trust-based framework for reliable data delivery and DoS defense in VANETs," *Veh. Commun.*, vol. 9, pp. 254–267, 2017.

[101] K. Verma, H. Hasbullah, and A. Kumar, "Prevention of DoS attacks in VANET," *Wirel. Pers. Commun.*, vol. 73, no. 1, pp. 95–126, 2013.

[102] K. Bian, G. Zhang, and L. Song, "Security in use cases of vehicle-to-everything communications," *IEEE Veh. Technol. Conf.*, vol. 2017–Septe, pp. 1–5, 2018.

[103] W. Whyte, J. Petit, V. Kumar, J. Moring, and R. Roy, "Threat and Countermeasures Analysis for WAVE Service Advertisement," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, vol. 2015–Octob, pp. 1061–1068, 2015.

[104] S. Boumiza and R. Braham, “Intrusion threats and security solutions for autonomous vehicle networks,” *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2017–Octob, pp. 120–127, 2018.

[105] L. Bariah, D. Shehada, E. Salahat, and C. Y. Yeun, “Recent advances in VANET security: A survey,” *2015 IEEE 82nd Veh. Technol. Conf. VTC Fall 2015 - Proc.*, pp. 1–7, 2016.

[106] A. Lautenbach and M. Islam, “HEAling Vulnerabilities to Enhance Software Security and Safety,” *Secur. Model.*, no. March, 2016.

[107] D. Hosseini and K. Malamas, “Design Flaws as Security Threats,” 2017.

[108] N. Shevchenko, T. A. Chick, P. O’riordan, T. P. Scanlon, and C. Woody, “Threat Modeling: a Summary of Available Methods,” no. July, 2018.

[109] D. Ward, I. Ibarra, and A. Ruddle, “Threat Analysis and Risk Assessment in Automotive Cyber Security,” *SAE Int. J. Passeng. Cars - Electron. Electr. Syst.*, vol. 6, no. 2, pp. 2013-01-1415, 2013.

[110] Z. Ma and C. Schmittner, “Threat Modeling for Automotive Security Analysis 2 Secure Development of Automotive Systems.”

[111] G. Macher, E. Armengaud, E. Brenner, and C. Kreiner, “Threat and Risk Assessment Methodologies in the Automotive Domain,” *Procedia Comput. Sci.*, vol. 83, no. 1, pp. 1288–1294, 2016.

[112] G. Macher, E. Armengaud, E. Brenner, and C. Kreiner, *A Review of Threat Analysis and Risk Assessment Methods in the Automotive Context*. 2014.

[113] M. M. Islam, A. Lautenbach, C. Sandberg, and T. Olovsson, “A Risk Assessment Framework for Automotive Embedded Systems,” *ACM Int. Work. Cyber-Physical Syst. Secur.*, pp. 3–14, 2016.

[114] D. Dominic, S. Chhawri, R. M. Eustice, D. Ma, and A. Weimerskirch, “Risk Assessment for Cooperative Automated Driving,” *Proc. 2nd ACM Work. Cyber-Physical Syst. Secur. Priv. - CPS-SPC ’16*, pp. 47–58, 2016.

[115] V. C. S. E. Committee, “SAE J3061: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems,” 2016.

[116] P. Saitta, B. Larcom, and M. Eddington, “Trike v. 1 methodology document,” *URL http//dymaxion. org/trike/ …*, pp. 1–17, 2005.

[117] Microsoft, “ASF: Application Security Frame,” *Mictosoft*, 2010. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649461(v=pandp.10).

[118] “Federal Office for Information Security,” 2009.

[119] S. P. Kadhirvelan and A. Söderberg-Rivkin, “Threat Modelling and Risk Assessment Within Vehicular Systems,” *Chalmers Univ. Technol.*, no. August, p. 52, 2014.

[120] Common Criteria, “Common Criteria for Information Technology Security Evaluation Part 1 : Introduction and general model September 2012 Revision 4 Foreword,” *ISO/IEC*

15408 Common Criteria, Part 12012, no. September, 2012.

[121] T. Report, "ETSI - TR 102 893 - V1.1.1 - Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA)," *Intell. Transp. Syst.*, vol. 1, pp. 1–86, 2010.

[122] ISO 26262, "Functional Safety - Road Vehicles, International Organization for Standardization," 2011.

[123] F. Report, "SAE J2735 Standard : Applying the Systems Engineering Process," no. January, 2013.

[124] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional datasets with application to reference matching," *Proc. sixth ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '00*, pp. 169–178.

[125] P. L. Sanjoy Dasgupta, "Performance Guarantees for Hierarchical Clustering," *15th Annu. Conf. Comput. Learn. Theory*, no. July 2010, pp. 351–363, 2002.

[126] D. S. Hochbaum and D. B. Shmoys, "A Best Possible Heuristic for the *k* -Center Problem," *Math. Oper. Res.*, vol. 10, no. 2, pp. 180–184,.

[127] S. Sharmila and M. Kumar, "An optimized farthest first clustering algorithm," *2013 Nirma Univ. Int. Conf. Eng. NUiCONE 2013*, pp. 1–5, 2013.

[128] P. Pandey and R. Prabhakar, "An analysis of machine learning techniques (J48 & AdaBoost)-for classification," *India Int. Conf. Inf. Process. IICIP 2016 - Proc.*, pp. 1–6, 2017.

[129] Y. Dong, B. Du, and L. Zhang, "Target detection based on random forest metric learning," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 4, pp. 1830–1838, 2015.

[130] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees," *Mach. Learn.*, vol. 40, no. 2, pp. 139–157.

[131] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Mach. Learn.*, vol. 36, no. 1998, pp. 105–139.

[132] P. Aggarwal and S. K. Sharma, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection," *Procedia Comput. Sci.*, vol. 57, pp. 842–851, 2015.

[133] G. Meena and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA," *2017 Int. Conf. Comput. Commun. Electron. COMPTELIX 2017*, pp. 553–558, 2017.

[134] B. Subba, S. Biswas, and S. Karmakar, "A Neural Network based system for Intrusion Detection and attack classification," *2016 22nd Natl. Conf. Commun. NCC 2016*, pp. 1–6, 2016.