Multilayered Framework for Securing Connected Autonomous Vehicle

by

Rafi Ud Daula Refat

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical, Electronics and Computer Engineering)
in the University of Michigan - Dearborn
2023

Doctoral Committee:

Professor Hafiz Malik, Chair
Associate Professor Anys Bacha
Assistant Professor Alireza Mohammadi
Associate Professor Samir Rawashdeh

M

DEARBORN

Rafi Ud Daula Refat

rerafi@umich.edu

ORCID iD: 0000-0003-3812-3244

DEDICATION

I dedicate this dissertation to my parents, my loving wife Noshin and my respected elder brothers Rokon and Asaf.

ACKNOWLEDGEMENTS

I have an incredible amount of gratitude to everyone who helped and motivated me along the way to earning my doctorate. I want to start by expressing my sincere gratitude to my supervisor, Dr. Hafiz Malik, for his continuous support, guidance, and knowledge. His advice and enlightening feedback were extremely beneficial in helping me polish and improve the caliber of my study. I am grateful to all of the members of my thesis committee for their insightful advice. Their knowledge and various viewpoints have made a significant impact on the creation and improvement of this work. I would like to express my sincere gratitude to the members of the information systems, security, and forensics (ISSF) lab for their selfless dedication of time and sharing of their viewpoints and experiences with this study. I want to sincerely thank my parents for their constant support, love, and sacrifices during my academic career. I owe my lovely wife Noshin a debt of gratefulness for her everlasting faith in me and her unflagging support. I want to express my appreciation for the intellectual and emotional support that my two siblings, my in-laws, my nephew and niece, and friends have given me. I am appreciative that the University of Michigan-Dearborns' ECE department gave me the tools, resources, and chances I needed to carry out my research. I want to express my gratitude to Amanda Donovan, the ECE department's administrative assistant, for resolving all the administrative concerns with a constant grin. Finally, I would like to acknowledge everyone who has helped me during this quest (too many to name individually). Your support and belief in me helped this Ph.D. thesis to be completed, as well as your contributions.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF ACRONYMS

ANN artificial-neural-network

CAN controller area network

CAV connected autonomous vehicle

CNN convolutional-neural-network

DL deep-learning

DSRC directed short-range communication

IDS intrusion detection system

IoT internet of things

KNN k-nearest neighbors

LIN local interconnect network

MOST media oriented system transport

OBD on-board diagnostics

SVM support vector machine

WEP wireless encryption protocol

WMA windows media audio

ABSTRACT

Connected autonomous vehicles (CAVs) hold the promise of not only enhancing functional safety but also improving mobility and the efficiency of transportation systems. The CAV is a cyber-physical system (CPS) that contains many networked electronic control units (ECUs), sensors, actuators, wireless interfaces, and an advanced driver assistant system (ADAS). Just like other CPS, CAVs rely on data gathered from the sensors, actuators, and software for critical decision-making to enhance efficiency, reliability, safety, and functionality. Besides, CAVs utilize connectivity to improve drivers' and passengers' experience by integrating built-in wireless interfaces like WiFi, Bluetooth, etc. The growing connectivity feature of modern vehicles is marking them more vulnerable to cyberattacks. Many researchers have successfully exploited the remote connectivity-induced attack surface. According to the recent industry report (1) on cyberattacks on CAVs indicated that more than 700 incidents were reported targeting vehicular systems between 2010-2020. Among them, in 27.63% of these incidents, attackers tried to control or manipulate the vehicle which could jeopardize passenger safety. Based on the literature, several intrusion detection-based solutions have been proposed to detect attacks on CAVs. While the solutions are effective against a certain range of attack vectors, they are limited in scope and effectiveness. For instance, existing solutions are unable to localize the attack. In addition, existing state-of-the-art require thousands of in-vehicular network (IVN) packets for intrusion detection. It is therefore important to develop reliable, robust, and real-time security solutions to safeguard CAVs by mitigating emerging cyber threats.

This dissertation aims to address the aforementioned cybersecurity challenges of CAVs by developing a robust and reliable framework to safeguard against attacks at different points through a multi-layered framework. Each layer of the proposed solution aims at neutralizing cyberattacks on in-vehicle networks by breaking some critical links in the attack chain. The first layer of the proposed framework aims to protect IVNs by developing a sender identification algorithm that utilizes the unclonable signal attributes to fingerprint transmitting ECUs. The proposed framework is novel and efficient that leverages the uniqueness of physical signals to create images and uses a deep learning algorithm for attack detection and localization. The second layer aims to protect IVNs against firmware attacks using ECU behavioral fingerprinting through a data-driven graph theory-based approach. The proposed methodology takes advantage of a huge amount of IVN data

to model the normal behavior of the network by using graph analytics and develops a network monitoring system to detect unusual behavior created by attackers.

The effectiveness of the proposed multilayered framework is evaluated by conducting a series of experiments using bench testing and as well on vehicular public data. The experimental results suggest that the proposed multilayered framework is capable of detecting IVN message injection attacks with higher accuracy and can reliably localize the attacker on the network. Additionally, the thesis hypothesis and solution were validated by conducting market research through active participation in both the regional and final programs of the National Science Foundation (NSF) I-Corps. In the future, I plan to build a prototype of the proposed framework and deploy it in actual vehicles for rigorous field-testing.

CHAPTER 1

Introduction

Vehicles are considered as one of the greatest inventions in the past 1000 years (8). They were first invented in 1885 and now it has become an integral part of the world's economy. According to the bureau of labor statistics, more than 872k people are directly employed by the automotive industry (9) as there is a very large vehicular market demand. In 2020, the north american automotive industry has produced 13.4 million vehicles and In the last 10 years, billions of vehicles have been sold only in the USA reported by (10). With the increasing curve of market demand, the transformation of vehicles is happening rapidly. Nowadays they are not only used as a form of transportation but also considered as IOT systems as they are equipped with sensors and have high computational power. Currently in the global automotive industry, there are three disruptive trends that may change the industry: autonomous driving, shared mobility, connectivity, and electrification (11). While autonomous vehicles perceive surroundings through sensors and use connectivity to collect other vehicles' intent and status to improve mobility, electric vehicles aim to reduce fuel consumption and control CO2 emissions to protect the environment. This work focuses on autonomous vehicles & connectivity problems rather than the electric transformation problems.

1.1 Connected autonomous vehicle - future mobility model

Connectivity & automation are the foundational attributes of autonomous vehicles. The connectivity is the backbone of the connected vehicles where the vehicles communicate with each other. On the other hand, automation is the foundation of self-driving cars. These two technologies can improve the luxury features, functional safety and traffic mobility in vehicles. As a demonstration, recent work shows that these technologies can reduce traffic accidents and improve transport system efficiency (12). According to the author in (13), the accident rate will be lower if human errors can be reduced and connectivity & automation can be used for this purpose. They can also be used to improve the lifestyle of citizens. They can help people who are not able to drive a vehicle to travel anywhere they want. They can even contribute to the complex operations of product delivery hence reducing human involvement in the risky task.

These two technologies are merged by the automotive industry to evolve the term 'connected autonomous vehicle' (CAV). Where the connectivity gathers information from the vehicle's surroundings and passes it to the intelligent decision-making unit of a vehicle. The vehicle can take the right decision using this valuable information. Although we have not reached the highest level of automation, there are some vehicles in the market that are at level 3 (controlled automation). Fully automated vehicles i.e. level 5 automation is not possible unless the decision-making unit of the vehicle is trustworthy and capable of performing driving tasks like an actual human without any fear of cyber attackers.

1.2 Brain of the CAV

The complex tasks inside a vehicle are performed by small mini computers called Electronic Control Units (ECUs). More precisely, they are a set of microcontrollers and are controlled by embedded software programs. According to (14), Modern vehicles contain around 70 to 100 of these ECUs and are connected through one or more in-vehicle networks. To operate the vehicle functionality, ECUs receives signals from the sensors and depending on the data, it controls the actuators of a vehicle. Each ECU is responsible for performing mini tasks and sharing important information with other ECUs. That is why, ECUs are called the brain of the vehicle.

1.3 Current security challenges of CAVs

Once the vehicle was considered a closed system when it was used only as a form of transportation. Back in those days, the only way to access the in-vehicle network data was to check the OBD-II port of the vehicle. But with the advancement of wireless technology, the vehicles can be accessed using bluetooth, wifi, hotspot etc. As a result, they are turned into open IoT systems nowadays that can be accessed externally. These external interactions can be used as an attack surface to perform attacks on the vehicles by exposing confidentiality, the integrity of the in-vehicle network. According to Forbes magazine (15), almost every automotive manufacturer has been hacked at least once. As an impact, the manufacturers had to recall vehicles for fixing (Chrysler recalling 1.4 million Jeep cars in 2015, (16)), shut down the production in the factory (Honda in august, 2020 due to ransomware attack, (1)), OTA automotive software update (Tesla in 2016 after message injection attack by Keen Security Lab of Tencent, (17)), etc. Earlier, the security of automotive systems was overlooked most of the time, until recently both the government and the automakers are paying attention to this matter. In March 2016 the FBI issued a warning about the potential security weakness of the car's on-board system. The automakers are launching different bug bounty programmers to make their vehicles more secure. . In 2020, OEMs, such as Tesla, GM, Ford, FCA, Daimler, & more and the Tire-1 companies all host bug-bounty programs on platforms like BugCrowd, HackerOne, or their own websites (1). And the program is actually working for them. Uber's bug-bounty program has received 1,500 plus software vulnerabilities reports by 2020. It has increased 13 % in 2020 from 2019. Companies like Tesla offered $1 million and a car as a bug bounty reward as well (17). So, nowadays, arguably the automotive security research area is an active research field.

The number of cyber attacks on vehicles is increasing exponentially because of the increased interaction of vehicles with the outside world. Apart from the driver, any outsider can access the vehicles through a wireless or wired medium. These connections can be used as attack surfaces and make an impact on the security of the vehicle. For example, a gray hat hacker connected an Arduino to the car OBD-II port and injected a message to the in-vehicle network of a Mercedes in 2019. Wireless attacks are growing significantly as well. Between 2010 to 2020, 79.6% of attacks have been performed remotely. Among the huge number of attack reports, the black hat hackers were responsible for 54.6 % of all the attacks (1). The target of the hackers is actually breaching data/privacy, controlling or manipulating car systems, disrupting service/business, fraud, tracking location, violating the privacy, etc. According to the 2021 report of (1), 27.63 % incidents tried to control or manipulate the car system or disrupt service or business. The previous incident like the famous hack from Charlie Miller and Chris Valasek in 2015 (16), tesla hack from the keen lab, etc (17). manipulated or controlled the car by injecting malicious messages to the in-vehicular networks (IVNs). That is why securing IVNs is one of the trending works in the cyber security space.

1.4 Problem statement

This thesis aims to identify, localize & mitigate cyber attacks to protect the in-vehicular network that bridges the safety-critical ECUs in a modern vehicle. The goal of this work is to safeguard CAVs against growing attack surfaces and vectors by developing holistic solutions through multiple seamlessly integrated layers of defense, with each layer aiming to mitigate a specific set of attacks. The reason for defense systems in different layers is important due to the difference in the nature of attack models. The attack models are discussed briefly in chapter 3. The intruder who wants to misguide the vehicle by making the in-vehicle network unavailable can be identified by the intrusion detection system based on the network packet-level analysis. This approach is called the behavioral analysis of the in-vehicle network. However, this approach can not localize the compromised ECU or the physical channel. To achieve the sender information, physical signals are needed to be analyzed and linked to either to the specific channel or to the ECU. That is why

IDS in the multiple layers of the in-vehicle networks is needed to mitigate cyber attacks.

1.5 Research objective

The following are the specific research objective for the thesis.

- Investigate message injection attacks in automotive IVNs.
- Design & develop novel multilayred framework to detect message injection attacks in IVNs.
- Evaluate the performance of the proposed frameworks using bench testing.
- Design & evaluate a framework to execute various message injection attack using an actual vehicle.

1.6 Outline of the dissertation

The thesis aims to detect cyber attacks for connected autonomous vehicles and localize the attacker to isolate the attack points. To achieve this, chapter 2 presents the overview of in-vehicle networks. Chapter 3, the access points for the intruders to access the IVNs is explained. The chapter is named as threat modeling and also holds the motivation of the attacker & the impact of the attack. In chapter 4, the related work is presented and the research problem is identified. The proposed multilayered framework is presented in chapter 5. Chapter 6,7,8 and Chapter 9 represents the detail of intrusion detection using behavioral fingerprinting approach and physical fingerprinting approach. Finally, the report is ended with a discussion & future work.

CHAPTER 2

CAV In-vehicle Network

2.1 In-vehicle network(IVN) protocols

Current automotive vehicles use multiple communication protocols to exchange information between ECUs, vehicle to vehicle (V2V), vehicle to infrastructure (V2I), and vehicle to pedestrians(V2P). The IVN protocols have their own characteristics like speed, specification and use cases. They help to reduce the complex wiring system of a car thus making it easy to maintain & reducing cost. This section presents the conventional automotive networking protocols such as LIN, MOST, CAN, FlexRay, Ethernet, etc.

2.1.1 LIN

Local Interconnect Network (LIN) was introduced in the late 90's as a low-cost alternative of CAN bus protocol to connect components in a car (18). It allows serial communication in a master-slave architecture that has a data rate of 20-25 kbit/s. However, due to the low bandwidth of data communication, it is impractical to use a high-speed communication system. But it became so popular to connect non-critical subsystems where the speed of communication is not considered as an issue because the protocol itself is low cost. It has been used in subsystems are like adjusting seats, mirrors, or controlling car windows, etc (19).

2.1.2 MOST

Media Oriented System Transport (MOST) is a high-speed multimedia network protocol and was developed in 1988 by MOST corporation (20). By architecture, it supports up to 64 ECUs to connect in a ring topology structure. It has a data rate of 24.8 Mbit/s to the maximum. (19). Due to the support of high-speed communication, it became popular for the car infotainment system. World's renowned car manufacturers like BMW, Mercedes-Benz, Porsche, Audi, Volkswagen, Jaguar, Hyundai, Toyota, Land Rover and many others using MOST for many years (21).

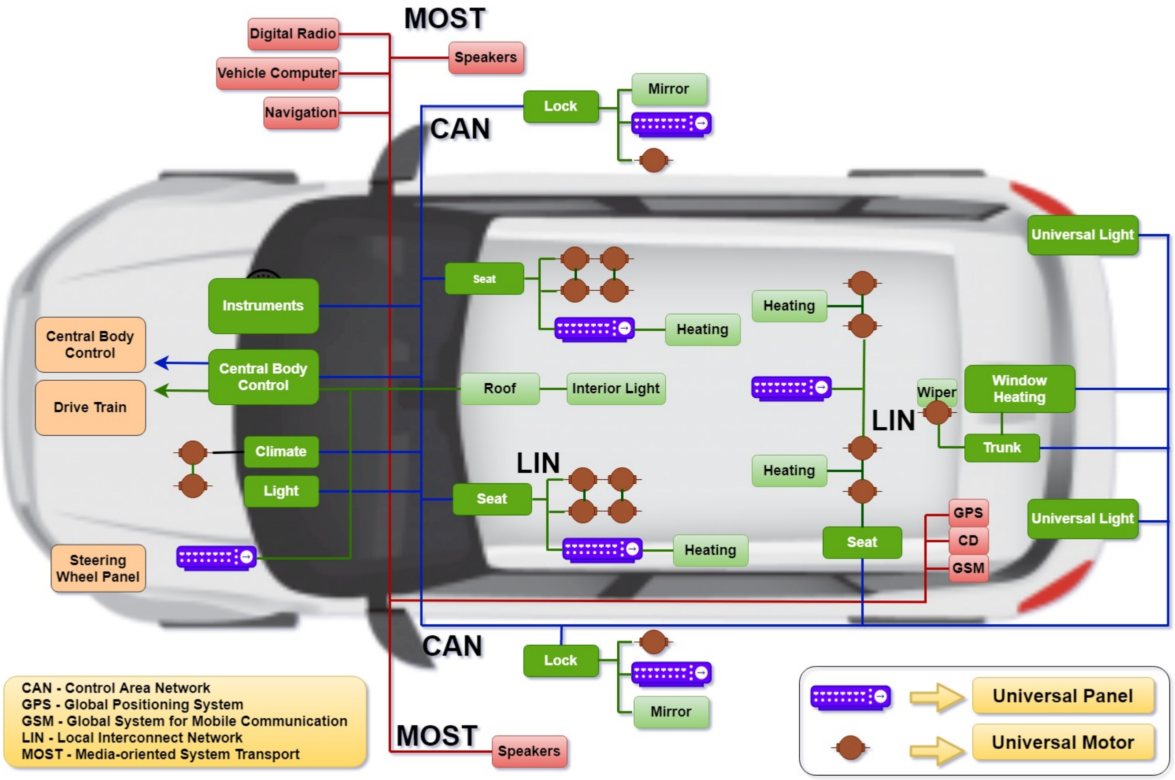


Figure 2.1: In-Vehicle Network Architecture.

2.1.3 FlexRay

FlexRay is another in-vehicle communication protocol and was first introduced in vehicles by BMW in 2006 (20). It has a data rate of 10 Mbit/s. That means, it supports high-speed communication between ECUs and is used for applications like active suspensions, adaptive cruise control, etc (19). BMW 7 Series is the first car that used FlexRay fully in the car (20). Other interesting characteristics of FlexRay are, it can fit any kind of topological network structure.

2.1.4 CAN

Controller area network (CAN) is the most widely used in-vehicle communication protocol. It was developed by German company Robert Bosch GmbH in 1980 (22) and was first made public in 1986 (20). The introduction of CAN protocol dramatically solved the complex wiring issue in a vehicle and it became popular in a short time. By architecture, it is a broadcasting system and any ECU can access the bus anytime it wants. For their access control, Carrier Sense Multiple Access with Collision Detection (CSMA/CD) is used with the help of a few bits in the CAN message frame called arbitration ID (23). In terms of speed, it has a data rate of 1 Mbit/s. Due to its simplicity and

Table 2.1: The in-vehicle network buses specification(4)

	AE	**CAN**	**FlexRay**	**MOST**	**LIN**
Bandwidth (Mb/s)	1000 (developing)	1 or 10 (CANFD)	20	150	0.02
Max number of nodes	No of switches	30	22	64	16
Network length	15 m per link	40 m	24 m	1280 m	40 m
Messaging	IP based	Multi-master	Multi-master	Cyclic frames /streams	Master-slave
Cost	High	Low	Low	High	Very low
Safety-critical functionality	Proven other industry	Yes	Yes	Yes	No
Availability	Growing	Many	Few	One	Many
Cabling	UTP	UTP	UTP	Optical,UTP	1-wire
Main applications	Infotainment, Backbone (future)	General bus	Safety-critical, X-by-wire	Infotainment	Seats, doors, switches

easy plug & play feature it is used in communication between safety-critical ECUs and is used by most of the leading car manufacturers in the world.

2.1.5 Ethernet

Another popular IVN used nowadays is the Ethernet. It is a physical network that connects different components of a vehicle. The popularity of Ethernet is visible by its recent use by Hyundai, BMW and Volkswagen (24). But as a communication protocol Ethernet is not new, it has been used in computer networking for more than 20 years and was not considered for the automotive industry because of OEM's automotive requirement. But extensive research work to develop driverless vehicle and camera-based ADAS systems has convinced the automotive industry that its components need high speed, higher bandwidth buses. As a result, IEEE 802.3 Ethernet was introduced by automotive makers. The modern software & electronics in vehicles can introduce exciting functionalities because of the speed and bandwidth offered by Ethernet. Unlike CAN protocol, Ethernet protocol has sender and receiver information in the message packet and is considered more secure.

2.2 Heterogeneity of IVN architectures

Current automotive networks are considered as an internet of things (IOT) where they are equipped with 70-100 ECUs. These ECUs serve different functionalities in a vehicle where some functionalities require real time interactions among ADAS cameras, sensors (ultrasonic, lidar, radar), powertrain sensors, actuators, etc. and some requires low data rates such as door lock/unlock, window lock/unlock. To satisfy different need of data rate of different units, different IVN protocols are used in a car heterogeneously. As the IVN protocols contribute to the overall production cost of the vehicle, this heterogeneously network architecture is highly adopted by the industry like shown in figure 2.1. This multiple different networks are supported by a computationally expensive intelligent device called gateway which works as a master in a master-slave network. This heterogeneity of networks poses different security challenges in a vehicle. To make a vehicle safe we can replace everything with the most secured network which is not so cost-efficient. For example, automotive Ethernet has a data rate of 10 MB/s to 100 GB/sec and has source and destination address in the message packet to ensure CIA in data communication, but it is so costly to implement in the vehicle as it requires switches to connect multiple ECUs and itself they are expensive. So, to connect everything with the automotive Ethernet is not feasible and cost inefficient. So, the Heterogeneity of automotive IVN is here to stay for next generation of vehicles.

CHAPTER 3

IVN Threat-Model

A cyber attack is defined by unauthorized access (physical/remote) to a system with the intent of interrupting, destroying it, and/or modifying, controlling its' behavior that can be motivated to achieve financial gains. This chapter comprises the details of cyber attacks on the automotive platform. As a part of that, we first started the chapter with the touch points the attacker can choose to enter the system, then we discuss how an attacker can take advantage of multiple attack surfaces to launch an attack. After that, the motivation of the attacker is discussed. Finally, we talk about the current report of the impact of the automotive cyberattacks and the built-in security vulnerabilities of IVNs.

3.1 CAV attack surfaces & attack vectors

Current vehicles have different interfaces that are used to establish access directly or indirectly to a vehicular system. In this sub section, we discuss the connectivity ports of the vehicle and how they can be used to mount an attack on a vehicle. The can come in the form of a physical interface or wireless interface.

3.1.1 Physical interfaces

Physical access ports are the touchpoints that can be used to gain access to the vehicle by intruders with the help of some hardware connection. For example the OBD-II port of a car, any cable connector (USB/AUX, etc.), CD/DVD players, etc. OBD-II port provides a significant direct physical access opportunity to the vehicle. It is mandated by US federal law and is currently used in all vehicles. Using the OBD-II port anyone can directly access the internal in-vehicle networks. This can be used by the attacker as a weapon to perform attacks on the automotive network. As shown in table 3.1 the pin 6 and pin 14 can be used to access the CAN bus directly. Any message injection attack or denial of service attack (DoS) can be mounted on a vehicle using the OBD-II port.

Table 3.1: OBD-II pinout description

Pin	Description	Pin	Description
1	Vendor option	9	Vendor Option
2	J1850 Bus +	10	J1850 Bus
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 Low
8	Vendor Option	16	Battery Power

CD players are used as a form of entertainment feature in a vehicle that can play audio or video. Audio and videos are a very essential element of entertainment and vehicles all over the world support most of the formats of audio (mp3, WMA, etc.) and videos (mp4, 3gp, etc.). However, researchers have identified vulnerabilities in the CD players. In (25), researchers were able to bypass the security of the CD player by embedding a text file in a WMA audio file. The player was unable to detect the text file while playing the audio. However, this is considered as a low-level threat surface as the popularity of CD players is decreasing day by day. Currently, all the vehicular system provides the feature of connecting cable to the system. They can be used to charge phones, play music, etc. They can be even used for firmware updates as well. The above-mentioned vulnerability of the CD player remains valid for mounting cyber attacks via cable connection.

3.1.2 Short range wireless interfaces

The short-range wireless access medium includes bluetooth, RFID, keyless entry, tire pressure monitoring system, dedicated short-range communication network, etc. They all can be used as a backdoor to mount cyberattacks. For example, bluetooth is very popular in vehicles nowadays. They are used as a wireless medium that connects mobile devices to the vehicular infotainment system. It supports the hand-free control of the sound system which helps the driver focus on driving and indirectly reduces accidents. However, most modern cars allow the users to sync and operate the mobile phones in the car infotainment system. This can be risky as the intruders can collect vital trip information using malicious applications.

Almost all modern vehicles use the RFID based remote key and keyless entry. RFID based remote key is the feature to prevent car theft because the car can not be started if the remote key is near the car. Keyless entry technology is used to unlock the car wirelessly. These wireless interfaces can be used to perform attacks as well. According to the reports of upstream 25.3

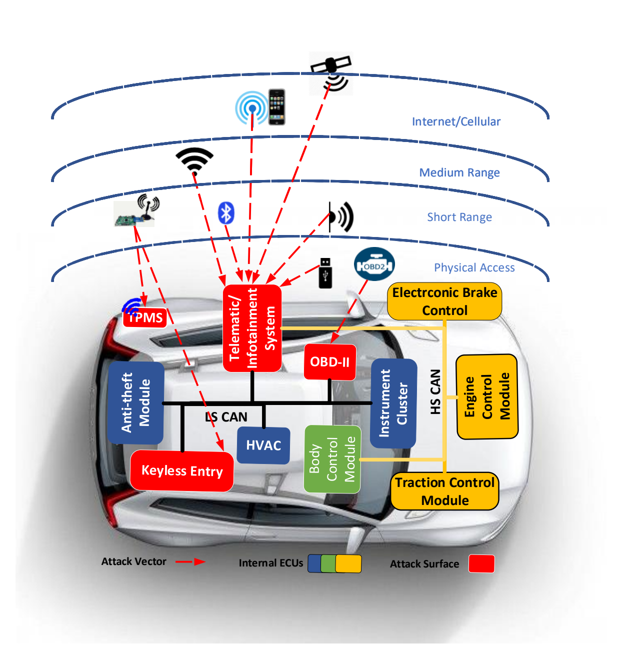


Figure 3.1: Current vehicle as an IOT system

% incidents happened through the keyless entry system. The tire pressure monitoring system acknowledges the status of the tire and sends a digital telemetry signal to alert the driver if the pressure is low. It is mandated by US law that every vehicle should have this feature. However, It can be used as an attack surface to send a false alerts to the driver according to (26).

3.1.3 Medium range wireless access

Directed short-range communication (DSRC) is a wireless interface that enables vehicles to interact with other vehicles or with the infrastructures. This interface plays a vital role in the functionality of connected vehicles. This can also be used as an attack vector to send false information to the receiver. Apart from that, an intruder can use wifi to establish connectivity with the car as well. The intelligent vehicles are currently equipped with Wi-Fi and consequently, they can connect to the internet via Wi-Fi hotspots on the roadway within the same range of the vehicle. However, some of these wireless hotspots might put the vehicle at risk for a variety of reasons. For instance, these wireless hotspots may employ outdated encryption standards, putting the vehicle security at risk.Such as the Wireless Encryption Protocol (WEP) in earliest version is deemed weak and vulnerable to hacking. This vulnerability introduced Wi-Fi protected access (WPA) as the wireless networking standard, but it, too, was proven to have flaws. Furthermore, these wireless hotspots may expose vehicles to a rogue or fake Wi-Fi hotspot (27). For example, in the case of the vehicle connect to a malicious hotspot, this allows the hacker to operate many activities on the vehicle such as transfer malicious code to the vehicle. As a demonstration, Nie et al.(17) were able to remotely hack a Tesla vehicle by exploiting the way that the secret key to an installed Wi-Fi was saved in plain text. Furthermore, Nakhila et al. (27) showed that by connecting to an illegitimate Wi-Fi access point, an attacker may eavesdrop on Wi-Fi activity. Vanhoef et al. also looked at the possibility of Denial of Service attacks against Wi-Fi Protected Access (28).

3.1.4 Long range wireless access

The long-distance digital access channels, which are divided into two types: broadcast channels and addressable channels, have been deployed in intelligent vehicles now. The broadcast channels, such as GPS, Traffic Message Channel, Satellite Radio, and Digital Radio, are indirect channels that receivers tune into as part of a media system that is connected to other important ECUs. However, because it is difficult to attribute and command multiple channels at once, these channels are subject to external surface attacks, which might allow an attacker to manipulate channels and their behavior. The addressable Channels, as opposed to broadcast channels, are direct channels that frequently employ cellular phone and data networks and may be accessed over arbitrary distances. However, this type of long-range wireless is vulnerable to attack by the remote transfer system that provides continuous connectivity through cellular voice and data networks (25).

The intelligent vehicles are currently equipped with cellular network technologies such as LTE, 3G, 4G and now 5G (29) and consequently, they can communicate to either another vehicle (V2V) or the infrastructure (V2I) at long distances on the scale of miles (30). Cellular networks, on the other hand, are prone to eavesdropping and jamming attacks (31). Cichonski et al. demonstrated that LTE can be hacked easily by jamming attacks and eavesdropping attacks (31). Other work by Muhammad et al. (32) demonstrated that the LTE and 5G-based vehicular networks are vulnerable to a huge number of attacks. This allows attackers to track vehicle whereabouts in order to get access to the vehicle and carry out harmful operations inside it. For instance, Miller et al. (33) have been able to hack and stop a Jeep Cherokee running on a highway remotely through 4G.

3.2 Motivation of the attacker

The motivation and goal of the attacker are as follows.

3.2.1 Theft

Theft is the main motivation of the attacker in most cases. It is a booming “business” among criminals, with rising numbers reported worldwide. The UK reported a 60% rise in car thefts in 2020 while in India a gang of thieves in India was arrested for stealing over 100 vehicles using electronic devices (1). From a technical point of view, an attacker can use the vulnerability of the keyless entry of the vehicle to unlock it, deactivate the security alarm of the vehicle and start it without the key. This break-in can result into steal the vehicle or owner's personal belongings or/and important documents. In a recent event, automotive security experts said they have discovered a technique for car theft that depends on having direct access to a car's system bus via the electrical wiring of a headlight. (34).

3.2.2 Privacy breach

An attacker could infringe the privacy of drivers by infusing spyware into a vehicle. An attacker could steal and access sensitive and private data about the driver for example, where he is located, his driving propensities, his credentials, his visa and banking information, his telephone number and call history, the music he tunes in to, and considerably more. According to an IBM Security report (35), a third party was able to gain access to the personal information of 27.7 million Texas drivers. An attacker can aim to obtain confidential information about a vehicle. Confidential information actually means the reverse engineering of an in-vehicle network or retrieving the source code of ECUs. By this attacker can sell the intellectual property to other companies for financial gain.

3.2.3 Electronic tuning

Electronic tuning happens when one tries to modify the ECU level of the vehicle in an unauthorized way. For example, one can try to reprogram the ECU to lower to millage of the vehicle before selling it to others. In another way, an owner can try to install more powerful equipment by programming the ECUs. Or he/she can buy a cheap ECU as a replacement to install it by him/herself. This can motivate someone to tune electronic equipment in a vehicle and can result in serious consequences in the future.

3.2.4 Financial gain

An attacker can restrict the driver's access to his vehicle by infecting the vehicle remotely with ransomware which can disable the vehicle's functionalities such as immobilize the motor, locking the in-vehicle radio and locking the doors. Such an attack could restrict the vehicle's functionalities in a way that the proprietor's car keys can no longer activate them. The attackers would then be able to demand payoff before these functionalities were re-enabled. As a demonstration for academic research purposes only, work by wolf et al. (36) showed that vehicle ransomware can be easily created and deployed. Additionally, researchers from McAfee security (37) demonstrated that the ransomware can block the use of the vehicle until the ransom is paid. Furthermore, fraud can be a major route for hackers to bring in cash. Hacked vehicles could give access to stalkers to be able to track the vehicle identification number of any potential victim through GPS since all intelligent vehicles nowadays have GPS. So in an event that an attacker can track any vehicle, the attacker can begin assistance for anybody that needs to track someone can in exchange for money. As a result, the attacker can gain a lot of money by tracking hundreds of vehicles. It's an extraordinary business. As an example of that, according to a report from Boston 25 News (38), an attacker was able to track a vehicle for many years by hiding a GPS tracking device on the victim's car. Another way of hackers to bring cash is automated toll booth payments, it may create more points of entry for hackers to steal individual information, for example, visa or banking data. Hackers are hoping to put forth the greatest benefit for the base attempt since the intelligent vehicles are going to have a lot of payment systems in order to provide the comfort for the driver to pay via his vehicle when he goes to toll roads and parking lots (39).

3.3 Impact of cyber attack

Attack impact is the consequence of cyber attacks. It is actually the scenario of a post-attack situation. From 2010-2020, there were more than 700 incidents reported that targeted the automotive systems. Among them, 30% incidents have resulted in data/privacy breaching. 28.14 % times the attacker stole the vehicle or just broke in (1). The attackers tried to control or manipulate the vehicle system 27.63 % times. The remaining incidents interrupted automotive production or violated the privacy of the owner or tracked the trip route of the driver. Although, the incidents that resulted in vehicular theft or data breaching is high in number, the incidents where the attacker manipulated or controlled the vehicle are considered as a threat to the passenger of the vehicle or even a threat for the pedestrians also. This section leads us to question how architecturally secured is the IVNs?

3.4 Security analysis of IVNs

As mentioned in the introduction section, vehicle connectivity can be leveraged by intruders and they can misguide the vehicle. It is considered a serious issue because passengers' life can be at risk if the hackers can remotely take control of the vehicle. In this section, we discuss the loopholes that each IVNs create architecturally. The MOST protocol is not discussed here as best to our knowledge, there is no work that takes advantage of MOST protocol to perform attacks.

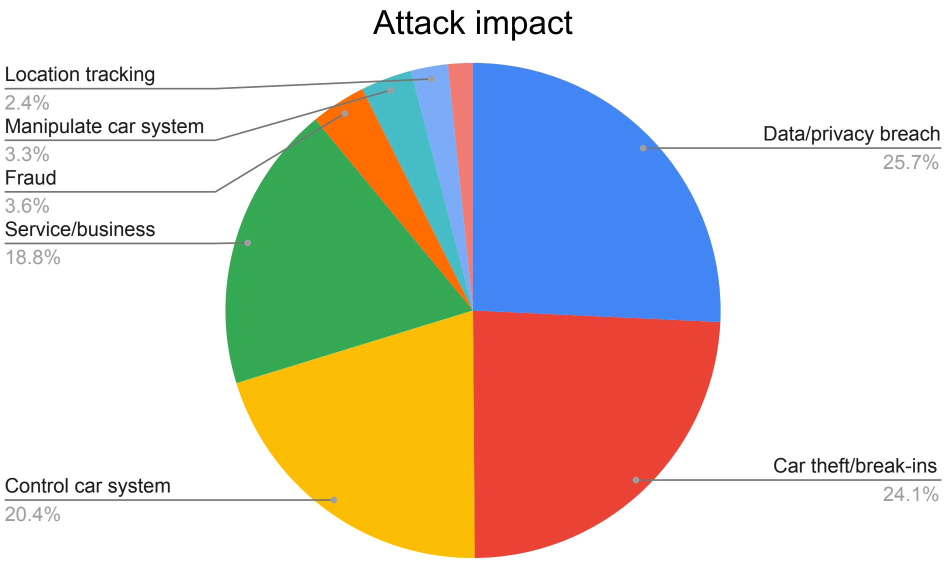


Figure 3.2: Cyber attack impact on automotive based on the actual events from 2010-2020 (1)

3.4.1 LIN

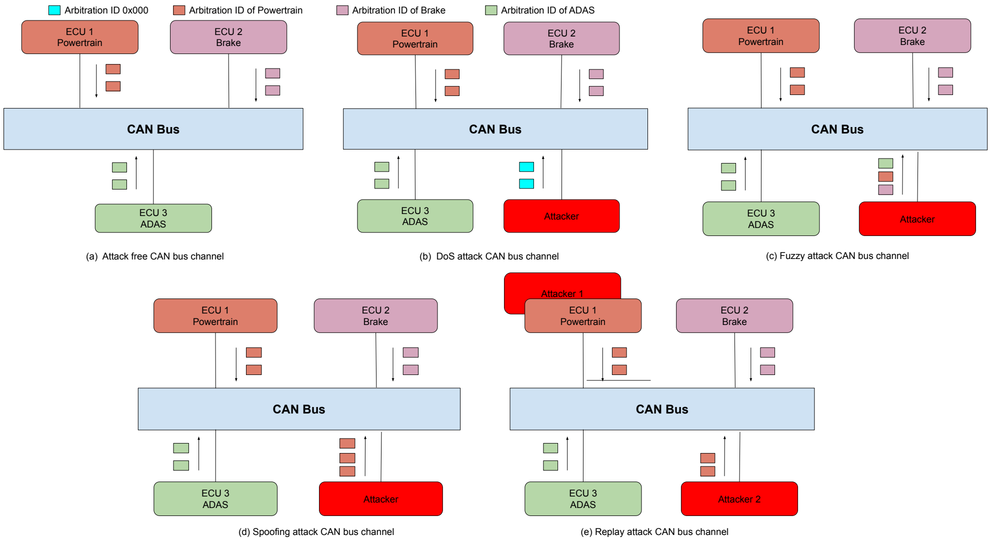
LIN is known as low cost IVN that connects non safety critical sensors and ECUs in a vehicle. By architecture it is a single wire system and has UART serial interface (40). Modern vehicle systems connect a master and several slaves to form a LIN network and they send two types of message in the bus i.e. (i) unconditional message frame & (ii) event triggered message frame. In an unconditional message frame the master specifies a slave to respond with a message and the slave follows the command. The event triggered message frame is sent to the bus by the master when information is requested from the slave. The main difference between unconditional message frame and event triggered message frame is the response of the slaves. Unlike the unconditional message frame, the slave did not respond in an unchanged state upon receiving the event trigger message frame. From an intruder point of view, this feature can be used to perform attacks. As master initiates action for all the slaves, gaining access to one of those gives the attacker access to the bus. Apart from this another type of attack has been reported by the researchers. That is sending the sleep command (41). Although this is considered as the unique advantage of LIN communication protocol to go to sleep mode to save power, it can be exploited by the attacker to disable the LIN bus system in a vehicle. Lastly, the normal state of a LIN bus system can be hampered by an attacker using electromagnetic interference as it is a single wire system (40).

3.4.2 FlexRay

The FlexRay is another IVN protocol that provides high-speed communication between ECUs. It has been used recently in several applications such as adaptive cruise control and active suspensions (19). It is tailored to the demands of today's automotive industry, with features such as flexible data transfers, support for any kind of topological network structure, fault-tolerant operation, and greater data throughput than prior standard protocols. It is also a dual-channel system that supports both asynchronous and real-time data transfer modes (20). Although it provides some protection features of data availability and data integrity since it uses CRC as a kind of data protection against transmission mistakes. Nevertheless, these features do not imply any guarantee of data confidentiality, authenticity, or freshness. A FlexRay network, in reality, does not have directly accessible interfaces, instead, it connects to other network protocols through gateways. Access to the FlexRay bus can be gained through such gateways. This makes the FlexRay protocol is insufficiently protected against attacks and makes the FlexRay bus is a likely target for attackers since the ECUs attached to it are utilized to provide control and mobility in the vehicles. These cyberattacks can target the in-vehicle network's control and maneuverability ECUs, causing significant harm to the driver. A variety of attacks have been easily created and developed on the FlexRay standard protocol including the Nilsson-Larson attacker model (42), in which an adversary has access to the in-vehicle network via the wireless gateway and can read, modify, flood, steal, drop, monitor, record, broadcast, spoof and replay messages. Also, the Man-in-the-middle attacks, or intercepting and dropping messages, are possible with such an adversary (43).

3.4.3 CAN

Controller area network (CAN) is known as the de facto standard for IVN communication. Architecturally it is a broadcasting system that means any CAN messages sent to the bus can be accessed by all the components connected to the network. This makes the vehicular networking system more simple and solves the complex wiring problem. But the design of the protocol lacks authentication features because it does not have any field containing sender or receiver information in their message frame.A typical standard CAN dataframe has 111 bits at most. Out of them, it has a unique arbitration ID (11 bits), that is used to control the accessibility of the CAN bus by establishing priority scheme. According to the scheme, lower arbitration ID has a higher priority to send messages to the bus. That means if two ECUs try to send CAN message to the bus at the same time, the protocol lets the sender with lower arbitration ID to send message first before the other one. Theoretically, if an ECU sends a CAN message with an arbitration ID of 0000 continuously, then the other ECUs will not get chance to transmit messages to the bus. There is also 15 bit CRC field in CAN dataframe that is calculated from the data fields(0-8 bytes), but that only



(a) Attack free CAN bus channel

(b) DoS attack CAN bus channel

(c) Fuzzy attack CAN bus channel

(d) Spoofing attack CAN bus channel

(e) Replay attack CAN bus channel

Figure 3.3: Different CAN bus channel scenarios

protects the data loads. By default there is no other ways to provide security to the CAN message frame. The attacker can take advantage of the CAN bus protocol characteristics discussed above and manipulate the bus. To be more specific, there is three important properties of the protocol that an intruder can utilize to perform attacks according to the state of the arts (3). First, the broadcasting nature provides a way for the attackers to read all the CAN messages in the network without getting noticed. Secondly, the priority scheme of the protocol can be misused by the intruder and denial of service attack can be performed by sending highest priority CAN message continuously. Finally, due to the lacking of sender identification an attacker can act as an authorized ECU and can send CAN messages to the bus to perform spoofing attack. The following are the message injection attack scenarios in CAN bus channel.

3.4.3.1 CAN bus message injection attack scenarios

Based on the above described adversary model, we consider the normal CAN-bus data (attack-free) in addition to four kinds of attack scenarios which are spoofing, fuzzy, DoS and replay as shown in figure 7.1.

Spoofing attack: This attack happens when an attacker injects a single message of randomly spoofed CAN identifier with arbitrary data like shown in figure 7.1 (d) Subsequently, it causes unintended vehicle behaviors since all ECUs will receive that message. To exploit the spoofing attack, an attacker can inject arbitrary data into one message of the in-vehicle messages and chose the target identifier of that message to create unexpected behaviors for the vehicle. Such behaviors include turning the signal lamps light irregularly, flickering the instrument board in incalculable ways, disabling the braking system and shaking the steering wheel colossally. (44; 45)

Fuzzy attack: This attack occurs when an attacker injects multiple messages with arbitrary data of randomly multiple spoofed CAN identifiers, unlike the spoofing attack which occurs by injecting only a single message of randomly a single spoofed CAN identifier. As a result, all ECUs will receive various messages which cause unintended vehicle behaviors like gearshift changes automatically, disabling the braking system, instrument panel blinks in incalculable manners and the steering wheel shakes gigantically. Figure Figure 7.1 (c) shows the fuzzy attack scenario in CAN bus. (44; 3)

DoS attack: This attack happens when an attacker injects high priority of CAN messages such as the 0x000 CAN ID packet in a short cycle on the CAN bus. Figure 7.1 (b) To exploit the spoofing attack, an attacker can easily occupy the bus by injecting the highest priority identifier of CAN messages such as 0x000 in a short cycle on the CAN bus. Subsequently, it yields latencies of other messages and causes threats in regards to availability with no reaction to the driver's commands since all ECUs share a single bus. Unlike the spoofing and fuzzy attacks, the DoS attack delays the normal messages through the occupancy of the CAN bus rather than cripple the functions of a vehicle. (44; 3)

Replay attack: To mount a replay attack, the adversary needs to compromise atleast two ECUs, one as a strong attacker and the other as a weak attacker, as shown in Figure 7.1 (e). The adversary monitors and learns which messages are sent at what frequency by the weakly attacked ECU, and then the strong attacker transmits the message with the ID of the compromised ECU at the same frequency. (46; 3)

3.4.4 CAN FD

As the CAN FD is designed based on CAN protocol. Unfortunately, it has most of the security limitations CAN protocol posses (47). Although it has a larger payload field than the CAN, so encryption based algorithms can be used to solve the security threats in CAN FD. However, it requires new hardware because current ECUs are 8-16-bit processors without cryptographic acceleration capabilities, and also requires a key management system. Moreover, the security of message encryption algorithms is also questionable as MAC uses the bandwidth.

3.4.5 Automotive Ethernet

The Ethernet has recently been employed in camera based ADAS systems and self-driving vehicles (48). It is a physical network that allows various components of a vehicle to be connected to each other. The Ethernet protocol offers greater features compared to the standard CAN protocol such as speed, flexibility, bandwidth, cost-effectiveness, and interoperability. The Ethernet protocol is also considered more secure compared to the standard CAN protocol since it relies on the TCP/IP protocol, which includes the sender and receiver's information in the message packet, which eliminates the need to broadcast each message. However, because Ethernet has long been the actual standard protocol for linking computers, years of expertise hacking computers may be applied to hacking vehicles. Furthermore, the Ethernet protocol might be vulnerable to attacks since it relies on TCP/IP protocol which focuses on communication for resource sharing. A variety of attacks have been implemented on TCP/IP protocol including source address attacks, sequence number spoofing attacks and mad authentication attacks (49). Another possible issue is that open-source protocol is not highly appreciated in the automotive sector due to common copyleft rules that compel companies to publicly reveal code updates and enhancements. Furthermore, the lack of validation of the open-source protocol implementations may cause further implementation issues, which potentially exposing serious vulnerabilities.

CHAPTER 4

Background & Related Work

4.1 Background

In modern electric vehicles, actuators and sensors are controlled through the electronic control units (ECUs). ECU is a device in modern vehicles which control electric subsystems. The ECUs are responsible for a variety of vehicle functions including engine control, braking, airbag deployment, door lock/unlock, antilock braking system (ABS), parking support system. Various network protocols have been proposed for in-vehicle communication between ECUs, such as controller area network (CAN), local interconnected network (LIN), media oriented system transport (MOST), automotive Ethernet, etc. (50). Based on the standards the automotive in-vehicle networks need to have atleast physical, datalink layer configurations, defined by the Open Systems Interconnection (OSI) model. CAN protocol is most commonly used for in-vehicle communication due to its robustness. Robert Bosch GmbH developed the CAN Protocol and published CAN 2.0 specification A and B in 1991 (51). In 1993, the international organization for standardization (ISO) released standard ISO 11898 for CAN protocol (51). Some of the advantages of CAN protocol are it decreased the cost of wiring in vehicles, had built-in error detection, increased robustness, higher speeds, and much more flexibility (52). CAN protocol consists of multiple abstraction layers. The two important layers are the physical Layer and the transfer Layer.

4.1.1 Physical layer

CAN is a broadcast-based communication protocol that is utilized in many different applications that have complex structure topology and require reliable communication between devices e.g. automotive, aerospace and trains etc. (53). CAN has 2 types of physical layer standards, low speed and high speed, which determine how the the CAN bus is structured and the speeds of the CAN network(50). The low speed standard has a baud rate up to $125\ Kbps$ that requires a single wired bus and devices that self terminate by $120\ ohm$ resistors on the CAN Bus (51). A high speed CAN bus consists of 2 wired half duplex serial network technology (51). The wires are called CAN

High (CAN-H) and CAN Low (CAN-L), which terminate at 120 *ohms* resistor. CAN is equipped to operate smoothly in different types of environments because of the electromagnetic shielding. CAN prevents electromagnetic interference (EMI) and protects communications from electromagnetic radiations that an automobile under goes daily. To prevent magnetic field radiation, the pair of wires are twisted. Furthermore, to prevent electric field radiation, a coaxial cable is used for the 2 wires. Also, another issue that can occur is electrostatic discharge (ESD) on the CAN Bus and it is prevented by the CAN transceiver.

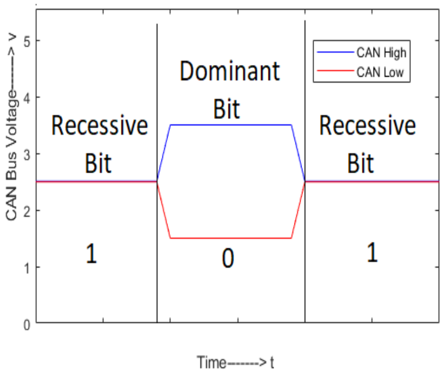


Figure 4.1: CAN-bus differential voltage representation.

One main key feature of the CAN Protocol is that it supports centralized communication control over ECU (53). ECUs can communicate with other ECUs on the network, and each ECU requires a micro-controller, CAN Controller, and CAN transceiver as shown in Fig. 4.2. The micro-controller controls when the message should be transmitted and analyzes messages received from the bus. The micro-controller is connected to the CAN controller which has two pins, transmitter (CAN-TX) and receiver (CAN-RX) (51). These two pins are connected to the CAN Transceiver and have digital voltages of 0V for logical '0' and 5V for logical '1'. The actual CAN bus does not support these voltages. Therefore, the CAN transceiver converts the digital logic voltages into a differential signal (53). The CAN transceiver drives and detects data communication to and from the bus. The differential voltages are outputs and consist of 2 states of voltages, dominant (or logical 0) and recessive (logical 1) (53). The differential voltages for the dominant (0) are 3.5V on CANH and 1.5V on CANL (54; 53; 55; 56; 45; 18; 57; 58). In addition, the differential voltages for the recessive (1) are 2.5V on both CAN-H and CAN-L. Figure 4.1 shows the CAN bus differential voltage representation. The two pins on the CAN Transceiver are connected directly to the bus which allows the ECU to transmit and receive messages from the bus (51). How the messages are transferred over, through the bus is discussed in the following section.

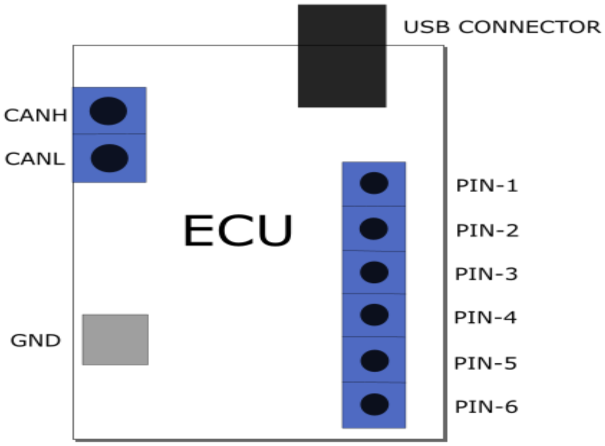


Figure 4.2: Electronic Control Unit.

4.1.2 Data link layer

The data link layer abstraction receives messages from the physical layer and transmits those messages using the CAN bus. This layer is responsible for timing synchronization, message framing, arbitration, acknowledgement, fault confinement, error detection, and signaling (59). These properties of the transfer layer are very important to the robustness of the CAN protocol which allows safe message communication between ECUs. These messages allow ECUs to communicate with any other ECU by the way of broadcasting the message to the shared CAN bus. Messages in CAN protocol are usually event driven which means an event must occur before any communication is established (60)(55). All other ECUs receive the transmitted message and depending on the parameters in the message, the ECU will either accept or reject the message. Communication in CAN protocol consists of 4 types of frames which are sent to all ECUs. These 4 types of frames operate differently and consists of different number of parameters. The types of frames are as follows, the Data Frame, Remote Frame, Error Frame, and Overload Frame (51). The 4 frames can be classified as error message frames or data message frames. The error message frames communicate errors that occur on the data message frames during the transmission on the CAN bus and they consist of the Error Frame and Overload frame. The data messages communicate actual data or request data to be communicated, which include the Data Frame and the Remote Frame. A data message frame, as shown in Figure 4.3, can have a maximum of 126 bits and consists of the following parameters:

- Start of Frame (SOF): The Start of Frame is a single dominant(0) bit which marks the start of a message and is used to synchronize the ECUs on the bus (51).

- Identifier (ID): The Identifier determines the priority of the message.The lower the ID value, the higher the priority and vice-versa (51). There are 2 types of ID's which are standard ID'S and extended ID'S. Standard ID's consist of 11 bits and extended ID's consists of 29 bits (51).

- Remote Transmission Request (RTR): The Remote Transmission Request is made up of one bit (51). If the bit is set to dominant (0), the message is considered as a Remote Frame (51). However, if the message is set to recessive (1), the message is considered as a Data Frame (51).

- Control: The Control consists of 6 bits which defines the type of data that will be transmitted (61). The first bit is the Identifier Extension (IDE) bit which determines if the ID is a standard ID (11-bits) set to dominant(0) or extended ID (29-bits) set to recessive(1) (61). The second bit is the Reserved Bit (R0) which is always dominant (0) and reserved for future needs (61). The next 4 bits are the Data Length Code (DLC) which determine the size of the data (in bytes) being transmitted (61).

- Data: The data consists of a maximum of 8 bytes depending on the set value of the DLC in the control setup (51). The data can send any type of information such as the temperature, speed and tire pressures.

- Cyclic Redundancy Check-(CRC): The Cyclic Redundancy Check consists of 16 bits, 15 message error correction bits and a recessive (1) delimiter bit (51). The CRC checks if the message transmitted is the same without any corruption and corrects any data corruption (51).

- Acknowledge (ACK): The Acknowledgement bits consist of the ACK bit and a recessive (1) delimiter bit (51). The ACK indicates an error free message has been sent (51). Every ECU that has received an accurate message overwrites this recessive bit from the original message as a dominant bit indicating success (51). If any of the ECUs detect an error, this bit is left as recessive indicating that there was an error and the message should be discarded and resent (51).

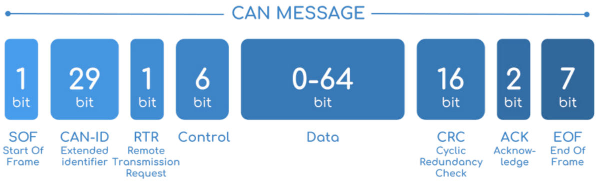


Figure 4.3: CAN message frame

- End of Frame (EOF): The End of Frame consists of 7 recessive (1) bits (51).
- Inter-frame Space (IFS): The Inter-frame Space consists of 3 consecutive recessive (1) bits which separate a data frame and remote frame (51). The proceeding bit will be regarded as the SOF bit of the next frame.

These message parameter fields allows CAN Protocol to be very flexible and have many different applications. The most important parameters from above are the ID, control, and the data fields.

One of the issues when using a single communication bus for transmitting and receiving messages is determining which ECU has control over the bus when two ECUs request bus access simultaneously. To resolve the issue, CAN Protocol implements bit-wise arbitration on the ID field of a frame to determine its priority (51). As stated above under the Identifier parameter, the lower the ID the higher the priority, and the higher the ID the lower the priority. For an ECU to win bit-wise arbitration, the ID's will be compared bit by bit and the dominant bit will always win the arbitration over the recessive bit (51). The ECU with the recessive bit will forfeit the arbitration until another opportunity arises (51).

4.2 Securing IVN - related work

Several security solutions against different kinds of message injection attack in the vehicular CAN bus, have been explored in prior work. The solutions can be classified according to (1) the specific technology used to address cyberattacks, and (2) the extent of the solution's coverage based on the OSI layer.

4.2.1 State-of-the-art classification

We divide these security solutions into five categories which are: time analysis based-solution, entropy based-solution, machine learning based-solution, fingerprinting based-solution and message authentication based-solution.

Time analysis based-solution Lee et al. (46) and song et al. (62) proposed a method of detecting an intrusion based on an analysis of the time interval of the CAN data by monitoring both the request time and the response time of the CAN data traffic. Although these models are considered to be lightweight, nevertheless they still have their limitations, especially when the in-vehicle environment change frequently; these limitations can be the continuous need for calibration and data update.

Entropy analysis based-solution Work by Müter et al.(63; 64) proposed an IDS that is based on monitoring the state of the traffic of the CAN bus and the entropy in in-vehicle networks. Despite this technique does not require any hardware modifications, nevertheless, this approach cannot detect irregular messages incoming.

Machine learning and deep learning based-solution Several methods that were recently proposed to detect intrusions on the CAN bus based on machine learning techniques (65; 66; 67; 68; 2). Such a method includes regression learning (65) and machine learning (66; 2). Additionally, the authors of (67; 68) proposed the use of the neural networks to detect an intrusion on the CAN bus. However, these methods are not feasible for a vehicular network due to the limited computing power of the ECUs to procedure the complex process.

Fingerprinting based-solution Multiple bodies of work have adopted physical fingerprinting techniques for IDS (53; 69; 70). For example, Avatefipour et al.(53) proposed a physical fingerprinting approach to detect the spoofing attack based on each physical ECU feature and the physical channel features. However, their technique can be failed when the channel length is increased which makes the physical ECU features are negligible. Additionally, work by (70) proposed a clock-based intrusion detection system (CIDS) that is used to fingerprint each ECU based on using the clock skew characteristic of ECUs. Although their approach is efficient, it is exhibited that CIDS can be failed against the spoofing attacker who can observe the clock skew and adjust his transmission accordingly (71).

Message authentication based-solution Several researchers have tried to secure a CAN channel from message injection attacks by implementing the cryptographic approach by appending message authentication code to the CAN packets (72; 73; 74; 75). For instance, Groza et al. (73) used the time synchronization of CAN messages with the help of key chains and proposed an authentication method. Work by Oguma et al. (72) proposed a novel verification server by implementing a master slave network while using CAN in order to authorize ECUs. Another researcher Lin et al. (74) proposed to send extra message to a CAN network to act as an heart beat signal to implement authentication. But it uses the available bandwidth of the CAN channel and can be an extra burden to the time critical network. Additionally, Herrewege et al. (75) proposed to append a unique hash message to the CAN data frame to authenticate CAN messages. Although these above mentioned methods provide some degree of security, but they need additional resources to implement on a CAN network.

4.2.2 State-of-the-art in OSI layer

Moreover, the state-of-the-art works can be described by their appearance in the layered OSI model. They are the data link layer solutions and physical link layer solutions.

4.2.2.1 Data link layer

Multiple bodies of work have adopted machine learning techniques for detecting intrusions on the CAN bus (76; 77; 78). For example, Theissler (76) proposed an anomaly detection system based on multivariate time series. An ensemble anomaly detector was made comprising of two-class and one-class classifiers in order to detect both known and unknown fault types in various driving conditions. However, his approach has limitations, especially when the in-vehicle environment changes frequently; these limitations can be the continuous need for calibration and data update. Other work by Barletta et al (77) proposed an IDS based on a combination of an unsupervised Kohonen Self-Organizing Map (SOM) network and k-means algorithm. The CAN IDs, time stamp, DLC and data field were used as features in order to identify attack messages sent on the CAN bus. Other work by Markovitz and Wool (78) proposed an anomaly detection system based on monitoring Constant fields, Multi-Value fields and Counter or Sensor fields of the CAN bus traffic. They used the Ternary Content Addressable Memory (TCAM) model to characterize those fields and build a model for the CAN bus messages based on those field types. Although they were able to achieve a low false-positive rate by evaluating their system on synthetic CAN bus traffic simulating 10 different message IDs. However, they didn't evaluate their system on actual attacks and against real CAN bus messages.

In (46) the author used the CAN bus remote frame to detect anomaly. The Idea was so simple that at random time each ECU uses the remote frame to request data from other ECUs. It will wait for a certain threshold amount of time for the response. If it does not hear back from the requested ECU, it flags an anomaly. It works really well for DoS and fuzzy attacks. In terms of impersonate attack, the experimental result showed that the reply characteristics will be different in attacked ECUs than the authorized ECUs. The main weakness of the method is the threshold time to work. After sending the remote frame the author waited for 7 messages for the response to take a decision about attack. It was not properly discussed.

In (5) the author proposed an intrusion detection system based on CAN bus message pattern. The proposed system has two stages in detecting anomaly in CAN bus data. The first stage is about learning the safe pattern and the second stage is to check the new upcoming CAN bus message pattern in the safe learnt pattern matrix. Where a pattern means sequential CAN bus arbitration ID. According to the author, if a CAN message with ID A comes just after a CAN message with ID B in the CAN bus, A and B has a safe pattern. In the test every CAN message where A and B comes one after another is flagged as safe. The main strength of this proposal is it is fit in a system with low computational power. The main drawback of this work is it is vulnerable to smart attackers.

Other researchers use the periodicity of CAN's message to detect anomaly (62) . Empirically, ECUs generates CAN message at a specific frequency. As a result, it is possible to detect anomalies in inter-arrival time when an external attacker injects messages. Another frequency-based IDS uses a one-class support vector machine to detect anomalies with high accuracy (79). However, real CAN message prone to variation, and often exhibits inconsistent inter-arrival time, reduces the reliability of these schemes.

Additionally, Minawi et al (80) proposed an IDS that utilizes machine learning and provides critical alerting features to protect vehicle operations. The CAN ID and Data field were the primary features used to determine if a message is benign or malicious. Furthermore, the Random Tree algorithm was used to achieve high accuracy in detecting DoS, impersonation, and Fuzzy injection attacks. Other work by Martinelli et al (81) proposed an IDS by considering the eight data bytes of the CAN packet as a primary feature to determine if a message is benign or malicious. Four fuzzy algorithms of classification were used: FuzzyRoughNN, NN, DiscernibilityClassifier and FURIA. These algorithms were applied to the eight bytes features and they were able to achieve 0.85 to 1 precession. Other work by Avatefipour et al (82) proposed an IDS for CAN bus based on the frequency of message IDs patterns that are transmitted in given normal traffic. A modified one class SVM was constructed and used based on a new meta-heuristic optimization algorithm called the Modified Bat Algorithm (MBA). Their IDS was evaluated on two datasets in the scope of CAN bus traffic anomaly detection. Although their IDS achieved a low false-positive rate. Nevertheless, it can't detect massages injection stacks. Additionally, Yang et al (83) proposed an IDS based on tree-based machine learning algorithms. The CAN IDs and the data field were used as features to detect threats both on the CAN bus and external networks. Although their system was able to achieve high accuracy by testing their IDS on two data sets for both intra-vehicle and external networks. However, their IDS has a high computational cost.

Several methods were recently proposed to detect intrusions on the CAN bus based on deep learning techniques (84; 85; 67; 86; 44; 87). Such a method includes an IDS based on a deep convolutional neural network (DCNN) to protect the CAN bus of the vehicle. The DCNN learns the network traffic patterns and detects malicious traffic without hand-designed features (87). Furthermore, work by Loukas et al (86) proposed a cloud-based cyber-physical IDS for vehicles by using the deep learning technique. Eight features were used to detect an intrusion which are network incoming and outgoing rates, CPU utilization, the rate of the written data to the disk, the time between two consecutive encoders, accelerometer readings, power consumption and the overall current drawn by the vehicle. However, by using RNN, they were able to achieve only 79% accuracy. Other work by Hossain et al (85) proposed a long short-term memory (LSTM) deep learning model-based on the intrusions on the CAN bus. The CAN ID, DLC and data field were used as features for in-vehicle CAN bus network attack. Other work by Seo et al (44) proposed an IDS model for the in-vehicle network based on GAN deep learning model. A large number of CAN IDs have been encoded and random fake data in the training process have been used instead of the real attack data. Although they were able to achieve an average of 98% accuracy. Nevertheless, their model was not able to distinguish anomalous traffic caused by normal malfunctioning of electronic components from anomalous traffic caused by intentional attacks by hackers. Additionally, Hanselmann et al (84) proposed an IDS based on a neural network architecture that is trained in an unsupervised manner. The CAN IDs and timestamp were used as features in order to detect intrusions and anomalies on the CAN bus. Although they were able to achieve high accuracy by evaluating their system on synthetic CAN bus traffic. However, they didn't evaluate their system on actual attacks and against real CAN bus messages.

In (88) the author distinguished all the possible cyber attacks to the CAN bus into three categories. They are (i) Weak Attacker, (ii) Medium Attacker and (iii) Strong Attacker. All the attacks were named from these attackers as random ID attacks, all zero ID attacks, replay ID attacks, Spoofing ID attacks etc. To prevent those attacks, a two stage novel Intrusion detection system was proposed. The first stage is the rule based IDS and the last one is the deep learning based IDS. Each CAN message first enters the first stage of IDS. This stage IDS considers a few properties to detect anomalous CAN messages. They are valid arbitration IDs of CAN message, time interval between two CAN message, message frequency of certain arbitration ID etc. The message marked as safe in this step enters the last stage of IDS for another verification. These last steps used deep learning to classify the anomalous CAN message. Artificial Neural Network algorithm with 5 hidden layers has been used for this purpose. To implement this deep learning algorithm four features have been selected. They are CAN message arbitration ID, message frequency in the past one second, relative distance between arbitration IDs and change in system entropy.

Although the above solutions provide some degree of security as shown in table 8.2. However, in addition to the additional resources required and complex computation costs needed, the deep learning approach is not sufficient for a vehicular network due to the limited computing power of the ECUs to procedure the complex process. Unlike prior work, we propose a lightweight IDS based on a new eight features. We find that the newly explored eight features are significant features to detect attacks on CAN bus messages with a high detection rate with minimal time without any modification in the standard procedure of the CAN protocol.

4.2.2.2 Physical link layer

Multiple bodies of work have been proposed by the academic community for preventing attacks on automotive vehicular network reported in past years (89), which are caused by CAN protocol's inability to identify sender in a network. One approach to integrate security to CAN protocol is to apply cryptography i.e. encryption and decryption scheme to CAN frames (90). While this approach can provide confidentiality of data (one important concept of CIA triad), but due to the limited data rate and the urgency to satisfy real time system requirements, the CAN protocol is not suitable for cryptographic approaches (91). Moreover for this kind of solution, key management is

Table 4.1: A comparison of state-of-the-art IDSs using machine learning techniques

Ref no	Machine learning algorithms	Features
(76)	Ensemble classifier	Timestamp
(77)	SOM and k-means	CAN IDs, timestamp, DLC and data field
(78)	One-class classifier	Constant, Multi-Value, and Sensor fields
(80)	Random Tree	CAN IDs and data field
(81)	NN, Discernibility Classifier, and FURI	Data field
(82)	One-class-classifier	The frequency of message IDs patterns
(83)	Tree-based	CAN IDs and data field
(87)	DCNN	The network traffic patterns
(86)	RNN	Network in and out rates, CPU utilization, written data rates, time between two consecutive encoders, accelerometer readings, power consumption, and the overall current drawn by vehicle
(85)	Deep Learning	CAN IDs, DLC, and data field
(44)	GAN Deep Learning	CAN IDs
(84)	DNN	CAN IDs and timestamp
(2)	SVM, KNN	CAN IDs and data field

an extra burden for a time critical system (7).

To ensure integrity in the CAN bus one approach is to implement message authentication scheme (92) by including a message authentication code (MAC) inside CAN frame. While it makes the CAN bus secure but according to the standards, the least size of the MAC is 64 bit to prevent collisions (7). So, the challenge of implementing the MAC based approaches is to add 64 bit MAC along with the data that needs to be transported to the network where the data field can only hold up to 64 bits of data (figure 4.3). To overcome the approach, researchers proposed two kind of MAC implementations. one is instead of using 64 bit MAC, they were using a truncated MAC to include integrity to CAN protocol (93; 94; 92) and the other approach is to use CAN+ protocol, an improvement of the existing CAN (95; 96) where additional data can be sent in time intervals to authenticate CAN messages. For example, researchers in crafted a 4 byte MAC and put it into the data field of the CAN packet to authenticate CAN message. The disadvantage of truncating CAN data field to include MAC (93; 94) is, it limits the size of data payload to be transmitted in a CAN packet and restrict the CAN protocol to transmit 8 bytes data payload. The proposed works in (95) sends two CAN messages where one contains the data payload the other one contains the MAC address. The approach resolves the issues originated by the truncated MAC approaches but it uses the limited traffic bandwidth of CAN network (1 Mbit/s) (6) as it needs to send two packets of data to securely send a single CAN data payload.

Apart from the CAN message authentication techniques, researchers have considered to fingerprint CAN senders by using physical unclonable characteristics such as clock skews (97) and voltage (7; 6; 98). The main idea of this approach is to identify the source of CAN transmitters.

Table 4.2: Computational complexity of common state-of-the-art statistical features

Feature name	**Equation**	**Time complexity**
Minimum	$min = min(x_i)$	$\Theta(n)$
Maximum	$max = max(x_i)$	$\Theta(n)$
Mean	$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n} = \frac{x_1+x_2+...+x_n}{n}$	$\Theta(n)$
Variance	$s^2 = \frac{\sum_{i=1}^{n}(x_i-\overline{x})^2}{n-1} = \frac{\sum_{i=1}^{n} x_i^2-n\overline{x}^2}{n-1}$	$\Theta(n^2)$
Skewness	$skewness = \frac{\sum_{i=1}^{n}(x_i-\overline{x})^3}{(n-1)*\sigma^3}$	$\Theta(n^2)$
Kurtosis	$kurtosis = \frac{\mu_4}{\sigma^4}$	$\Theta(n^2)$

The concept is adopted from the famous physical layer identification (PLI) (99) technique where the unique characteristics of transmitters are extracted to link the physical signals to the senders. The techniques for CAN PLI can be classified into two categories.

Clock skew based fingerprinting: The quartz crystal clock determines the different clock frequencies on an ECU, resulting in random clock drifts which can be used to uniquely identify an ECU. Cho and Shin proposed a Clock-based IDS (CIDS) (97) which exploits the intervals of periodic message to estimate the clock skews as the fingerprint of the transmitter ECU. The idea was used to estimate clock behaviors of ECUs to detect the intrusion and identify the source of the message. They have tried to fingerprint ECU and used it to detect anomaly. The proposed method uses the periodic behavior of CAN messages to fingerprint each ECU. It actually creates a base clock behavior of each ECU and uses that as a reference for detecting anomaly. Any deviation from the base clock behavior is marked as an anomaly. The author has performed their experiment on not only CAN bus prototype but also on real vehicles. The author in (100) indicated one drawback of this proposed method. The method uses CAN message periodicity to fingerprint ECU. But in the vehicle, it is possible to send CAN messages with different arbitration IDs from a single ECU. In that case a single ECU will have a different fingerprint. And it opposes the initial assumption of the method. Moreover, this method is only effective in a temperature-stable environment(101). On the other hand, the strongest part of the work is, it can handle most of the attacks.

Voltage based fingerprinting: Authenticating the CAN message transmitter based on the unique and immutable physical characteristics such as the voltage, is termed as physical fingerprinting. This area of research has gained popularity now a days where utilizing the voltage characteristics is the core idea. For example, Kneib et al. (101) used voltages for fingerprinting ECUs, utilizing rising edge, falling edge of the dominant bits. The framework achieved an accuracy of 99.85% in identifying ECUs by using statistical features like mean, standard deviation, variance, skewness, kurtosis, root mean square, maximum and energy etc. Researchers in (6) extracted time domain and frequency domain statistical features using voltages captured from the ECUs and proposed a neural network based ECU classifier. They achieved an accuracy of 98.3% on an experimental setup using microcontrollers. Authors in (7) proposed an edge based identification method using voltage collected using picoscope (software defined oscilloscope) and a naive bayes classifier. As a feature they used statistical time domain features such as mean, variance, skewness, kurtosis, radio max plateau, plateau, overshoot height, irregularity, centroid, flatness, power and maximum. Similar work has been proposed in (102) that uses 10 time domain features and 10 frequency domain features and achieved an accuracy of 98.94 % accuracy at maximum while voltage data is collected using an oscilloscope at a sampling rate of 2 GS/s. Bellaire et al. (98) proposed a machine learning based ECU fingerprinting framework by handcrafting signal processing features on voltage data such as transient response length, maximum transient voltage, energy of the transient period, average dominant bit steady-state value, peak noise frequency and average noise. Similar kind of approaches are also proposed in (103; 104).

CHAPTER 5

Multilayered Framework

5.1 Generalized intrusion detection system architecture

Although the IVNs are not security proven at least in terms of their architecture, nevertheless they have been used in automotive systems for decades and the production of vehicles is increasing day by day due to the high market demands. For providing a better passenger experience, these vehicles are equipped with intelligent software and are treated as cyber-physical systems. As the safety of passengers is directly related to the security IVN, there are two popular approaches in general that can increase the security of safety-critical IVNs (i.e. CAN bus). The first one is implementing an encryption algorithm to secure the CAN bus channel. The other one is to monitor the network traffic data and/or analog CAN signal and report unusual behavior. One process of providing solutions by using CAN data is data driven technology (i.e. statistical machine learning and deep learning), which is currently becoming famous in the domain of network security. In this section, a brief description of the intersection point between machine learning and IVN security will be discussed. The section will present an overview of the machine learning pipeline when it is used in IVN security. The overall diagram of the machine learning pipeline is displayed in figure 5.1.

As the CAN bus in modern vehicles is exposed to a huge number of threats and becomes an attractive target for attackers, the need for an Intrusion Detection System (IDS) for the CAN bus is becoming one of the most important security components in modern vehicles. The existing IDSs for the CAN bus can be divided into behavior-based IDS and fingerprinting-based IDS. Where the behavior-based IDS lies on the data link layer and the fingerprinting-based IDS lies on the physical layer. The behavior-based IDS is used to monitor the network traffic data in the data link layer and report unusual behavior while the fingerprinting-based IDS is used to utilize the physical characteristics of the analog CAN signal in the physical layer and report any anomalies. One process of using the network traffic data and the physical characteristics of the analog CAN signal in order to build an automated solution is machine learning technology, which is currently

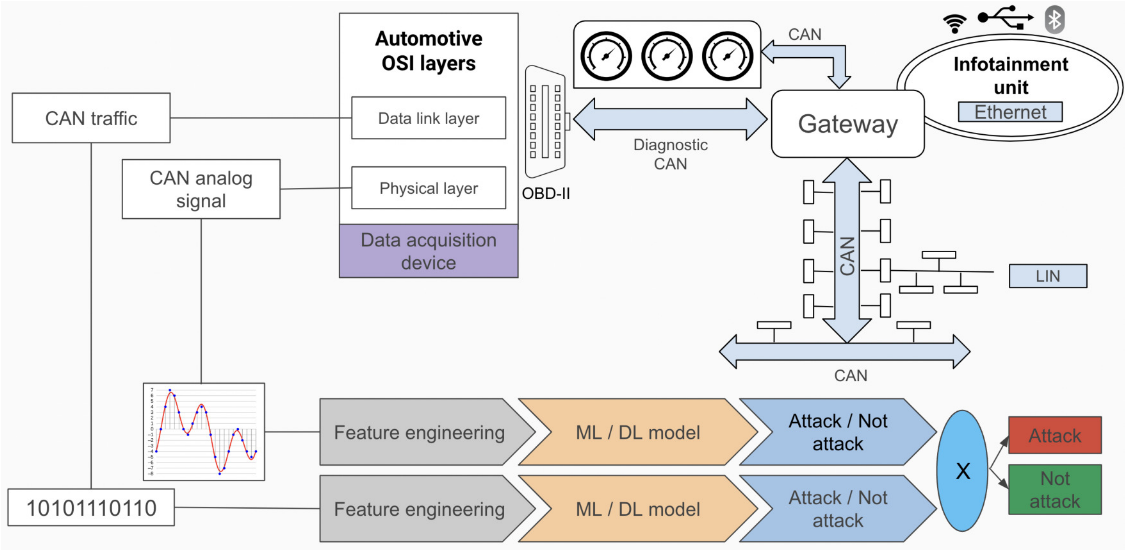


Figure 5.1: Machine learning pipeline in CAN security

becoming one of the most popular technologies in the domain of security.

5.1.1 CAN data acquisition

The first step for implementing a machine learning system to defend CAN bus against cyberattacks, is to acquire CAN data. The data acquisition is a complex process where the researcher needs to access the CAN bus of the vehicle first. To access the CAN bus, a standardized connector to a vehicle is needed and is called OBD-II. Since 2006, the OBD-II port was required for every consumer vehicle and was mandated by law in the USA. The OBD-II port has 16 pins and they are summarized in the Table in 8.1. The CAN - H and CAN - L pins can be wired to a microcontroller unit like an Arduino with the help of a CAN bus shield (CAN bus transceiver) to capture raw CAN bus data traffic. There are also few commercial dongle interfaces available that can be used to capture raw CAN bus data like panda OBD-II OBD2 interface, OpenXC, OBDLink SX, etc. The streamed CAN bus raw data can be captured in the data link layer of Open Systems Interconnection (OSI) model as network traffic and is presented in figure 5.2. As the data acquisition from an actual vehicle needs to be accessed through the OBD-II port, preparation should be taken for safety according to (105). To avoid this complex task people have also used car simulators with a virtual CAN like ICSim which is becoming popular day by day. Researcher's in (105; 106; 107) have gathered CAN bus traffic data using the ICSim simulator. The simulator simulates a car's instrument cluster, e.g. speedometer, its controls, e.g. throttle and steering, and the corresponding CAN traffic. It is built based on the SocketCAN - a Linux-based library for the CAN network. According to (107), the CAN traffic generated from ICSim seems realistic like real vehicular traffic.

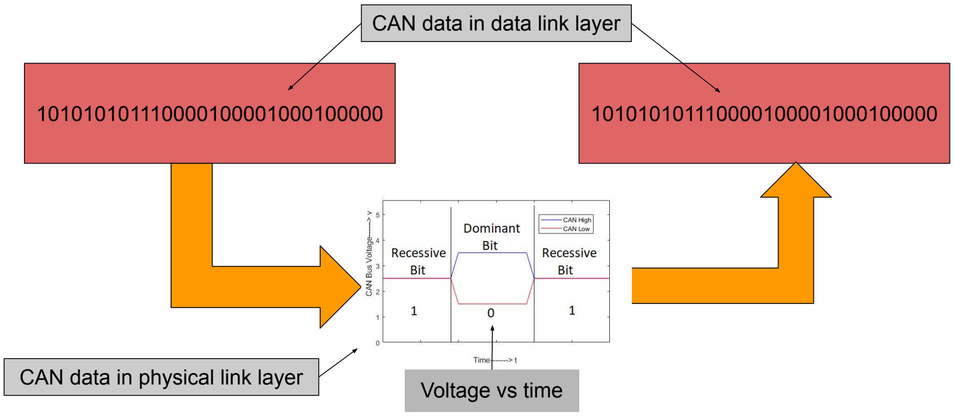


Figure 5.2: CAN bus data acquisition in data link layer and physical link layer

On the other hand, the CAN bus data can be collected in the physical layer of the OSI model shown in figure 5.2 that is the analog data (voltages). To collect the analog CAN signals data TivaC, TM4C123GXL, MCP, TriCore, NXP MPC, STM32, and oscilloscope instruments can be used (53; 54; 108; 109; 110; 69; 111; 112).

Table 5.1: Physical layer features

Feature Type	Feature Name
Time-Domain Features	Maximum, Minimum, Mean, Variance, Std-Dev, Average Deviation, Non-Negative Count, Zero Crossing Rate (ZCR), Root Mean Square, (RMS), Amplitude, Energy, Power, Skewness, Kurtosis
Frequency-Domain Features	Spectral Std-Dev, Spectral Entropy, Spectral Spread,Spectral Flux, Spectral Roll off, Spectral Skewness,Spectral Brightness, Spectral Kurtosis, Spectral Flatness, Spectral Centroid, Irregularity K
Signal Descriptive Features	Ratio Max Plateau, Plateau, Overshoot Height, Maximum
Deep Features	Deep features were extracted using Recurrent Neural Network (RNN)

5.1.1.1 Available public dataset

While capturing CAN bus traffic in the data link layer is pretty straightforward, there are few variations when analog CAN signals are captured. When talking about analog CAN signal, the researchers in (53; 54; 108) have used an oscilloscope with 2 (GS/s) in order to collect 144k samples,4.147M samples, and 3.15M samples of the CAN signals where the CAN signals are captured of different CAN cable types with various lengths, as well as different number of ECUs with the same input of the CAN bus message. Similarly, work in (110; 69) have used an oscilloscope with 2.5 (GS/s) and 1 (GS/s), in order to record the CAN signals frames. Other work in (112) have used

Table 5.2: Available public datasets

Dataset Reference	CAN data source type
(46)	KIA Soul
(113)	Ford Escape
(114)	Synthetic CAN bus data

an MCP2515 controller and an MCP2551 transceiver and by using a 20 (MS/s), they were able to capture 56.56k frames of the CAN signals where the CAN signals were recorded at the output of ten ECUs. Similarly, work in (109) have also used the MCP2515 controller and the MCP2551 transceiver to capture the CAN signal and they were able to capture 48.128k frames of the CAN signals of five ECUs by using only 2 (MS/s). Additional work in (111) have used TM4C123GXL microcontroller integrated with TivaC instrument and by using 50 (MS/s), they were able to collect 10k frames of the CAN signal of ten ECUs. Unfortunately, none of the aforementioned datasets are publicly available for researchers. On the other hand, when talking about the data link layer, the CAN traffic is captured and is publicly available for researchers. The CAN datasets are presented in table 5.4.

While real vehicle data provides a realistic nature of the attacked or attack free condition of a vehicle, synthetic CAN bus data can provide different types of attacked scenario that can be a challenge to gather from a real vehicle. By design, a vehicle is very complex system and it takes a lot of time to learn about the system before performing attacks. On the other hand synthetic CAN bus data can be cost and time efficient to collect and conduct further research. But, the data from real vehicle provides actual characteristics under an attacked situation.

Table 5.3: OBD-II pinout description

Pin	Description	Pin	Description
1	Vendor option	9	Vendor Option
2	J1850 Bus +	10	J1850 Bus
3	Vendor Option	11	Vendor Option
4	Chassis Ground	12	Vendor Option
5	Signal Ground	13	Vendor Option
6	CAN (J-2234) High	14	CAN (J-2234) Low
7	ISO 9141-2 K-Line	15	ISO 9141-2 Low
8	Vendor Option	16	Battery Power

5.1.1.2 Data acquisition using test-bed

According to the state-of-the-art, the researchers in (111) were able to build a testbed consisting of ten Arduino Unos, each with two identical CAN shields, and use a TM4C123GXL microcontroller combined with a TivaC instrument with using 50 (MS/s) in order to capture CAN signal frames. Other work in (112) and kneib2020easi were able to create testbeds in order to collect the CAN signals frames from the Fiat 500 and the Porsche Panamera vehicles. The testbeds in (112) and kneib2020easi are consisting of ten Arduino Unos and five Arduino Unos, respectively. Each Arduino Uno is equipped with two identical CAN shields and both work in (112) and kneib2020easi have used an MCP2515 controller and an MCP2551 transceiver using 20 (MS/s) and 2 (MS/s), respectively in order to record CAN signal frames from both the Fiat 500 and the Porsche Panamera vehicles. Additionally, two Raspberry Pis where each one is equipped with a CAN Shield were connected to increase the number of ECUs. One of the Raspberry Pis was connected to the OBD-II port and the second one was connected to the CAN bus in order to capture CAN signal frames.

5.1.2 Data processing

Data processing is a step of using domain knowledge to extract information (characteristics, properties, attributes) from raw data. The success of machine learning depends on this information which makes the learning easier if the extracted characteristics correlate with the target class. On the other hand, if the class is a very complex function of the extracted information, you may not be able to learn it properly. Often, the raw data is not in a form that is amenable to learning, but you can construct features from it. This is typically where most of the effort in a machine learning project goes. It is also one of the most interesting parts, where intuition, creativity, and “black art” are as important as the technical stuff. To implement machine learning, it is important to choose attributes that are different between the benign CAN bus traffic and the malicious CAN bus traffic. This subsection will provide an overview of the feature extraction, engineering & selection process when applying machine learning in the domain of CAN bus security.

5.1.2.1 Feature extraction

To extract important information from the CAN data, first, it is important to understand the CAN bus traffic. For standard CAN bus protocol, each ECU sends at most 111 bits CAN message in the bus. Out of 111 bits, 11 bit is known as arbitration ID, 0-8 byte data payload, and the remaining consists of start bit, RTR bit, CRC bit and end bits, etc. The figure 5.3 shows a packet of CAN messages. These portions of CAN bus messages have been used as machine learning features for years. Especially, CAN arbitration ID and the data fields have been used in (80; 82; 83) for

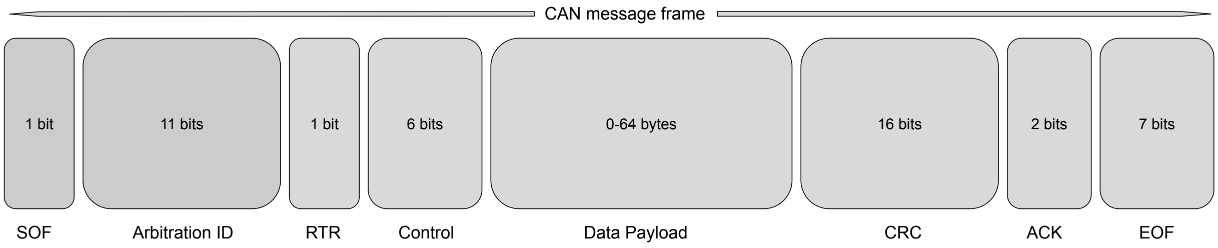


Figure 5.3: CAN bus message frame

detecting benign and malicious CAN bus messages. Apart from these traditional features, authors in (76) used the timestamp of each message and used it as a feature for the classification.

5.1.2.2 Feature engineering

The above-described sub section shows that there are few features in the CAN bus data. Obviously, too few features can not show the complexity of the data, which will affect the intrusion detection performance. So, to increase the performance of machine learning models the idea is to generate features that represent hidden differentiable characteristics of the CAN bus. To do this, one popular approach is to construct entropy-based features (115; 66), where two or more features are used to generate a new feature that improves the attack detection rate of the model. Another popular technique is to extract graph theory-based features from CAN message (3; 116). This interesting approach considers a window of CAN messages to construct a graph and then extracts graph-based attributes like a number of edges, nodes, degree of nodes, etc to use as machine learning features. For example, in (116), the author took every 200 CAN message into consideration and built a graph using them. He further used these graphs to gather features i.e. number of edges, number of nodes, radius, diameter, density, reciprocity, average cluster coefficient, and assortativity coefficient. The result section shows the effectiveness of feature engineering in CAN security. The graph-based features approach shows better performance than the traditional CAN bus message features by at most 1.44% when using the same machine learning algorithm as the attack detector.

When working with analog CAN signals, feature engineering is a mandatory step. According to the state-of-the-art, statistical features were extracted from CAN analog voltages and used in intrusion detection. Fingerprinting based IDSs (53; 54; 108; 109; 110; 69; 111; 112) utilizes this approach in order to detect any anomalies. Where these statistical features represent the signal fingerprint in addition to the associated ECU. These statistical features can be classified into four main types which are time-domain features, frequency-domain features, signal descriptive features, and deep features. These statistical features are presented in table 5.3.

5.1.2.3 Feature selection

The last important task before training the model is to select the appropriate features. This process can be done manually or automatically. The main goal of this sub-process is to select the features that correlate with the target variable. People sometimes think this process is the least important step than feature extraction or engineering and does not affect the accuracy of the model. In fact, by using irrelevant features the model memorizes rather than generalizes the problem from the training data. Thus they decrease the accuracy of the model. In CAN bus security research, the two most common approaches are selecting features based on the feature differences (116) or based on a feature rank score on the model prediction (83). For example, in (116), the author plotted the graph features as box-plots for attack free and attacked graphs and selected 7 features out of the 8 features which have high feature difference in terms of data distribution. The feature, ‘a number of nodes’ is the same for both the attacked and attack-free graphs, hence does not have a significant effect on the model’s prediction. To determine a ranking score between the features using a machine learning algorithm on the training data is another way for selecting the features. The tree-based algorithms calculate the importance of each feature based on every single tree and then average the output of the trees to make the result more reliable. Additionally, different traditional feature selection methods such as information gain, entropy, and Gini coefficient can be utilized also for this purpose.

5.1.3 Algorithms to detect CAN bus attack

In a machine learning project, the next task after feature selection is modeling. It is the process of constructing a mathematical equation by iterating a set of data to find a perfect line that can either categorize the data or predict future values. This process can be broken down into three parts, first one is algorithm selection, then hyperparameter tuning and finally model validation. In the following subsections, the description of these three procedures will be explained in terms of CAN bus data.

5.1.3.1 Algorithm selection

The selection of machine learning algorithms in CAN data security is dependent on the quality of data and available differential features. In a modern vehicle, there are 70-100 ECUs and they communicate with each other very frequently, hence we can assume there is a large amount of data available for detecting CAN bus attacks. So, KNN, decision trees, or kernel SVM algorithms can be used. (116; 2) used KNN & kernel SVM, (80; 83) used decision trees to detect CAN bus attacks and achieved high accuracy in detection. Author in (84; 44; 85) used deep learning-based algorithms to detect CAN bus intrusion which is computationally expensive as shown in Table 4.

Table 5.4: Experimental environment for deep learning-based algorithms

Paper	CPU	GPU	RAM
(84)	3.50 GHz	N/A	32 GB
(44)	3.60 GHz	yes	32 GB
(85)	2.20 GHz	yes	16 GB

When working with the physical CAN signals, according to the state-of-the-art fingerprinting based IDSs (53; 54; 108; 109; 110; 69; 111; 112), work in (111; 112) have used Artificial Neural Network (ANN) algorithm to detect any anomalies in the CAN bus. Similarly, work in (54; 53) have also used ANN algorithm to detect any anomalies in the CAN bus, where 70% and 65% of the collected samples were used for training the model and the 30% and 35% of the collected samples were used for testing. Other work in (108) have used a recurrent neural network with long short-term memory (RNN-LSTM) to detect any anomalies in the CAN bus. Additional work in (110; 69) have used Support Vector Machine (SVM), ANN, and Bagged Decision Tree (BDT) to detect any anomalies in the CAN bus. Similarly, work in (109) have used the same algorithms used in (110; 69) in addition to Logistic Regression (LR) and Naive Bayes to detect any anomalies in the CAN bus.

5.1.3.2 Hyperparameter tuning

To train an unbiased, highly efficient machine learning model, it is important to tune the hyperparameters of the selected algorithm and select the parameters before moving to model validation. In CAN bus security, the state-of-the-art papers do not explain the hyperparameter tuning, hence an example is given how the hyperparameter can be selected using the KNN algorithm. This will help to interpolate the model and provide an idea of how tuning can be done. In the KNN algorithm, k is a parameter that needs to be tuned. But unfortunately, there is no conventional way to tune the parameter. The go-to approach is to try with different values of k, monitor the training and test error rates and finally choose the k with a relatively smaller amount of test and training errors among them. Figure 5.4 shows the results of applying the KNN algorithm with different values of k with the selected features in paper (116) while using the same public dataset (46). According to the plot the value of k = 4 is chosen on the training data as it provides the smaller training and test errors. But the whole hyperparameter process can be automated as shown by the authors in (117; 118).

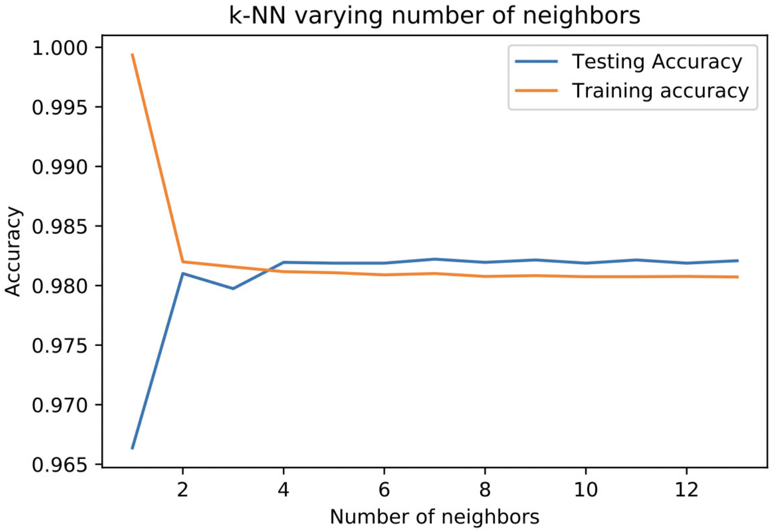


Figure 5.4: Tuning number of nearest neighbors (k) in KNN algorithm

5.1.3.3 Model validation

Once the hyperparameters are set, the model is finally trained and ready to be validated. It is always advisable to validate the model with an unknown set of data. This will help the researcher to understand whether the model has been generalized from the training data or it has only been memorized. To execute this step, the total data set can be divided into three chunks into a ratio of 60%:20%:20%. (i) training data, (ii) test data & (iii) validate data. The training data and test data are used till the hyperparameter tuning and the validation dataset will be used to measure the final performance of the model. For example, with a k of 4, the KNN achieved an accuracy of 98.00 % when working with 8 graph-based CAN feature from (116).

5.1.4 Challenges of ML in IVN research

The first challenge of applying machine learning to secure IVN is to supply quality network data to the model. The CAN data acquisition is not a straightforward task where access to the network of the vehicle is required to capture such data. Acquiring network data from an actual vehicle is a complex process, hence people usually build vehicular networking prototypes which is also challenging. Because this process requires extra analog data acquisition devices such as an oscilloscope which is expensive is not portable. The oscilloscope can be replaced by low-cost microcontrollers but they need to be programmed additionally. To capture quality physical layer data, a higher sampling rate should be used to read analog voltage and it is also a challenging task (109). On the other hand, while using automotive network's data link layer data in the machine learning model, feature engineering needs extra attention because of the lack of direct features. Another challenge is monitoring and detecting any anomalies in the CAN bus in real-time, where the majority of research so far has been done to detect intrusions using datasets and is not suitable for real-time detection. Moreover, there is no well-known and widely recognized dataset that can be used to assess the effectiveness of intrusion detection systems for the vehicles and unfortunately, no datasets for the physical layer are publicly available for the researchers. The limited computing power of the ECUs to process a complex machine learning to detect any anomalies in IVN are also challenging.

5.1.5 Future opportunities in this domain

One critical threat to vehicles nowadays is malware -a Malicious software designed to gain unauthorized access to data or to disrupt computer operations. Malware can infect vehicles via a variety of interface vulnerabilities including wireless connections with roadside networks, Wi-Fi hotspots, Internet connectivity, Bluetooth, and culler networks like 5G. It can also infect vehicles through cell phones, removable media, iPods and laptops that are associated directly with vehicles (25). Malware can cause a wide range of disturbances and harm to the vehicle system once it is inside the vehicle (119). Some examples of how malware affects the vehicle's regular operations are: tampering with the in-car radio so the driver can't turn it on, locking automotive functions such as locking the car doors, occupying the memory and CPU cycles of the ECUs, and disabling the vehicle's safety features (119). The above-mentioned examples all believe that vehicle systems are a high priority that must be appropriately handled in order to effectively safeguard them. According to the state-of-the-art (120), machine learning approaches are successful in production level applications for defending malware. and we think the machine learning techniques can be used in the future to address such cases and protect the vehicle systems from malware effectively.

5.2 Multilayered framework

This research is divided into mainly two parts. In one part intrusion detection systems are based on performing behavioral analysis of the in-vehicle network. The second part tries to link the signal to the sender by characterizing the unclonable signals that flow through the CAN bus physical channel. The whole concept is described by the figure in 5.1. Both the approaches follow the pipeline. It starts with acquiring CAN bus data. For the behavioral fingerprinting, it is the in-vehicle CAN bus network traffic data. And for the physical fingerprinting analysis, it is the physical signal acquired from the CAN bus channel. Then different features are extracted from the data to gain more information about the system. Finally, the features are used to train a model to detect vehicle cyber attacks in different layers. In this section intrusion detection in two layers is discussed briefly.

5.2.1 Behavioral fingerprinting

The first layer is to build a defense mechanism system by learning the general behavior of the in-vehicle network and continuously monitoring the nature of the network. Any deviation from the natural behavior can be flagged as an attack. Because the motivation of the attacker is to inject malicious messages into the network and misguide the vehicle. The beauty of this approach is that it can easily detect DoS, fuzzy, and spoofing attacks (46; 3). Few researchers have used this approach to detect CAN bus attacks. However, we will propose novel behavioral methods to detect CAN bus attacks.

To generalize the in-vehicle network behavior, the sequential CAN bus message will be transformed into temporal graphs. Then the graph theory-based features will be used to estimate the normal behavior of the network. The reason for using graph theory is that it has been used for anomaly detection in the industries like finance, computer & social networking, data centers, etc. for decades (121). In our case, the graph theory properties will be used to model the network behavior and will be used as a base to detect CAN bus attacks.

Generally, in the area of network security, machine learning-based & statistical based approaches are becoming popular day by day (122; 123; 124). Because the fundamental concepts of machine learning algorithms are a good fit for this purpose since they analyze the data to generalize system behavior. These algorithms try to mimic the human learning system by finding patterns from past incidents and taking decisions according to the knowledge. Based on this, unwanted system behavior can be detected. That's why a statistical machine learning algorithm is a perfect approach for attack detection of any kind of system.

5.2.2 Physical fingerprinting

Behavioral analysis technique works great in detecting spoofing, DoS, or fuzzy attack, while the localization of the compromised ECU still remains a challenge in this approach. However, by default, in a CAN bus network linking the CAN packet to the sender node remains a challenge. Because there is no sender information in the CAN message packet. There is CAN arbitration ID which rather uses a conflict resolver by prioritizing each message. In order to localize the compromised ECU, physical fingerprinting techniques are applied that use physical signal attributes of the received packets to link to the sender.

Although, the CAN packets do not have the sender information embedded in it, there is two natural random artifact that is unique in CAN bus channel.

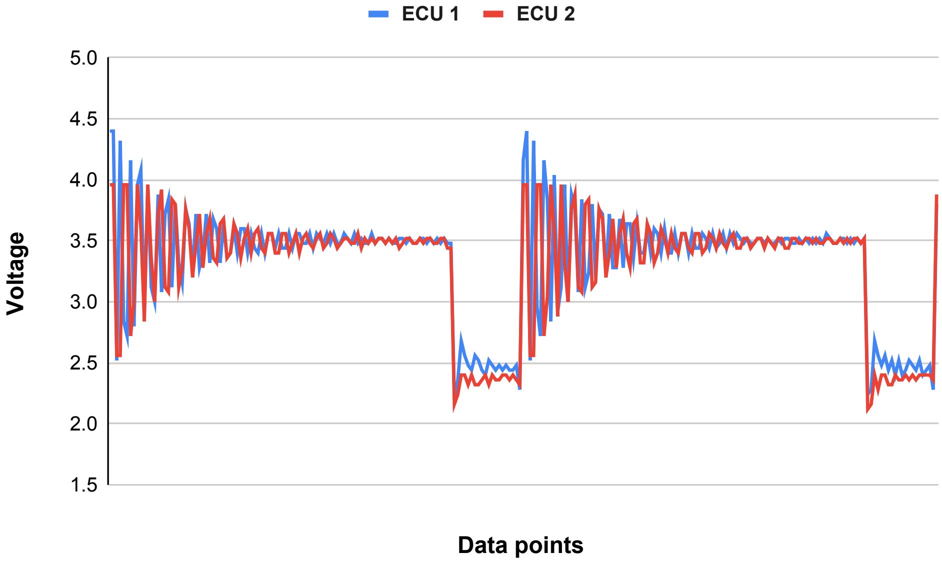


Figure 5.5: ECU signals are influenced when two different ECUs are added at the same location on the same CAN bus.

- Device artifact
- Channel artifact

The electronic device artifact originates in the manufacturing process due to material asymmetry. On the other hand, the channel electrical signal gains some distortion due to the channel it propagates as well. Therefore, the uniqueness and randomness of the material of the electronic device can be found in the physical raw signal it generates. And again the unique channel also imposes its' artifacts to the physical channel as well. Figure 5.5 shows if two ECUs are connected to the same point and send the same data packet, the physical signal will be different due to the different electronic device material artifact.

As these characteristics are unclonable and unique thus can be used to build profiles of receiving packets to the sender easily. These profiles can be used to find our compromised sender in a CAN bus attack scenario. This section provides step by step description of the intrusion detection system using physical fingerprinting of CAN bus signals.

CHAPTER 6

Behavioral Fingerprinting via Graph Theory - Statistical Approach

As mentioned in chapter 4, CAN protocol has some serious security breaches in its core. The protocol actually works like a broadcasting system where it contains no mechanism for checking the identity of sender. Several researchers have tried to provide solutions to increase security of CAN bus (70; 125; 46; 126). Most of them work for a certain types of attack situations. Currently The increasing amount of research work on autonomous vehicle inspired us to work on detecting anomaly in CAN bus for autonomous vehicle.

Several researchers have proposed different solutions for defending against cyber attacks in a vehicular system. The attack can be performed through either a weak or an strong agent. Weak agent can spoof CAN bus by injecting with arbitration ID with high priority (ID 0000, DoS Attack) or with any arbitration ID (Spoofing or Fuzzy Attack) (70; 46). On the other hand, a strong agent uses two attackers at the same time to perform attacks. In that case one attacker tries to damage the targeted ECU and the other attacker sends CAN bus message with the targeted ECU arbitration ID (70; 46) . In (5) the author proposes an anomaly identification technique based on recurring sequential message IDs. The idea was so simple that in a CAN bus system, if two message IDs come one by one, they are more likely to appear again. But this is vulnerable to weak and strong attacks. We have considered this concept and tries to find complex relationship about CAN bus data.

Graph is popular for finding complex relation about data. And is used hugely in anomaly detection techniques. Graph based anomaly detection techniques are used in the industries like finance, fraud detection, computer and social networking, data centre monitoring etc (121) . On graph based structured data it is pretty easy to find out the unusual substructure (127). The unusual substructure can be used flagged as an anomaly. First, We have tried to construct graph from CAN bus data and then search unusual behaviour to flag as anomaly. Our experimental result showed significant success in detecting anomaly in this approach.

On a graph based on an attack free CAN bus data, the distribution of edge remains normal in nature. On attack scenario, significant differences were found in terms weak attacks and strong attacks. In terms of a strong attack, if we consider distribution of maximum degree of a set of graphs, the attack root cause can be found. In summary we can claim that distribution of graph properties based on CAN bus message can provide a strong protection against all kind of cyber attacks.

In particular the major contribution of this work is as follows:

- It is the first ever Graph based cyber attack defence system for CAN bus communication.
- It is the first ever Chi - Square implementation for detecting attack in CAN bus communication.
- We propose a four stage IDS for detecting cyber attack on CAN bus system. Here, Graph based approach was used to find out patterns in the data set and chi square test is used to compare two data pattern distributions.
- The proposed algorithm has detected attacks without any change in CAN protocol. Therefore, it is applicable to any in common vehicles using CAN protocol.

In summary, the work has demonstrated the usability of graph properties in CAN bus environment to detect attacks. The rest of the chapter is organized as follows: section 6.1 presents the attack model, 6.2 presents the proposed methodology. At the end of this section, we present details about the experimental results and limitation, respectively.

6.1 Attack model

These inside and outside attackers can initiate different kind of attacks in the CAN bus. If we generalize those attacks, we can represent it like the following.

- Fabrication Attack: Through an in-vehicle ECU compromised to be a strong attacker, the adversary fabricates and injects messages with forged ID, DLC, and data. The objective of this attack is to override any periodic messages sent by a legitimate safety-critical ECU so that their receiver ECUs get distracted or become inoperable. Example. DoS attack (74), spoofing attack (128), fuzzy attack (46).
- Suspension Attack: To mount a suspension attack, the adversary needs only one weakly compromised ECU, i.e., become a weak attacker. The objective of this attack is to stop/suspend the weakly compromised ECU's message transmissions, thus preventing the delivery/propagation of information it acquired, to other ECUs (129) . Effect of the attack is

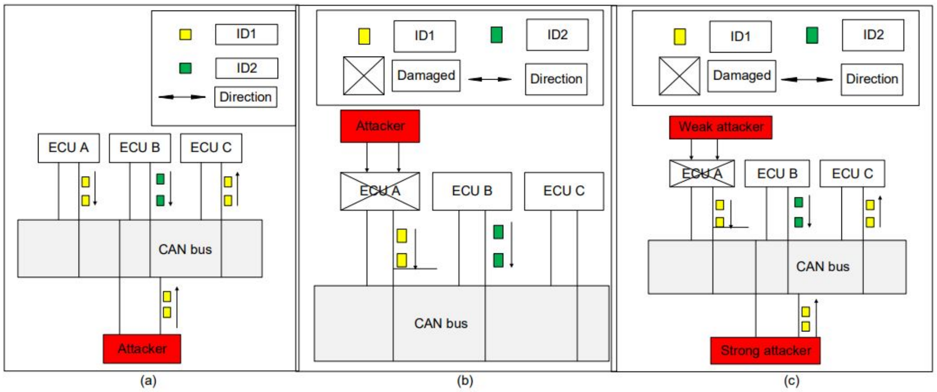


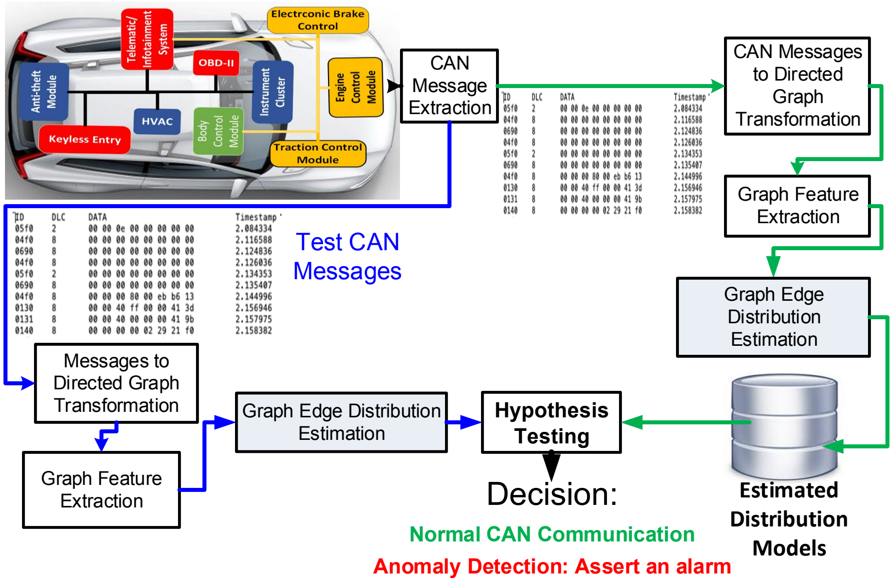
Figure 6.1: CAN bus attack scenario. (a) Fabrication attack, (b) Suspension attack and (c) Masquerade attack

for some ECUs, they must receive certain information from other ECUs to function properly. Therefore, the suspension attack can harm not only the (weakly) compromised ECU itself but also other receiver ECUs. Example. DoS attack (74)

- Masquerade Attack: To mount a masquerade attack (74), the adversary needs to compromise two ECUs, one as a strong attacker and the other as a weak attacker. the adversary monitors and learns which messages are sent at what frequency by its weaker attacker and then the Strong attacker transmits message with the ID of the weak attacker at the same frequency. Example: replay attack or impersonate attack (110) .

6.2 Methodology

An Intrusion Detection System (IDS) to secure the CAN bus communication system is proposed in this section. To detect the anomaly, an strong intrusion detection system is required.In this work, a four step CAN bus anomaly detection system has been introduced. The proposed IDS first builds an hypothesis based on benign CAN bus data. It takes the graph properties as a base to build the hypothesis. When the the benign CAN message hypothesis is achieved, the IDS can take decision about a unknown chunk of CAN messages by a hypothesis testing. Figure 6.2 shows the overview of our graph based intrusion detection system for controller area network.The IDS uses statistical analysis as a base for detecting anomalies and is divided into few of steps. They are (i) transferring CAN bus message to a more meaningful structure (Graph) (ii) extracting graph based features based on the node - edge relation (iii) constructing hypothesis based on the safe population window, (iv) comparing test population window to the base population This section is organized as following.


Telematic/ Infotainment System
Electrconic Brake Control
OBD-II
Anti-theft Module
Instrument Cluster
Engine Control Module
HVAC
Keyless Entry
Body Control Module
Traction Control Module
CAN Message Extraction
CAN Messages to Directed Graph Transformation
Graph Feature Extraction
Graph Edge Distribution Estimation
Test CAN Messages
Messages to Directed Graph Transformation
Graph Feature Extraction
Graph Edge Distribution Estimation
Hypothesis Testing
Estimated Distribution Models
Decision:
Normal CAN Communication
Anomaly Detection: Assert an alarm

```
ID      DLC     DATA                        Timestamp
05f0    2       00 00 0e 00 00 00 00 00     2.084334
04f0    8       00 00 00 00 00 00 00 00     2.116588
0690    8       00 00 00 00 00 00 00 00     2.124836
04f0    8       00 00 00 00 00 00 00 00     2.126036
05f0    2       00 00 00 00 00 00 00 00     2.134353
0690    8       00 00 00 00 00 00 00 00     2.135407
04f0    8       00 00 00 80 00 eb b6 13     2.144996
0130    8       00 00 40 ff 00 00 41 3d     2.156946
0131    8       00 00 40 00 00 00 41 9b     2.157975
0140    8       00 00 00 00 02 29 21 f0     2.158382
```

```
ID      DLC     DATA                        Timestamp
05f0    2       00 00 0e 00 00 00 00 00     2.084334
04f0    8       00 00 00 00 00 00 00 00     2.116588
0690    8       00 00 00 00 00 00 00 00     2.124836
04f0    8       00 00 00 00 00 00 00 00     2.126036
05f0    2       00 00 00 00 00 00 00 00     2.134353
0690    8       00 00 00 00 00 00 00 00     2.135407
04f0    8       00 00 00 80 00 eb b6 13     2.144996
0130    8       00 00 40 ff 00 00 41 3d     2.156946
0131    8       00 00 40 00 00 00 41 9b     2.157975
0140    8       00 00 00 00 02 29 21 f0     2.158382
```

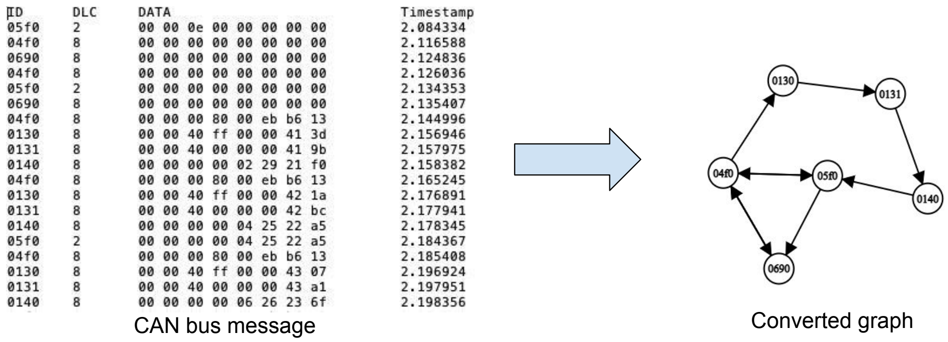
Figure 6.2: Graph based intrusion detection system

6.2.1 Methodology terminologies

Throughout this section few terminologies were used. For better understanding it will be better to be familiar to those terms. they are window and population window. A range of raw CAN bus messages will be called as a single window. A window can be the length of CAN messages. We took 200 messages as the window size. In the results section we will talk about it in detail.And then comes the population window. It is actually a set of Windows. It actually represents a distribution of windows and is used by our methodology to perform hypothesis testing.

6.2.2 Transferring CAN message to graph

In (5), the author proposes an anomaly identification technique based on recurring sequential message IDs. The idea was so simple that in a CAN bus system, if two message IDs come one by one, they are more likely to appear again. He has achieved good success in defending the normal attacks. But this model is vulnerable to intelligent attacks. We considered this as an starting point for our IDS. We have divided the CAN bus messages into a number of windows and then tried to derive relation among all the arbitration IDs for each window. We know that Graphs are a common method to indicate relationship among data. Its purpose is to present data that are too complicated to express using simple text or words. For that reason it has now turned into one of the most popular fields of research. As graph theory can represent complex relationships of data in a very simple manner. After this step of IDs we will have a lot of CAN bus data windows in a meaningful



```
ID      DLC     DATA                            Timestamp
05f0    2       00 00 0e 00 00 00 00 00         2.084334
04f0    8       00 00 00 00 00 00 00 00         2.116588
0690    8       00 00 00 00 00 00 00 00         2.124836
04f0    8       00 00 00 00 00 00 00 00         2.126036
05f0    2       00 00 00 00 00 00 00 00         2.134353
0690    8       00 00 00 00 00 00 00 00         2.135407
04f0    8       00 00 00 80 00 eb b6 13         2.144996
0130    8       00 00 40 ff 00 00 41 3d         2.156946
0131    8       00 00 40 00 00 00 41 9b         2.157975
0140    8       00 00 00 00 02 29 21 f0         2.158382
04f0    8       00 00 00 80 00 eb b6 13         2.165245
0130    8       00 00 40 ff 00 00 42 1a         2.176891
0131    8       00 00 40 00 00 00 42 bc         2.177941
0140    8       00 00 00 00 04 25 22 a5         2.178345
05f0    2       00 00 00 00 04 25 22 a5         2.184367
04f0    8       00 00 00 80 00 eb b6 13         2.185408
0130    8       00 00 40 ff 00 00 43 07         2.196924
0131    8       00 00 40 00 00 00 43 a1         2.197951
0140    8       00 00 00 00 06 26 23 6f         2.198356
```

Figure 6.3: Generated directed graph from CAN messages

structure.

So, first a meaningful structure from raw CAN bus data is extracted. This is the graph building part. Each window is selected from the dataset and graphs were built. The figure 6.3 shows a generated graph from a CAN bus data. The nodes of the graph represent arbitration IDs of the CAN bus message. Any edge between two nodes indicates CAN bus sequential messages. And the direction of the edge indicates the order of the sequence of the messages. For example, if node 0440 has an edge with node 043f and the direction of the edge goes to 043f from 0440, it means data with arbitration ID 0440 and 043f was emitted in the CAN bus one after another with message 0440 was followed by message with arbitration ID 043f. The figure 6.3 contains 14 arbitration IDs as nodes and the edges show us the flow of messages through the CAN bus.

6.2.3 Extracting graph based features based on the node - edge relation

We know that graph comes with different properties, depending on the structure like number of edges, number of nodes, the in-degree, the out-degree. In this step, our proposed method will characterize each of the windows first and then will build the population window. To extract graph properties from a graph constructed using a window of CAN message, our methodology has used the outcomes of stage 1. Table 6.1 shows the extracted graph features from CAN bus message and their significance. The general graph feature like vertex, edge, degree, cycle and root can be extracted from the CAN bus message chunk. The experimental result suggests the edge distribution has a significant feature difference between the benign and malicious CAN bus message. Hence, the IDS considered the edge distribution as a base for the IDS.

6.2.4 Constructing hypothesis based on the safe Population window

In this step of IDS, a hypothesis is built based on the information of the population window. We have used the popular Chi Square statistical test to build the hypothesis. A Chi Square test can be

Table 6.1: Graph properties in terms of CAN messages

Properties	**Significance in Graph**	**Significance in CAN bus**
Vertex	Node of the graph	Arbitration ID
Edge	Link between two nodes	Arbitration ID sequence
Degree	How many Neighbors	How many arbitration IDs are sequential with current ID
Cycle	Loop between the nodes	Loop between the sequential arbitration ID
Root	Start of the graph	First CAN bus message

defined as the following equation.

$$X_c^2 = \sum_{i=0}^{c} (O_i - E_i)^2 / E_i \tag{6.1}$$

Where c is the degree of freedom, X_c^2 is the value to compare against the base hypothesis. O is the observed value and E is the expected value. By the end of this stage our proposed model will provide us a value that will indicate the difference between two distributions. We will be using this value in the next stage to detect anomalous population.

6.2.5 Comparing test population window to the base population window

This step of the IDS consists of two functionalities. The first one is to calculate the chi square value for the test population window and the last one to compare the value with the threshold. By using the following formula we can detect anomalous population.

$$\text{Attack free if } X_c^2 <= threshold \tag{6.2}$$

$$\text{Attack if } X_c^2 > threshold \tag{6.3}$$

The threshold value is selected using the chi square table. The threshold value depends on the degree of freedom and the level of significance value. Level of significance was tuned properly and discussed in the result section.

6.3 Experimental result

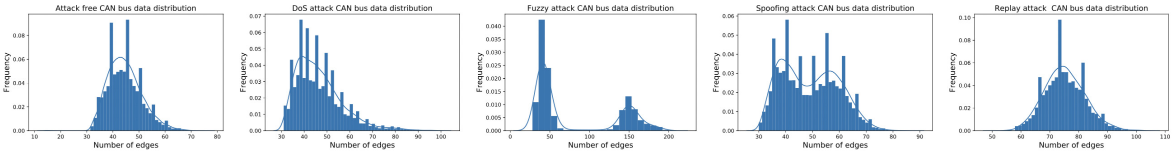
In order to verify the proposed methodology, a real CAN dataset is used to perform analysis on an Intel Xeon(R) 3.8 GHz 8-core processor with 32 GB RAM using our proposed algorithm in

Python language. For the dataset 23k graphs were extracted to do the analysis. This chapter the result section is discussed on several different aspects. First the graph based feature extraction is discussed.

6.3.1 Graph feature extraction

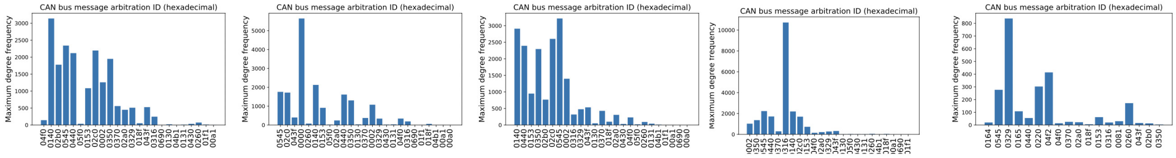
The goal for this experiment is to find features from those graphs that can distinguish the benign CAN data and malicious CAN data. For this experiment a real vehicle data set provided by (46) is chosen. These datasets includes (i) Attack free CAN data, (ii) DoS attack CAN data, (iii) Fuzzy attack CAN data, (iv) Spoofing attack CAN data.

The real vehicular CAN bus data are divided into a few windows first. The size of the window is 200 messages. Then the graph for each of the windows is built and graph properties like edge number, node number, degree etc for each window are derived. The following figures show the detailed characteristics about graphs that are built with attack free CAN bus data set, DoS attack CAN bus data set, Fuzzy attack CAN bus data set and spoofing attack CAN bus data set.



(a) Attack free (b) DoS Attack (c) Fuzzy Attack (d) Spoofing Attack (e) Replay Attack

Figure 6.4: Edge distribution



(a) Attack free (b) DoS attack (c) Fuzzy attack (d) Spoofing attack (e) Replay attack

Figure 6.5: Maximum degree distribution

Figure 6.4 and 6.5 shows information about the details of edge and degree of the graph for different kinds of attacks. In figure 6.4 (a) and figure 6.5 (a) represents the data distributions of number of edges of each graph and frequency of maximum degree of a collection of graph data for attack free normal CAN message. Figure 6.4 (a) clearly shows the edge distribution maintains a normal distribution. Figure 6.4 (b), 6.4 (c), 6.4 (d) and 6.4 (e) represent the distribution of edges for DoS, fuzzy, spoofing and replay (impersonate) attack sequentially. Like the attack free edge distribution, the DoS, fuzzy, spoofing attack edge distribution do not have the normal form. For

DoS attack it is positively skewed, for spoofing and fuzzy attack it is bimodal. If we consider figure 6.4 (e), the edge distribution for replay (impersonate attack) is nearly similar to attack free distribution. That's why this attack is known as the most intelligent attack and can perform similar behavior like the attack free state of the CAN bus system.

Figure 6.5 shows the maximum degree frequency of a graph collection. If we look closely at 6.5 (a), it is clear that in an attack free state, a CAN message with ID 0140 has formed more pairs with other CAN messages in every certain period of time. 6.5 (b) represents the situation of maximum degree for DoS attack. And the figure clearly proves the definition of DoS attack (CAN message with highest priority appears most of the time to block the channel , so CAN message with arbitration ID 0000 appeared most in the collection of graph). Figure 6.5 (c), 6.5 (d) and 6.5 (e) indicates the maximum degree distribution for fuzzy, spoofing and replay attack correspondingly.

The data distributions of different attacks are not only different from the attack free CAN bus distribution but also different from each other attacks. In terms of statistical terms we can summarize the overall situation of the data distributions by the following table.

Table 6.2: Statistical analysis of different kind of attacks

Analysis	**Attack free**	**DoS attack**	**Spoofing attack**	**Fuzzy attack**	**Replay attack**
Mean (Edge)	44.59	46.60	49.12	79.82	75.17
Median (Edge)	44.0	45.0	49.0	46.0	75.0
Skewness (edge)	Moderate	High	Similar	Moderate	Similar
Max degree ID (%)	0140 (16.9)	0000 (30.8)	0316 (46.4)	0545 (16.8)	0164 (31)

We have only considered the central tendency or mean of the distribution and the asymmetry of probabilistic distribution. If we consider the mean of edges in a graph collection, attack free graph distribution has a mean of 44.59, on the other hand DoS, fuzzy, spoofing and replay attack has mean of 46.60, 49.12, 79.82 and 75.17 correspondingly. The table also shows the skewness of all the edge distributions for attack free and all kinds of attacked dataset. Spoofing and replay attack edge distribution seems symmetric. on the other hand, attack free and fuzzy attack edge distribution is moderately positive skewed. And finally DoS attack edge distribution is highly skewed. If we look at the maximum degree of CAN arbitration ID in the graph distribution, we find variation of CAN message appeared as maximum degree in each kind of attack.

Finally, Those exploratory analysis proves that the conversion between raw CAN data to graph gives us a clear indication. And that is we can fetch some extraordinary information from the converted graph. Finally, the graph properties can be used to distinguish different situations (attack free, DoS, Fuzzy, Spoofing, replay etc) of the CAN bus system.

6.3.2 Graph size selection

The number of CAN messages considered to build a graph is an important parameter. Because it will impact the attack detection time for the proposed approach. So, it is important to tune the graph size in order to keep the approach lightweight and at the same time also efficient. As each graph will represent a chunk of CAN messages, so it is important that each chunk will contain information for most of the unique arbitration IDs. So, the percentage of unique arbitration IDs in each graph is calculated and is shown in figure x. The figure shows that if each graph is built considering 200 CAN bus messages, 90% of all arbitration IDs appear in each graph on average. If we consider 100 CAN bus messages per graph only 84% of all arbitration IDs appear in each graph on average. Increasing graph size to 500 CAN messages does not increase the percentage of unique CAN arbitration IDs much. It increases the computation of 300 CAN messages while it only increases 1% extra information about all the arbitration IDs.

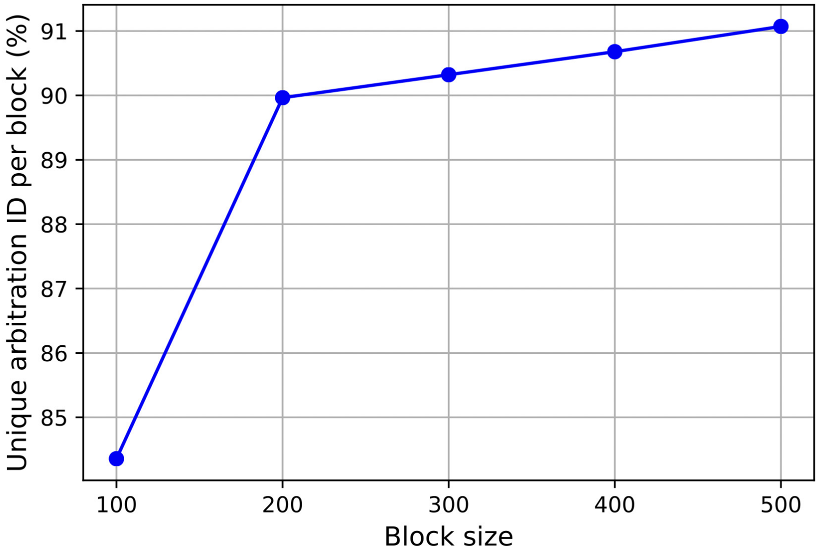


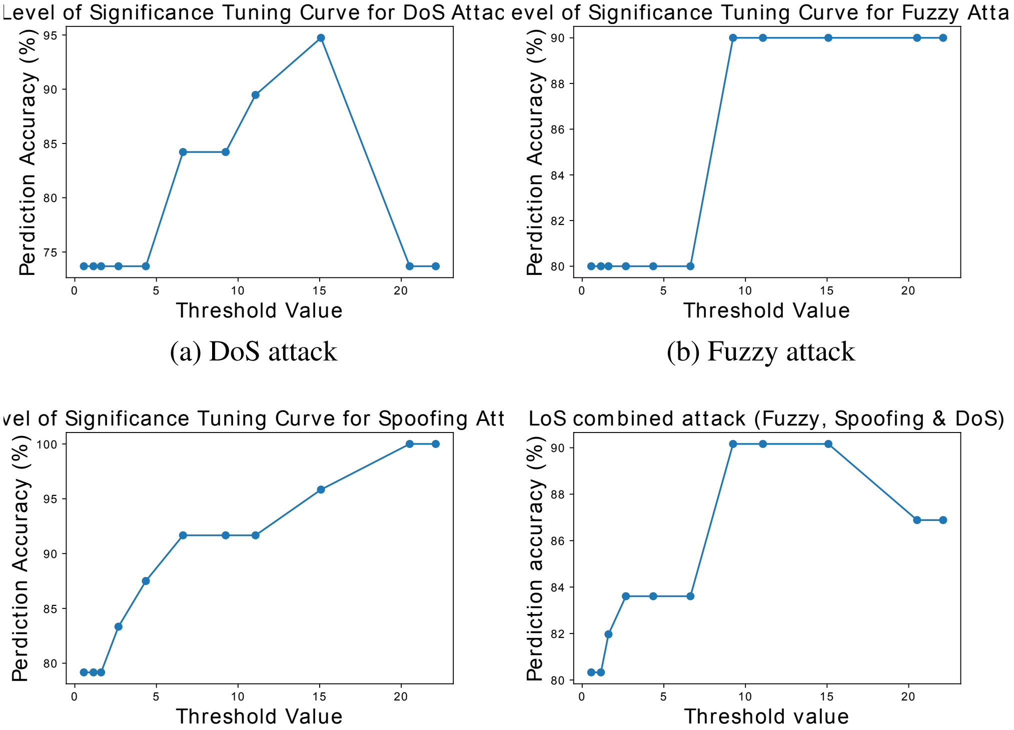
Figure 6.6: Selection of the block size.

6.3.3 LoS parameter tuning

The significance level is the probability of rejecting a hypothesis. It is important because the threshold value changes depending on the level of significance. Our test result showed the best level of significance we should choose for picking up the threshold value for comparing attack free or attacked distribution. The threshold for chi square test depends on the degree of freedom and the level of significance directly. (130) The equation is as follows,

$$DoF = (i - 1) * (j - 1) \tag{6.4}$$

Where i is the number of rows and j is the number of columns considered for the distribution. In our case we have two rows. One for the reference distribution and the other for test distribution. And column number is 6 as we divided each of the two distributions into 6 portions and compared them. We divided them into 6 portions was totally intentional.



(a) DoS attack

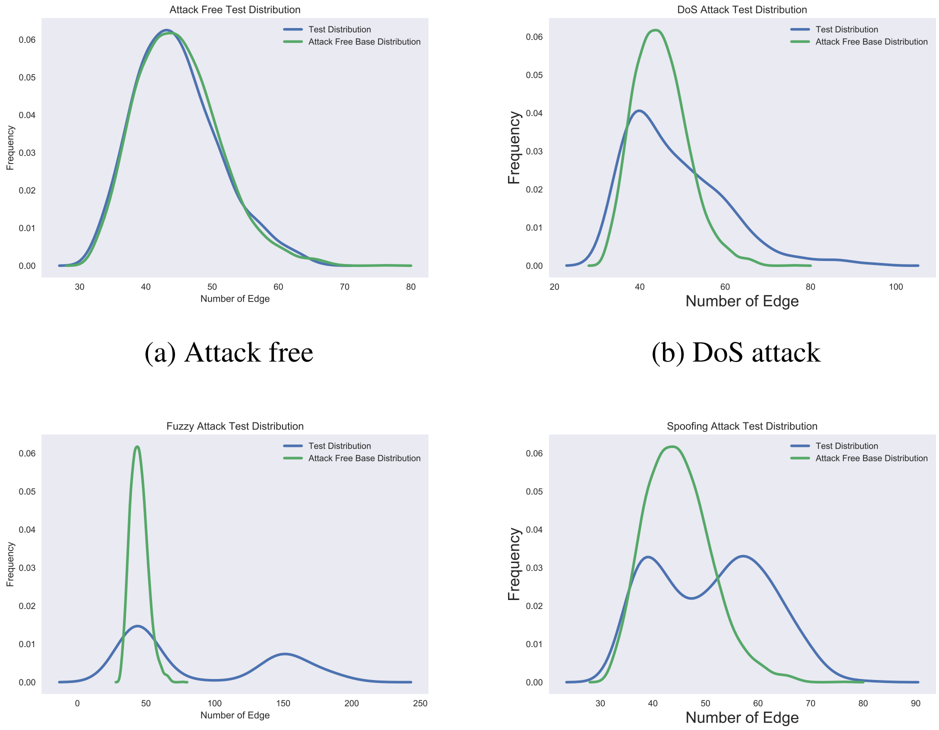
(b) Fuzzy attack

(c) Spoofing attack

(d) Mix attack

Figure 6.7: Level of significance tuning

From our exploratory analysis we know the attack free edge distribution is normally distributed. And that is our reference distribution in chi square. Besides, in statistics the empirical rule states that in a normal distribution 99.7 data points should be within $(mean \pm 3 * (\text{standard deviation}))$ of the distribution. So, we divided the reference and test distribution into 6 portions ranging from $(mean - 3 * (\text{standard deviation}))$ to $(mean + 3 * (\text{standard deviation}))$ in a step of single standard deviation initially. But in (131), researchers proved that for detecting outliers, considering median in place of mean gives better results. That means, Both median and mean signifies central tendency of a distribution but median does not affected by anomalous data. For that reason, we divided the reference and test distribution ranging from $(median - 3 * (\text{median standard deviation}))$ to $(median + 3 * (\text{median standard deviation}))$ ie 6 portions with a step of single standard deviation. So, the column number is 6 and row is 2. From the equation we get our degree of freedom is 5. From the chi square table (132) we can choose the level of significance given the threshold and the degree of freedom. Our exploratory analysis showed the distribution of different attacks has different patterns. Hence, we propose different levels for DoS, Fuzzy and Spoofing attack. Figure 6.7 suggests a significance level is 0.01 (threshold 15.086) give us the best accuracy for DoS attack. In terms of Spoofing attack and fuzzy, we propose a significance level of 0.001 (threshold 20.515) and 0.1 (threshold 9.24).



(a) Attack free

(b) DoS attack

(c) Fuzzy attack

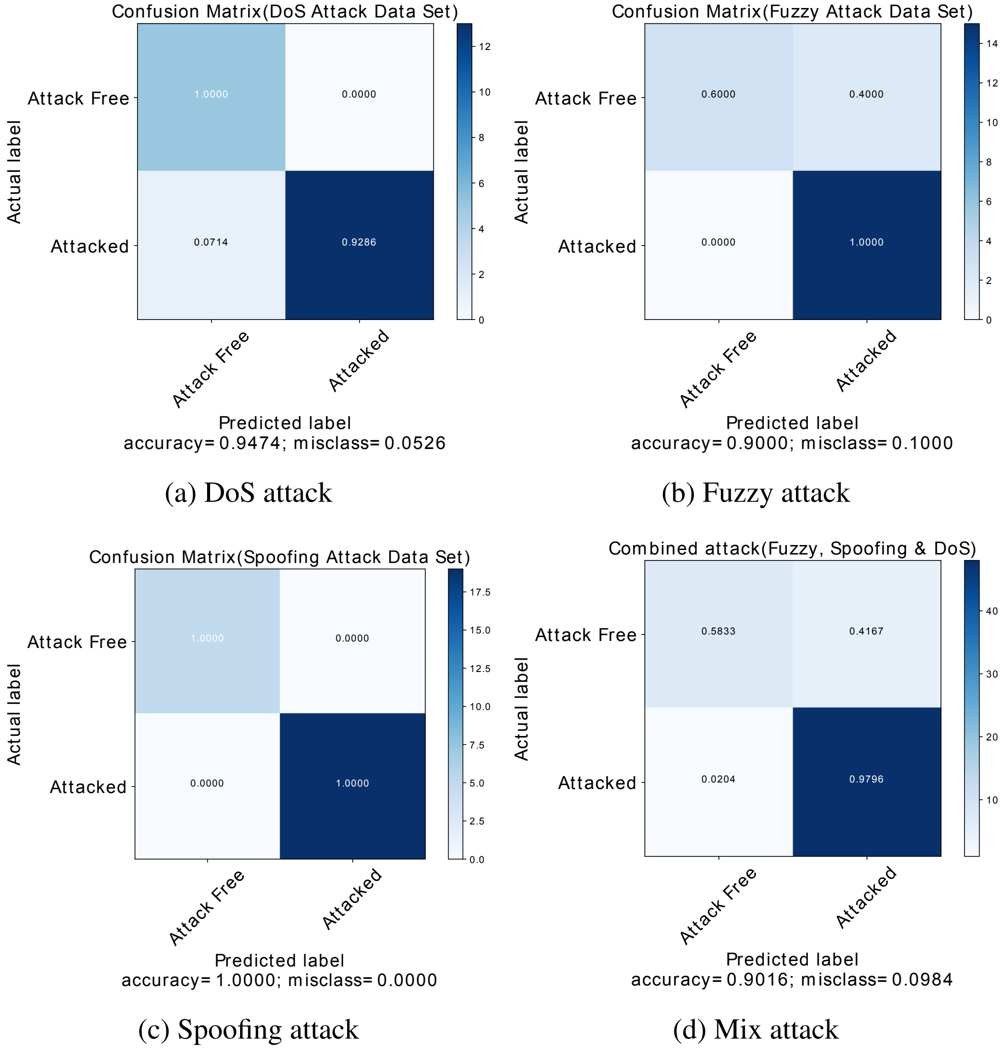
(d) Spoofing attack

Figure 6.8: Chi Square test (base and test distribution plot)

6.3.4 Detection of DoS, fuzzy & spoofing attack

For detecting the attack using chi square, first a base hypothesis is made by explorating the attack free CAN data. We call it a base distribution. After any distribution can be compared with the base hypothesis and differences can be found easily. Figure 6.8 (a) has a visual representation of our Chi Square test for attack free distribution. The distribution colored as green represents the safe distribution and on the other hand, the blue distribution represents the distribution we want to test. We take one of the distributions built from the attack free CAN data for test. And figure 6.8 (a) shows they are similar. After that test distributions built from DoS, fuzzy and spoofing attack was chosen for comparison with the safe distribution. Figure 6.8 (a) , 6.8 (b), 6.8 (c) shows the test distribution is significantly different from the safe distribution. For testing DoS, Fuzzy, Spoofing attack and attack free data sets we have only used edge distribution of graphs and found significant results.

Figure 6.9 shows that our methodology has achieved a very good accuracy in detecting all three attacks described in the section when they are individually tested. The misclassification rate is very low. It is 0.0526% and 0.1% for DoS and Fuzzy attack. For spoofing attacks we were able to classify all of the test cases successfully.



(a) DoS attack

(b) Fuzzy attack

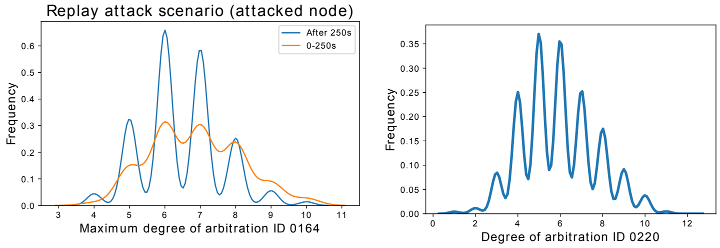
(c) Spoofing attack

(d) Mix attack

Figure 6.9: Confusion matrix

6.3.5 Detection of replay attack

To detect the smart attack, the central tendency theorem does not provide a good amount of accuracy. This is because the benign graph edge distribution and reply attack edge distribution is nearly similar in shape. We were able to achieve 66% accuracy. In terms of replay (smart) attack, there is no change in edge distribution between malicious and benign CAN bus data set. For that the maximum degree was considered of a particular arbitration ID in a distribution. Our result showed that the attacked arbitration ID (Example in figure 6.10 (a) 0164 arbitration ID) has different distribution when reply attack is made (after 250 seconds in the dataset) and other safe arbitration ID maintains an identical distribution. (Example 0220 arbitration ID in figure 6.10 (b)).



(a) Replay attack arbitration ID (b) Attack free arbitration ID

Figure 6.10: Maximum degree analysis for replay attack

6.3.6 Detection of a mix attack

To measure the efficiency against a mix of DoS, fuzzy and spoofing attack the data sets were merged together and the efficiency were measured with respect to different levels of significance. Figure 6.9 (d) shows a level of significance of 0.1 gives us a maximum of 90.16% accuracy. At this level of significance a minimum of 9.84% misclassification rate has been achieved also.

6.3.7 Comparison with the state-of-the-art

The efficiency of the proposed approach to detect an attack in CAN's message was evaluated by comparing it to one of the state-of-the-art IDS (5). To the knowledge this is the only methodology that uses CAN bus message sequence to identify CAN attacks. The proposed method is also constructing a graph to find a pattern among CAN bus message arbitration IDs and using it to detect an attack. Therefore we implemented their approach in the same experimental environment using another real vehicular CAN message dataset (46) to estimate the effectiveness of our approach. When we considered spoofing attack, the proposed methodology has 13.73% better accuracy compared to the existing method, as shown in Table 6.3. When we consider a fuzzy attack, the proposed method exhibits comparable accuracy, however, for DoS attack, the proposed methodology shows 5.26% lower accuracy to the existing ID sequence method. One of the most attractive features of the proposed method is it can detect replay attacks with 66% accuracy, while the existing method could not detect any replay attacks. Table 6.3 shows the misclassification rate of the proposed method compared with the state-of-the-art (5).

Table 6.3: Comparison with the state-of-the-art (5)

Method	**DoS detection rate (%)**	**Spoofing detection rate (%)**	**Fuzzy detection rate (%)**	**Replay detection (Yes/No)**
Graph based IDS	94.74	100	100	Yes
State of the art (5)	100	99.04	86.23	No

6.4 Discussion

As the involvement of modern technologies in the vehicular industry is increasing the number of cyber threats, we very much need a robust security mechanism to detect them. In this framework, we analyzed the characteristics of all kinds of CAN monitoring-based attacks and proposed a four-stage IDS with the help of graph theory, statistical analysis, and the chi-square method. To the best of our knowledge, this is the first graph based IDS for CAN bus communication. Our experimental results show that we have a very low misclassification rate in detecting attacks or attack free data. Apart from that this work has some limitations. First is that it takes around 200 CAN messages to build graph and then 1500 graph distribution to take a decision. It means 3,00,000 CAN message to take a decision. In the future, we will like to see how minimum number of CAN message is required to detect attack using graph based approach. Moreover, we would like to consider other graph properties such as distribution of nodes, cycles, weighted edges etc. In addition, we will apply different machine learning algorithms in place of the chi-square test to identify anomalies.

CHAPTER 7

Behavioral Fingerprinting via Graph Theory - Machine Learning Approach

To increase CAN bus security researchers have proposed several solutions by providing message authentication (133) (74) (73). But, these solutions are not practical on CAN bus protocol since CAN bus has limited data byte (8 bytes). Additionally, the implementation of message authentication will add overhead and will limit the existing bandwidth (500 kbps). So, the techniques like designing IDS (intrusion detection system) is becoming more popular as they do not limit bandwidth (87) and do not modify existing CAN bus protocol.

In this chapter, an IDS is proposed to detect DoS, fuzzy and spoofing CAN bus attacks by using machine learning. Popular machine learning algorithms SVM (support vector machines) and KNN (k-nearest neighbors) are used for intrusion detection. SVM and KNN have been used recently by (2) to increase CAN security. Unlike (2), the proposed IDS uses seven features (six novel features and a single feature from the state-of-the-art (3)) with high feature differences among benign and malicious CAN messages, which gives better classification accuracy. The experimental results show that the selected seven features represent the accurate behavior of CAN benign and malicious messages. The proposed work is inspired by the work done in (3). The author converted the CAN messages into graphs and used a single graph property i.e edges to detect CAN bus attack by using the statistical chi-square method. Unlike (3), new seven graph-based CAN bus features are explored that are used to detect CAN intrusions. Moreover, the proposed IDS can detect attacks using a single graph, whereas (3) needs distribution of graphs for CAN bus attack detection. The followings are the contribution of this chapter:

- To the best of our knowledge, this is the first machine learning-based CAN bus IDS that uses graph-based features.
- The IDS takes advantage of a total of seven graph-based properties that represent the actual behavior of the CAN bus.

- The experimental results show that the classification based on graph-based features are performing better than classification based on traditional CAN bus message features.

- The proposed IDS can detect three types of CAN attack and is applicable to in-vehicle networks.

7.1 Attack model

In this section, we first present the adversary model. Afterward, we discuss three attack scenarios that can seriously ruin in-vehicle functions: spoofing attack, fuzzy attack, and Denial-of-Service (DoS) attack.

7.1.1 Adversary characteristics

In this chapter, we assume an adversary can physically or remotely compromise in-vehicle ECUs via several attack surfaces such as OBD-II, CD players, USB, Bluetooth, and cellular (25). We consider an attacker who wants to control or disable or paralyze in-vehicle ECUs' functionality. An attacker can accomplish this by either injecting arbitrary messages repeatedly into the CAN bus or by injecting unauthenticated messages with a spoofed ID into the in-vehicle network. In this chapter, we discuss three kinds of attackers who can inject malicious messages in the in-vehicle network through the CAN bus. We assume an attacker can inject malicious messages in order to control and breakdowns vehicle functionality. Accordingly, we consider three types of message injection attacks which are spoofing, fuzzy and DoS attacks.

7.1.2 Attack Scenarios

Based on the above described adversary model, we consider the normal CAN-bus data (attack-free) in addition to three kinds of attack scenarios which are spoofing, fuzzy and DoS as shown in Figure 4.

Spoofing attack: This attack happens when an attacker injects a single message of randomly spoofed CAN identifier with arbitrary data. Subsequently, it causes unintended vehicle behaviors since all ECUs will receive that message. To exploit the spoofing attack, an attacker can inject arbitrary data into one message of the in-vehicle messages and chose the target identifier of that message to create unexpected behaviors for the vehicle. Such behaviors include turning the signal lamps light irregularly, flickering the instrument board in incalculable ways, disabling the braking system and shaking the steering wheel colossally.

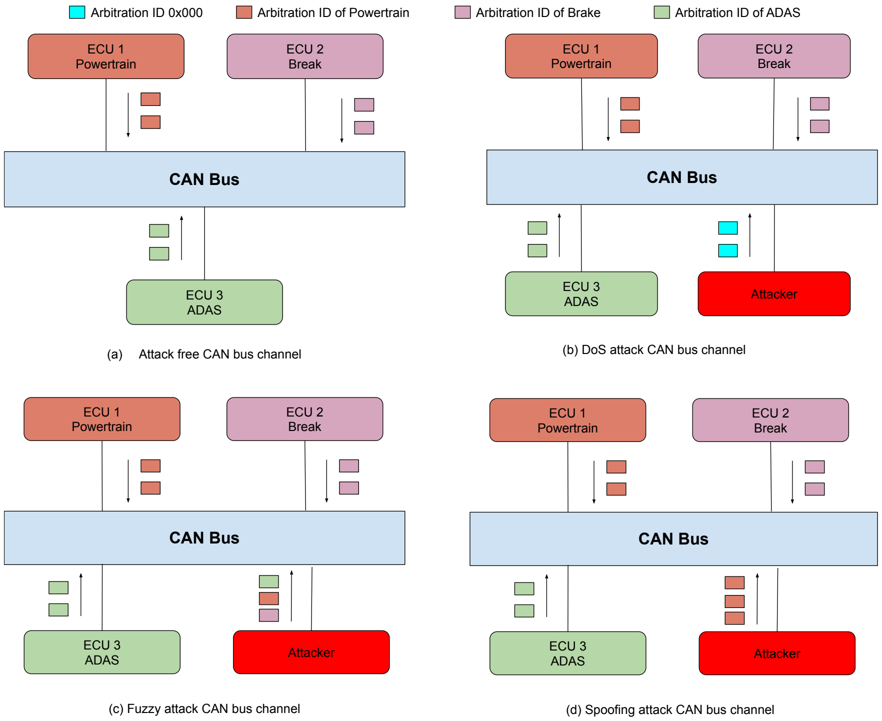

Arbitration ID 0x000
Arbitration ID of Powertrain
Arbitration ID of Brake
Arbitration ID of ADAS
ECU 1 Powertrain
ECU 2 Break
CAN Bus
ECU 3 ADAS
Attacker
(a) Attack free CAN bus channel
(b) DoS attack CAN bus channel
(c) Fuzzy attack CAN bus channel
(d) Spoofing attack CAN bus channel

Figure 7.1: Different CAN bus channel scenarios

Fuzzy attack: This attack occurs when an attacker injects multiple messages with arbitrary data of randomly multiple spoofed CAN identifiers, unlike the spoofing attack which occurs by injecting only a single message of randomly a single spoofed CAN identifier. As a result, all ECUs will receive various messages which cause unintended vehicle behaviors like gearshift changes automatically, disabling the braking system, instrument panel blinks in incalculable manners and the steering wheel shakes gigantically.

DoS attack: This attack happens when an attacker injects high priority of CAN messages such as the 0x000 CAN ID packet in a short cycle on the CAN bus. To exploit the spoofing attack, an attacker can easily occupy the bus by injecting the highest priority identifier of CAN messages such as 0x000 in a short cycle on the CAN bus. Subsequently, it yields latencies of other messages and causes threats in regards to availability with no reaction to the driver's commands since all ECUs share a single bus. Unlike the spoofing and fuzzy attacks, the DoS attack delays the normal messages through the occupancy of the CAN bus rather than cripple the functions of a vehicle.

7.2 Methodology

This section presents the proposed IDS in detail. Sub-section 3.1 starts with the overview of the IDS. It is followed by a CAN bus message to graph conversion. Sub-section 3.3 represents the

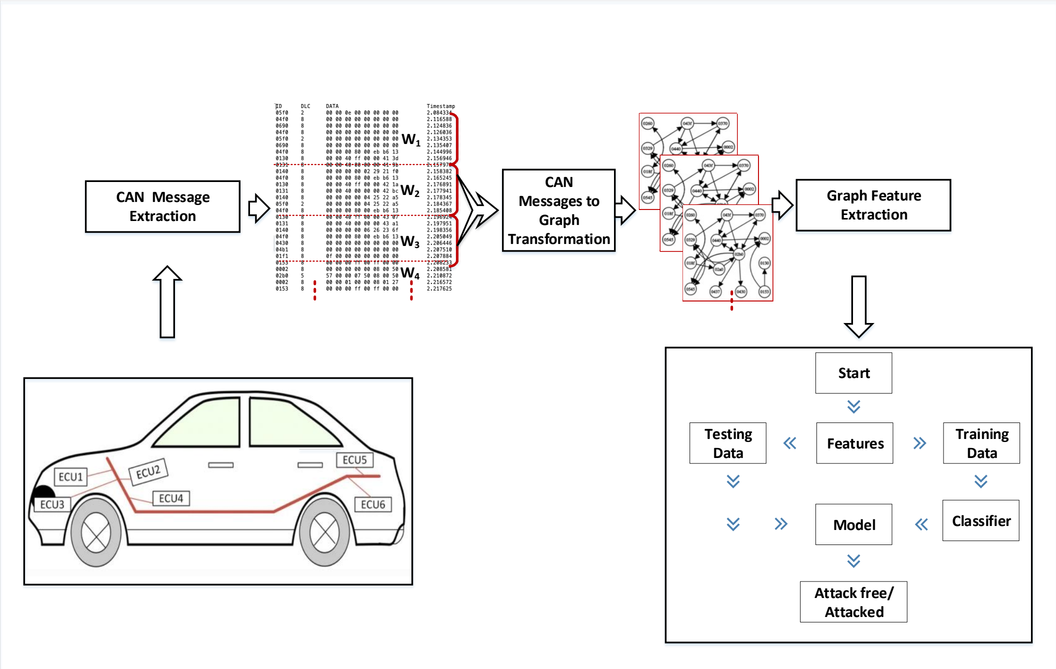


Figure 7.2: Overview of the proposed IDS

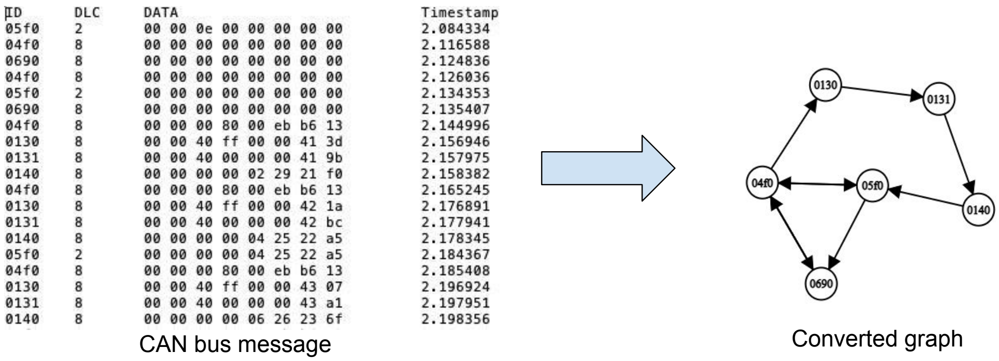
extraction of graph properties from CAN message based graphs. And finally we conclude the section with classification of CAN bus graphs section.

7.2.1 Overview of the IDS

As shown in figure 5, the proposed IDS has three main sub-components of conversion of CAN bus messages to graph structure, extraction of graph properties from CAN message based converted graphs and classification of CAN bus graphs based on the graph features. First the CAN bus messages are converted into graph structures. In the feature extraction phase 8 features have been extracted based on the properties of the graph. Based on the feature differences between benign CAN bus graphs and malicious CAN bus graphs, 7 features were selected for classification phase. In the classification , support vector machine (SVM)(134) and K-nearest neighbor (KNN) (135) is used on the selected features based on attack free CAN bus message and attacked CAN bus message. Experimental result shows classification with SVM and KNN based on the selected features exhibits good accuracy in attack in CAN bus.

7.2.2 Conversion of CAN bus message to graph

To extract the normal behavior of CAN bus system, author in (3), considers a window of CAN bus messages and converts it to a graph. Our proposed IDS takes these concepts and converts a chunk



```
ID      DLC     DATA                            Timestamp
05f0    2       00 00 0e 00 00 00 00 00         2.084334
04f0    8       00 00 00 00 00 00 00 00         2.116588
0690    8       00 00 00 00 00 00 00 00         2.124836
04f0    8       00 00 00 00 00 00 00 00         2.126036
05f0    2       00 00 00 00 00 00 00 00         2.134353
0690    8       00 00 00 00 00 00 00 00         2.135407
04f0    8       00 00 00 80 00 eb b6 13         2.144996
0130    8       00 00 40 ff 00 00 41 3d         2.156946
0131    8       00 00 40 00 00 00 41 9b         2.157975
0140    8       00 00 00 00 02 29 21 f0         2.158382
04f0    8       00 00 00 80 00 eb b6 13         2.165245
0130    8       00 00 40 ff 00 00 42 1a         2.176891
0131    8       00 00 40 00 00 00 42 bc         2.177941
0140    8       00 00 00 00 04 25 22 a5         2.178345
05f0    2       00 00 00 00 04 25 22 a5         2.184367
04f0    8       00 00 00 80 00 eb b6 13         2.185408
0130    8       00 00 40 ff 00 00 43 07         2.196924
0131    8       00 00 40 00 00 00 43 a1         2.197951
0140    8       00 00 00 00 06 26 23 6f         2.198356
```

Figure 7.3: CAN bus message to graph conversion

of CAN messages into a meaningful graph structure. To build a graph, the CAN arbitration ID of every CAN bus message is considered as a node and an edge is put between two graph nodes (arbitration IDs) if two CAN messages come sequentially on after another. Figure 6 shows how a chunk of CAN messages CAN be converted to a graph. As graphs can find meaningful hidden structure of data, so graphs converted from CAN bus data, represent the meaningful behavior of CAN bus.

7.2.3 Graph properties as features

In graph theory, the properties like number of nodes, number of edges etc. are totally dependent on graph structure. To distinguish between benign CAN bus data with malicious (DoS, fuzzy and spooning attack) data the initial plan is to choose the properties that show the characteristics of the graph. Hence, the proposed IDS extracts graph properties like number of nodes, number of edges (3), radius (136), diameter (137), density (138), reciprocity (139), average clustering coefficient (140) and assortativity coefficient (141). Figure 7.4 shows the box plot for each of these graph properties mentioned above in different CAN bus attack free and attack scenarios. The plots clearly show that number of nodes is not distinguishable between benign and malicious CAN bus scenarios. Hence the proposed IDS excludes the number of nodes from classification feature list and selects the other 7 differential features.

7.2.4 Classification of attack detection

The classification step is dependent on the feature extraction & selection process. It takes the selected 8 features as an input and classifies benign and malignant CAN bus data. Two popular machine learning algorithms i.e support vector machines (SVM) and k-nearest neighbor (KNN) are applied to classify the performance.

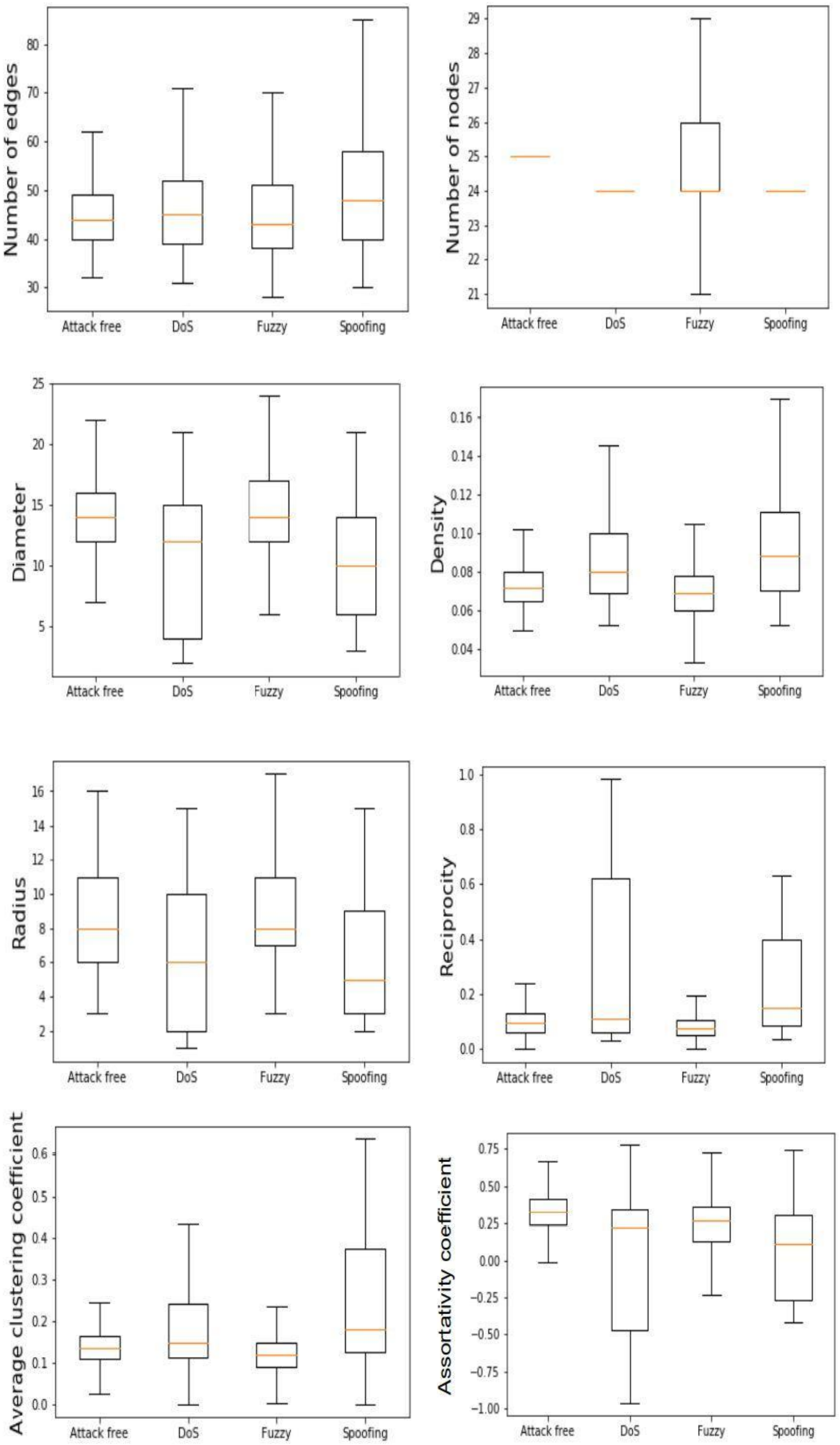


Figure 7.4: Graph based CAN features

- Support Vector Machine (SVM)(134): SVM is a supervised learning algorithm used for classification and regression problems. It actually tries to find a suitable hyperplane in a finite dimension of data that clearly classifies the training data points. In the testing phase, the test instances are compared with the hyperplane to predict the appropriate category of the instance. The algorithm is used in many applications like classification of images, satellite data, categorization of text & hypertext data, handwritten recognition etc. successfully over the years (142).
- K-Nearest Neighbor (KNN) (135): KNN is a non-parametric machine learning algorithm that can be used for both classification and regression. It does not make any assumptions on the underlying data distribution. In the training phase it keeps the similar data near to each other and it is utilized during the testing phase. First , the distances between training data and a test instance are measured; And finally using majority voting among the k-nearest training instances, the class of the test instance is predicted.

Our assumptions and selected features exhibit excellent results in classifying attack free and attacked CAN bus messages.

7.3 Experimental result

7.3.1 Description of the dataset

To verify the effectiveness of the proposed IDS, an experiment is designed and performed on real vehicular dataset (87). The dataset contains three kinds of CAN bus attack data along with benign CAN data. That is DoS attack, fuzzy attack and spoofing attack. Table 8.1 shows the summary of the dataset. To prepare the dataset for our proposed IDS, the CAN messages are converted to temporal graphs. Along the lines of the authors in (3), each graph is built using 200 CAN messages.Using the dataset, we were able to extract 5,558 DoS attack graphs, 2,802 fuzzy attack graphs and 11,263 spoofing attack graphs.

7.3.2 Validation metrics

Each datafile of the selected dataset was fed into the classifier and was treated as a binary classification problem. The performance of the proposed IDS was validated based on precision, recall, F-1 score, accuracy and area under the receiver operating characteristic curve. Out of them precision and recall is considered to measure the quality and quantity respectively. The F-1 score is used to find how precise the classifier is. In the experiment accuracy means the percentage of

Table 7.1: CAN traffic dataset details

Dataset	No of CAN messages	No of Graphs	No of attack-free graphs	No of attack graphs
DoS attack	3,631,600	18,158	5,558	12,600
Fuzzy attack	3,053,400	15,267	2,802	12,465
Spoofing attack (RMP)	4,566,200	22,831	11,263	11,568

correctly classified graphs. The area under the receiver operating characteristic curve is considered to measure the ability of the classifier to classify benign and malignant graphs. The equations for this performance metrics can be found here (143) .

7.3.3 Simulation result

The proposed IDS was designed and implemented using python programming language on a 1.8 GHz windows 10 computer system with 16 GB RAM. The selected dataset was fed into two types of machine learning classifier i.e (i) SVM & (ii) KNN and the performance was measured based on the validation metrics discussed in subsection 4.2. First, 60% of the overall data is considered as the training dataset. The remaining 40% is divided into two equal halves as a validation dataset and test dataset. The reason for using a validation set is to fine tune the classifier hyper-parameters on unknown data while the classifier is fit using training data. The total separate test dataset helps to eliminate the overfitting tendency of a model. While feeding the dataset files into SVM classifier, we achieved accuracy of 99.90%, 99.93% & 96.43% for DoS, fuzzy and spoofing attack respectively. On the other hand, KNN provided accuracy of 99.86%, 99.79% & 96.55%. In order to check the robustness of the IDS, a mix attack by combining DoS, fuzzy & spoofing attack dataset is also performed. While defending the mix attack the SVM classifier achieved 97.92% accuracy while the KNN has achieved 97.99% accuracy. Table 8.3 shows the details about all the validation metrics discussed in subsection 4.2 while defending DoS, fuzzy, spoofing & combined attack.

7.3.4 Comparison with the state-of-the-art

Finally, the proposed IDS is compared with one of the state of the art works (2). The particular reason for choosing (2) for comparison is because it uses CAN bus data message frames as features to detect CAN intrusion using SVM and KNN classifiers. Therefore the effectiveness of the proposed IDS is compared with the state of the art by considering accuracy on real vehicular CAN bus data . While tested on 20% data, (2) achieved accuracy of 97.40% & 96.50% while using SVM

Table 7.2: The results of the fuzzy and the DoS attacks

Attack	Classifier	Accuracy	Precision	Recall	F-1	AUC-ROC score
DoS attack	SVM	99.90	0.9994	0.9969	0.9981	0.9969
	KNN	99.86	0.9993	0.9955	0.9977	0.9955
Fuzzy attack	SVM	99.43	0.9996	0.9952	0.9973	0.9952
	KNN	99.79	0.9989	0.9865	0.9926	0.9865
Spoofing attack	SVM	96.43	0.9761	0.9363	0.9537	0.9361
	KNN	96.55	0.9767	0.9384	0.9554	0.9385
Combined attack	SVM	97.92	0.9861	0.9609	0.9726	0.9608
	KNN	97.99	0.9895	0.9623	0.9737	0.9623

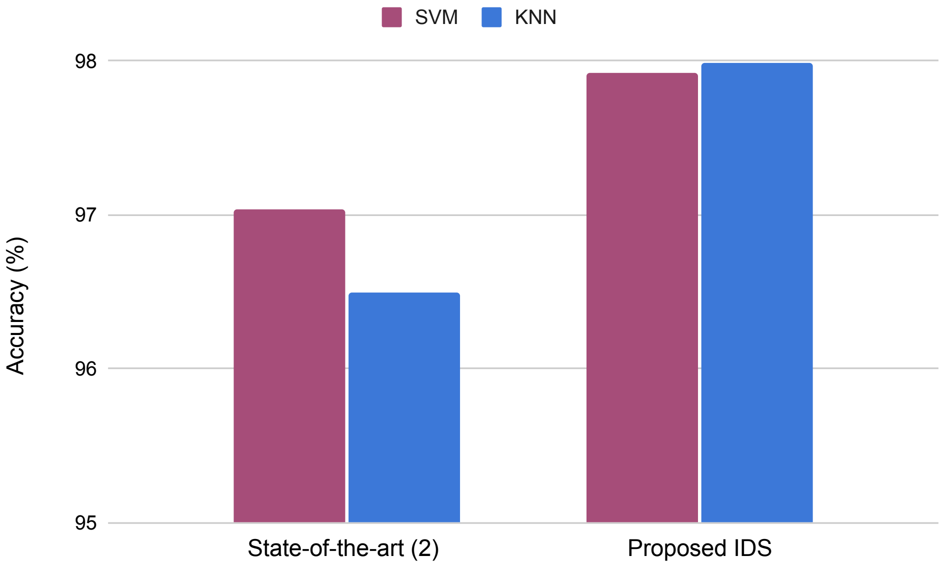


Figure 7.5: Comparison with the state-of-the-art (2) in combine attack scenario

& KNN. On the other hand the SVM & KNN based on our proposed graph based features achieved accuracy of 97.92% & 97.99% respectively. Figure 7.5 shows the comparison between the state of the art (2) and proposed IDS. The figure clearly demonstrates that the proposed IDS has achieved better accuracy than the state of the art for both SVM & KNN.

7.4 Discussion

In the modern transportation system, more and more connectivity is added in vehicles with the outside world. More connectivity adds more threat surfaces in vehicles which poses serious threat to the safety of passengers and security of vehicles. In order to make the in-vehicle network secure a strong intrusion detection system is required. In this chapter, a CAN intrusion detection system is proposed that uses graph based features to detect CAN attack. This novel and pragmatic approach uses graph based features to classify authentic and malicious CAN messages for in-vehicle communication. The experimental results showed that using graph-based features, an accuracy of 97.92% & 97.99% was achieved using SVM & KNN algorithms respectively. In future, we would like to consider other graph properties to detect intrusion for in-vehicle communication. In addition, we will apply different machine learning algorithms in place of the SVM & KNN to see the robustness of the selected features.

CHAPTER 8

Behavioral Fingerprinting via Graph Theory - Ensemble Machine Learning Approach

As mentioned above, most of the current vehicles have some degree of autonomous features that interact with the environment through Bluetooth, WiFi and Cellular, etc, which can be exposed to the remote attacks (144). For example, the authors in (25) demonstrated that an attacker can compromise the in-vehicle ECUs remotely. Once the ECU is compromised, an attacker can take control or paralyze the vehicle entirely by injecting malicious messages in the in-vehicle networks (113). As another demonstration, Miller et al. (33) have hacked a Jeep Cherokee remotely and misguided it while running on a highway remotely. As a consequence, 1.4 million vehicles were recalled by the car manufacturer. In 2016, Keen Security Lab of Tencent was able to perform message injection attack remotely while the vehicle is on driving mode as well as parking mode (17). Due to these remote attacks, automotive security has become one of the most critical issues that may caused by the vulnerabilities or bug of different sub-systems in a vehicle. The automotive companies are also concerned of these cyber threats and most of them have bug-bounty programs (1).

This chapter presents a novel lightweight IDS framework to reliably detect anomalies in the CAN bus traffic in order to secure the automotive system. Work in (3) has demonstrated that CAN traffic can be modeled using a finite state machine and proposed a graph-based IDS for CAN protocol. However, this state-of-the-art work (3) requires around 300,000 CAN message for attack detection which is one of the limitations of this approach. Unlike the prior work, our approach aims to improve the response time of the graph-based IDS proposed in (3). The proposed lightweight graph-based IDS requires a single chunk (consisting of only 200 CAN message) for attack detection by calculating the graph neighborhood-based similarity scores. To increase the reliability of the IDS, an ensemble based approach is applied to the similarity scores. The performance of the proposed IDS is evaluated on a real vehicular CAN dataset consisting of three different types of CAN message injection attacks(spoofing, fuzzy & DoS attacks). The experimental results indicate that the proposed method improves the state-of-the-art and is capable of reliably detect attacks.

Overall, this chapter makes the following contributions:

- Proposes a lightweight IDS based on graph neighborhood similarity analysis to improve the performance, reliability and accuracy of the CAN protocol.
- Develops a novel algorithm to detect anomalies on transmitted CAN bus by comparing any single graph of CAN messages with multiple graphs of attack-free CAN messages. This algorithm is beneficial to detect message injection attacks with low detection time.
- Evaluates the performance of the proposed IDS against three kinds of message injection attacks(spoofing, fuzzy & DoS attacks).

8.1 Attack model

At first the adversary model is presented in this section. Then, three different types of message injection attack (spoofing, fuzzy and DoS attack) targeting the CAN bus is presented in details.

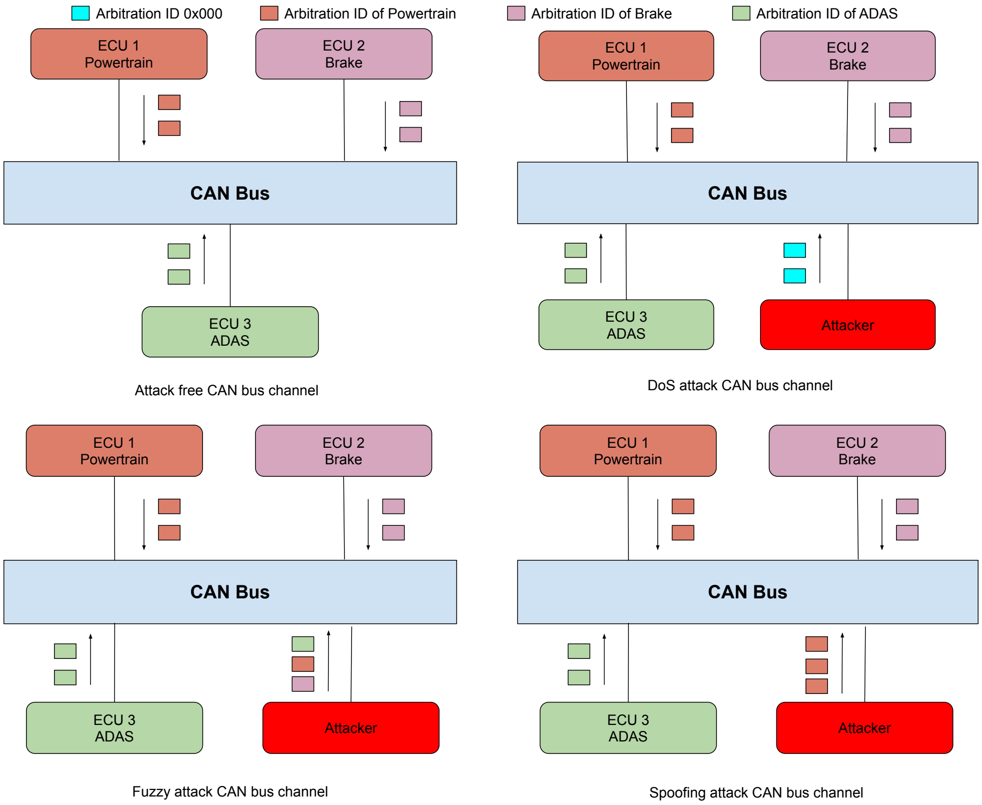
8.1.1 Adversary model

We know that current vehicles are equipped with both wired and wireless interfaces like OBD-II, CD players, USB, Bluetooth, Wi-Fi and cellular (25). We assume that an intruder can use these interfaces as attack surfaces to gain access to the vehicle & try to control the vehicle. As a part of that the hacker may try to inject malicious message to the CAN bus to disable, or paralyze the in-vehicle ECUs' functionalities. The attacks performed by message injection can be defined by three types which are spoofing, fuzzy and DoS attack.

8.1.2 Attack scenarios

The attacker can send unauthorized messages to a CAN network by using the above mentioned attack surfaces and can change the network traffic in a CAN network. We present the traffic of packets of CAN bus channel under attack free normal scenario, spoofing attack scenario, DoS attack scenario & fuzzy attack scenario in this subsection. Figure 8.1 shows the overview of these scenarios.

Spoofing attack: If an attacker injects CAN messages with an specific authorized arbitration ID (AID), he can misguide the vehicle as the receiving ECUs can not distinguish the authorized and unauthorized CAN message with the AID. For example, the intruder can control safety critical functions (disabling brake, shaking vehicle) and non-safety critical (radio on-off, light on-off) functions.



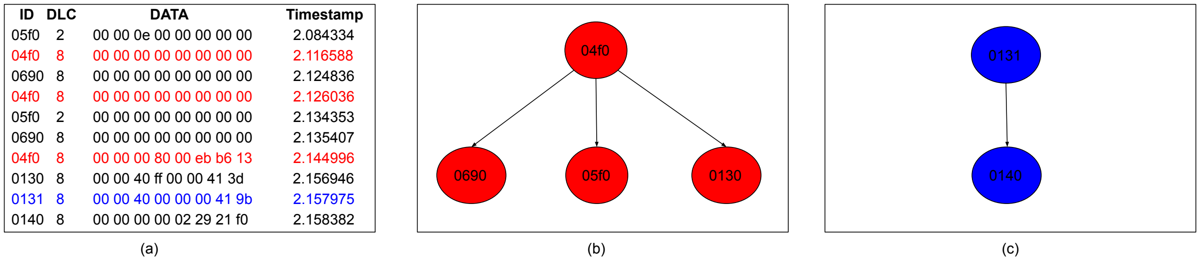
Attack free CAN bus channel

DoS attack CAN bus channel

Fuzzy attack CAN bus channel

Spoofing attack CAN bus channel

Figure 8.1: Attack-free , DoS, Fuzzy and spoofing attack scenarios on a CAN bus.



ID	DLC	DATA	Timestamp
05f0	2	00 00 0e 00 00 00 00 00	2.084334
04f0	8	00 00 00 00 00 00 00 00	2.116588
0690	8	00 00 00 00 00 00 00 00	2.124836
04f0	8	00 00 00 00 00 00 00 00	2.126036
05f0	2	00 00 00 00 00 00 00 00	2.134353
0690	8	00 00 00 00 00 00 00 00	2.135407
04f0	8	00 00 00 80 00 eb b6 13	2.144996
0130	8	00 00 40 ff 00 00 41 3d	2.156946
0131	8	00 00 40 00 00 00 41 9b	2.157975
0140	8	00 00 00 00 02 29 21 f0	2.158382

(a)

(b)

(c)

Figure 8.2: (a) CAN bus data, (b) Informative neighbors of ID 04f0, (c) Uninformative neighbors of ID 0131.

Fuzzy attack: If an attacker sends CAN messages with multiple authorized AIDs in CAN messages, then fuzzy attack happens. Unlike the spoofing attack, the intruder can control multiple vehicle functions at once in fuzzy attack. But it needs more work for the intruder to map multiple AIDs that control vehicle functions.

DoS attack: In DoS attack, the attacker try to utilize the CAN protocol prioritization schema to paralyze the vehicle. To disable the entire vehicle, the attack keeps sending high priority message (theoretically 0x000) and can occupy the channel while making it unavailable for other authorized ECUs.

8.2 Methodology

This section presents the details of the proposed lightweight IDS method. To understand its internals better, first, the conversion of CAN message arbitration IDs into a graph neighbor to neighbor (N-N) relation is introduced in section IV-A. Afterward, graph structural similarity measure techniques using the neighbors of graph nodes are presented and similarity of two blocks of CAN messages based on those techniques is described in section IV-B. Lastly, the final architecture of the ensemble-based IDS is presented in section IV-C.

8.2.1 Conversion of CAN IDs into graph N-N relation

At first, a block of CAN messages is converted into relational directed graphs (vertex and edges) and the neighbors of each CAN arbitration ID (vertex) from the graphs are extracted. In a particular graph, we assume two CAN arbitration IDs are considered to be neighbors or have an edge if they are sequentially transmitted as messages into the CAN bus channel one after another. For instance, If a CAN message with arbitration ID B is sent into the bus channel after a CAN message with arbitration ID A, then B is a neighbor of A. Figure 8.2 shows an example of the process of extracting the neighbors of arbitration ID 04f0 and 0131 from a bundle of the CAN messages. In the same way, a block of CAN messages is converted into the graph N-N relation.

Additionally, every arbitration ID is categorized into two categories which are informative and uninformative based on the number of its neighbors. Any arbitration ID, who has only one neighbor, is considered an uninformative arbitration ID. Otherwise, it is an informative ID. Figure 8.2 (b) and (c) show an example of the informative and uninformative arbitration IDs, respectively.

8.2.2 Similarity between two graphs

The neighbors of the informative graph vertices can be used to determine the structural equivalence of two CAN arbitration IDs as uninformative vertices do not provide much information. According

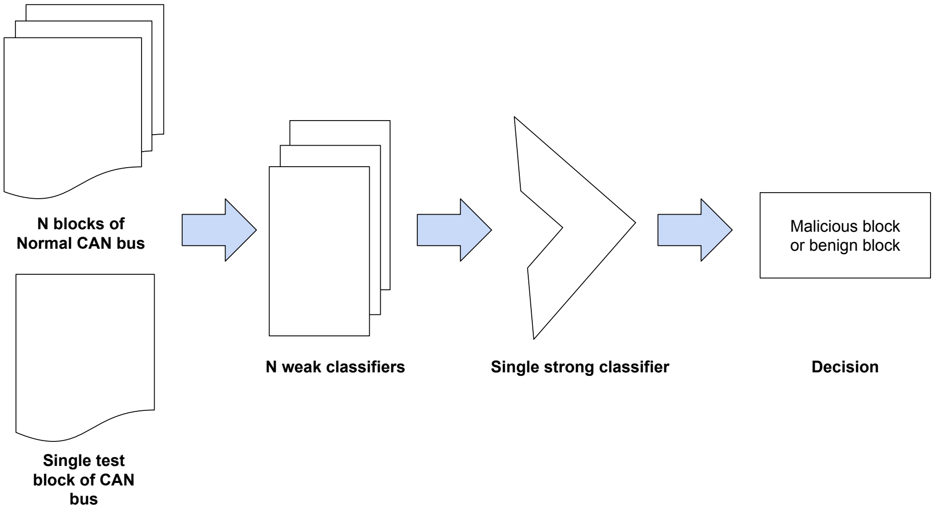


Figure 8.3: Architecture of ensemble-based IDS.

to graph theory, two vertices are said to be equivalent or similar if they have same neighbors. In a real network, perfectly similar nodes are rare to find. Therefore similarity measurement notion like Jaccard score (145), Pearson correlation coefficient (146) etc are used. According to (146), when working with various sized or complex networks, the Jaccard score can not quantify the diversity of graphs, while the Pearson correlation coefficient works in such scenario. Hence, the technique Jaccard is said to be insensitive to the smart attack i.e. spoofing attack, while it works perfectly for suspension attacks i.e DoS & fuzzy. Our experimental result proves that as well. In summary, Jaccard similarity technique is used in this IDS in order to detect suspension attacks such as DoS & fuzzy attack. While the Pearson correlation coefficient is used to detect spoofing attack.

8.2.2.1 Similarity using jaccard score

Jaccard similarity measure technique is a mathematical (145) way to quantify relationship between two vertices of graph.

Based on the process of converting a block of the CAN messages into the graph N-N relation which was described in subsection IV-A, two sets of neighbors of a CAN arbitration ID can be extracted from two blocks of CAN messages. Figure 8.4 shows the similarity of neighbors of an arbitration ID in two different blocks of transmitted CAN messages. It can be observed from figure 8.4 that the intersection area in the figure indicates the similarity of neighbors of an arbitration ID between two blocks of captured CAN message at different times. In other words, the larger overlapping area means the neighbors of the arbitration ID are more similar in the CAN network. Thus the neighbors can be used to determine how similar two graph structure is. This process can be formulated by equation (1) based on the jaccard similarity measure technique (145). This technique provides a numeric value that indicates similarity between two vertices of two graphs. Then vertices similarity is used to determine similarity between the two graphs. Based on the characteristics of the CAN network, only informative nodes are used to determine graph similarity and we consider two graphs are similar if at least 50% nodes are similar between the two graphs.

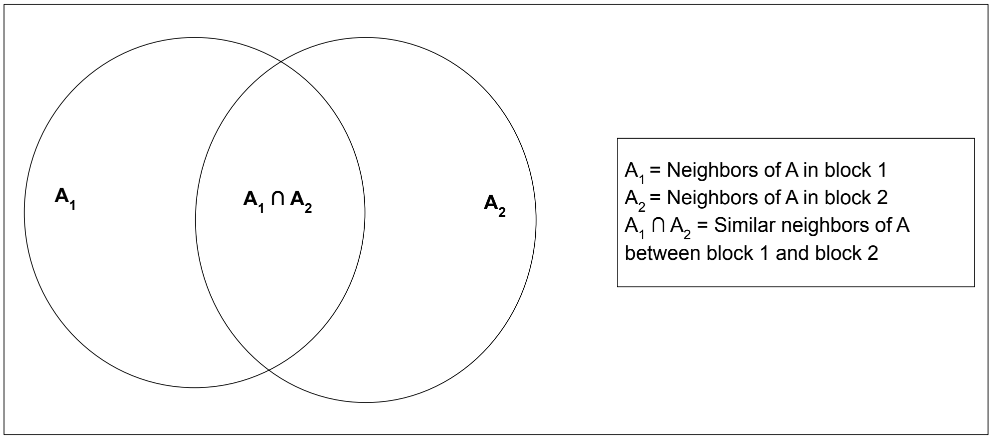

A1
A1 ∩ A2
A2
A1 = Neighbors of A in block 1
A2 = Neighbors of A in block 2
A1 ∩ A2 = Similar neighbors of A between block 1 and block 2

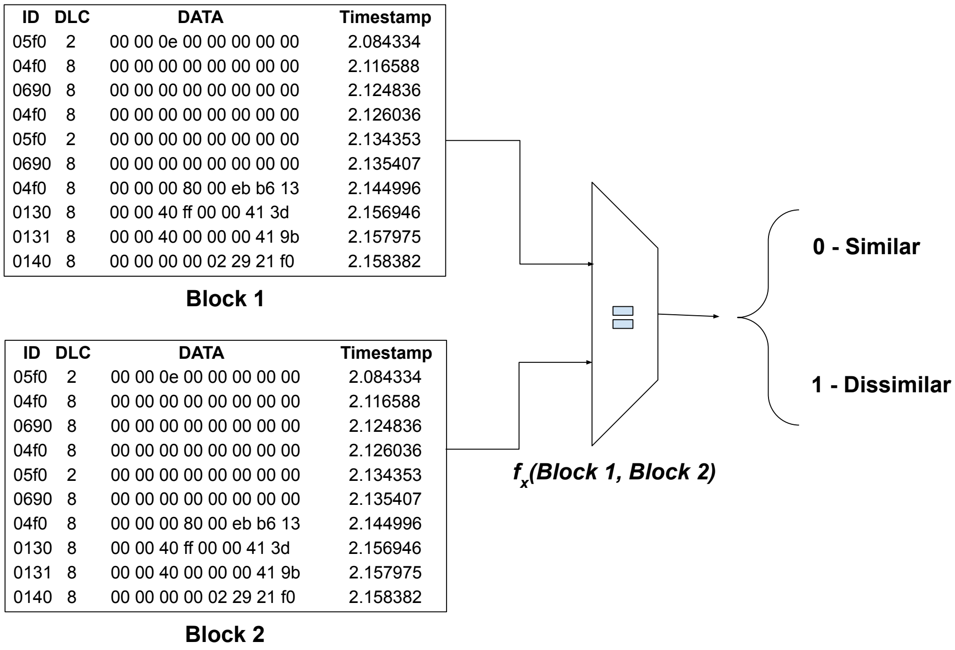
Figure 8.4: Neighbor similarity of a single arbitration ID between two blocks of CAN messages.

Out of them jaccard similarity (145) (for DoS & fuzzy attack) and Pearson correlation coefficient (for spoofing attack) are used to determine similarity equation. Theoretically, DoS attack & fuzzy attack is responsible for the CAN traffic changes for all CAN AIDs. Hence, jaccard similarity method is a perfect fit for the two attacks and seems insensitive to spoofing attack. The experimental observation shown in the result section proves that as well. On the other hand, spoofing attack only changes the CAN traffic of a single CAN AID. So, pearson correlation coefficient algorithm seems perfect for spoofing attack. jaccard similarity equation (145) given in (1) and pearson correlation coefficient is given in (2).

$$Jaccard\ Similarity = \left(\frac{(A_1 \cap A_2)}{(A_1 \cup A_2)} \times 100\% \right) \tag{8.1}$$

8.2.2.2 Similarity using Pearson correlation coefficient

To compare complex networks, another popular technique can be used which is pearson correlation coefficient (146). It can be used in several areas like social media engineering, medical science etc for many years. It can be formulated by equation (8.2). Where, α and β can be in degree or out degree, E is the number of edges in the graph, j_i^{α} is the α-degree of source node i and k_i^{β} is the β-degree of target node j , $\bar{j^{\alpha}} = E^{-1} \times \sum_i j_i^{\alpha}$, $\sigma^{\alpha} = E^{-1} \times \sum_i (j_i^{\alpha} - \bar{j^{\alpha}})^2$, $\bar{k^{\beta}}$ and σ^{β} are similar respectively (146). To find similarity between two CAN graphs, the two graphs are first quantified using this technique and the they are considered to be similar if their quantified values remain



ID	DLC	DATA	Timestamp
05f0	2	00 00 0e 00 00 00 00 00	2.084334
04f0	8	00 00 00 00 00 00 00 00	2.116588
0690	8	00 00 00 00 00 00 00 00	2.124836
04f0	8	00 00 00 00 00 00 00 00	2.126036
05f0	2	00 00 00 00 00 00 00 00	2.134353
0690	8	00 00 00 00 00 00 00 00	2.135407
04f0	8	00 00 00 80 00 eb b6 13	2.144996
0130	8	00 00 40 ff 00 00 41 3d	2.156946
0131	8	00 00 40 00 00 00 41 9b	2.157975
0140	8	00 00 00 00 02 29 21 f0	2.158382

ID	DLC	DATA	Timestamp
05f0	2	00 00 0e 00 00 00 00 00	2.084334
04f0	8	00 00 00 00 00 00 00 00	2.116588
0690	8	00 00 00 00 00 00 00 00	2.124836
04f0	8	00 00 00 00 00 00 00 00	2.126036
05f0	2	00 00 00 00 00 00 00 00	2.134353
0690	8	00 00 00 00 00 00 00 00	2.135407
04f0	8	00 00 00 80 00 eb b6 13	2.144996
0130	8	00 00 40 ff 00 00 41 3d	2.156946
0131	8	00 00 40 00 00 00 41 9b	2.157975
0140	8	00 00 00 00 02 29 21 f0	2.158382

Figure 8.5: Similarity between two blocks of CAN bus message based on neighborhood similarity.

within a certain distance (threshold).

$$r(\alpha, \beta) = \frac{\left(E^{-1} \times \sum_i [(j_i^\alpha - \bar{j^\alpha}) \times (k_i^\beta - \bar{k^\beta})]\right)}{(\sigma^\alpha \times \sigma^\beta)} \tag{8.2}$$

The above-discussed similarity techniques are used to find out the similarity between two blocks of CAN messages. We consider an arbitration ID that appeared into two blocks of CAN messages as similar if its score is more than or equal to a specific threshold value. Otherwise, we consider it as dissimilar. To compare two blocks of CAN messages, out of these one of the blocks contains only attack-free CAN messages and the other one contains unknown CAN messages which can be either attack-free or a mix of attack-free and attacked CAN messages. First, each unknown block of CAN messages is compared to a number of attack-free blocks of CAN messages. Second, if any arbitration ID in the unknown block has a similarity with any of the attack-free blocks greater than a decided threshold value, the arbitration ID is considered to be a benign ID as it has similar neighbors that are extracted from the attack-free Block; otherwise, it is considered as a malicious ID. Third, by following this method, the benign and malicious arbitration IDs can be extracted for the unknown CAN block. In order to reduce the overhead of extracting the benign and malicious arbitration IDs, the IDS does not consider the uninformative arbitration IDs, which have only one neighbor for a block of CAN messages due to its lack of information about other neighbors. Finally, the IDS considers the unknown block to be benign if the number of benign arbitration

IDs is greater than the number of the malicious arbitration IDs, Otherwise, the unknown block is considered malicious. In general, this technique can be used to find out the similarity between any two blocks of CAN messages. We choose to use the arbitration IDs to distinguish CAN message blocks because the attack models discussed in Section III-B use arbitration ID as a weapon to perform the attacks. Figure 8.5 shows the similarity between two blocks of CAN bus messages arbitration IDs based on the neighborhood similarity technique.

8.2.3 Ensemble-based IDS

The main goal of the IDS is to detect message injection attacks in CAN bus by taking a block of CAN messages as input and try to detect the presence of abnormality. To achieve the goal, the proposed IDS first divides the benign (attack-free) CAN bus messages into k graphs, which are modeled to form classifiers, and are called weak classifiers. Each formed graph out of those k graphs has n number of CAN messages. Afterwords Each of the weak classifier compares a single block of unknown CAN messages (test graph) to verify if it is either similar to the the attack-free graph or not. If the similarity is determined, then, the test block of CAN bus messages is considered normal. Otherwise, the IDS considers it as an attack. Finally, In order to make the final decision, the k weak classifiers are assembled to produce a strong classifier that considers the majority voting method (147) as a basis to identify an unknown block. The IDS considers the graph as attack-free if more than 50% of weak classifiers vote the test graph is similar to graphs constructed using benign CAN messages. Otherwise, the graph is labeled as attacked graph. Figure 8.3 shows the final architecture of the ensemble-based IDS.

8.3 Experimental result

In this section, the performance of the proposed IDS is evaluated via simulation on a real CAN bus dataset (87). Attack-free, spoofing, fuzzy and DoS attacks' related data files are used in the experiment. These data files consist of malicious and benign CAN messages. So, in order to evaluate the robustness of the proposed IDS, different performance evaluation metrics are considered such as the attack detection rate, the attack-free detection rate, the accuracy, precision, recall, F1-score and detection time. In the following subsections, the selection of simulation hyperparameters and the simulation results subsections are presented one by one at first. Then, the comparison between the proposed IDS and state-of-the-art is presented to end the performance evaluation.

8.3.1 Description of the dataset

The overall performance of the IDS is evaluated on a publicly available real vehicular dataset (46) focusing mainly on normal, spoofing, fuzzy and DoS attacks CAN bus messages. Table 8.1 shows a detailed overview of the used dataset for our experiment. Dataset spoofing, fuzzy and DoS attacks have a mix of benign and malicious CAN messages, whereas, the attack-free dataset has only normal (benign) CAN messages. From the fuzzy attack dataset and spoofing attack dataset, 12485 and 11,812 blocks of malicious CAN messages were extracted respectively while the DoS attack dataset has 12653 blocks of a malicious CAN message. The attack-free dataset does not have any malicious CAN messages. So, the number of blocks of benign CAN message for the attack-free dataset, fuzzy attack dataset, DoS and spoofing attack dataset is 18565, 6709, 5675 and 11,296 respectively.

Table 8.1: CAN traffic dataset details

Dataset	**No of CAN messages**	**No of CAN blocks**	**No of attack-free blocks**	**No of attack blocks**
Attack free	37,13,000	18,565	18,565	0
Fuzzy	3,838,860	19,194	6,709	12,485
DoS	3,665,771	18,328	5,675	12,653
Spoofing	4,621,702	23,108	11,296	11,812

8.3.2 Selection of hyperparameters

In order to achieve the goal of an efficient and robust IDS, two parameters are tuned and the optimal value of each parameter is determined. These parameters are the block size of CAN messages and the number of weak classifiers.

8.3.2.1 Impact of block size

The main objective of this subsection is to find the impact of block size on the accuracy of the proposed IDS. Block size is the number of CAN messages that eventually fed as input to the weak classifiers. The block size parameter is selected by performing an experiment on the attack-free CAN dataset which has 28 unique arbitration IDs. In order to find the optimal block size, we need most of the arbitration IDs of the CAN bus channel to appear in each block in order to have the maximum information about the CAN bus channel. Therefore, the percentage of unique arbitration

IDs across different sizes of benign CAN message blocks is determined as shown in Figure 6.6 . According to figure 6.6, a block size of 200 CAN message allows 90% of total arbitration IDs of the CAN bus channel to appear in each block. Although the block size of 500 messages increases the unique appearance of arbitration IDs by only 1% in each block, nevertheless, it increases the iteration of neighbor extraction of each block by 300 messages. Subsequently, it yields more latencies on processing each block. Therefore, the block size is chosen to be 200 messages per block.

8.3.2.2 Number of the weak classifiers

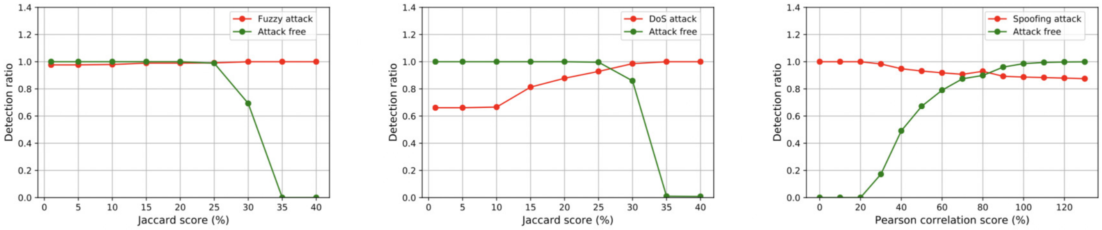
To classify an unknown block of CAN messages, the block was compared with several blocks of attack-free CAN messages. Each comparison task is called a weak classifier. In order to determine the optimal number of the weak classifiers, we placed a hypothesis to test the proposed IDS on the DoS attack dataset which is more critical than spoofing and fuzzy attacks, therefore, we tested our IDS on 651 blocks of the DoS attack dataset across a different number of the weak classifiers. The performance of the IDS based on our hypothesis is observed across a different number of weak classifiers as displayed in Table 8.2. According to Table 8.2, we find that using 500 classifiers provides the highest detection rate and the minimum processing time of classifying a single unknown-block, whereas using 1000 classifiers requires processing time about two times of using 500 classifiers to classify a single unknown-block even though using 1000 classifiers provides less detection rate than using 500 classifiers. Therefore, we chose 500 to be the optimal number of weak classifiers. To validate our hypothesis, the results section demonstrates that our hypothesis is successfully able to provide high accuracy on detecting not only DoS attack but also fuzzy attack and attack-free (normal CAN messages).

Table 8.2: Comparison of weak classifiers

No of weak classifiers	**DoS attack detection rate (%)**	**Single block processing time (ms)**
500	93.85	156.02
750	93.85	227.49
1000	93.54	296.4

8.3.2.3 Selection of the optimal threshold

We consider an arbitration ID as benign if it has a certain overlapping or similar neighbors in two blocks of CAN messages. We use a Jaccard similarity score to calculate the similarity of



(a) Fuzzy attack (b) DoS attack (c) Spoofing attack

Figure 8.6: Detection ratio by Jaccard score of (a) Fuzzy, (b) DoS and Pearson correlation coefficient of (c) Spoofing attack

neighbors of an arbitration ID between two blocks of CAN messages in terms of DoS & fuzzy attack. Those two blocks are considered similar if the Jaccard similarity score is greater than or equal to a specified threshold value. In order to determine the optimal Jaccard score threshold value, we tested our system on attack-free, DoS attack and fuzzy attack datasets across different threshold values as shown in Figure 8.6 (a) & (b), it is observed that using a threshold of 25% provides the highest detection rate of the attack-free, DoS attack and fuzzy attack. Consequently, we chose 25% to be the optimal value of the Jaccard similarity score.

When working with smart attacks i.e spoofing, the Pearson correlation coefficient is used to find similarities between two graphs. As stated in subsection IV-B, perfectly similar graphs are unlikely to found in the real world, a threshold is needed to be tuned. In this situation, the threshold is determined in terms of attack detection rate and attack free detection rate and is calibrated by changing the percentage distance between Pearson correlation coefficient of the two graphs. According to the experimental result, it is decided that a threshold of 80% provides the highest detection rate for spoofing attacks.

Table 8.3: Experimental results for attack free, fuzzy attack, DoS attack & spoofing attack

Dataset	**Attack-free detection rate (%)**	**Attack detection rate (%)**	**Accuracy (%)**	**Detection time (ms)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Attack-free	99.21	N/A	99.21	186.86	100	99	100
Fuzzy attack	99.38	99.51	99.46	183.24	99.38	99.45	99.41
DoS attack	92.86	99.66	97.55	149.64	98.03	96.26	97.09
Spoofing attack	89.89	92.93	91.01	73.61	91.39	91.41	91.37

8.3.3 Simulation results

To evaluate the performance of the proposed IDS, our algorithm is developed using Python programming language on a 1.8 GHz laptop with 16 GB of RAM, windows 10 os and it is tested on attack-free, spoofing, fuzzy and DoS attack datasets(87). As mentioned earlier, the attack detection rate, the attack-free detection rate, the accuracy, precision, recall and F1-score have been calculated by using the equations are given in (143), respectively. Also, the detection time which is the time required to identify an attack or attack-free block has been considered to evaluate the performance of the proposed IDS. Table 8.3 shows the results of attack-free, fuzzy, DoS and spoofing attacks. It can be observed that our IDS is able to detect the attack-free, fuzzy, DoS and spoofing attack with the accuracy of 99.21%, 99.51%, 99.66% and 92.93% respectively. Furthermore, we observe that our IDS can detect a fuzzy attack with 99.38% in less than 0.2 second with 99.46 accuracy and the DoS attack with 92.86% in only 0.15 second with 97.55% accuracy while using the Jaccard similarity technique. When using the same technique for detecting spoofing attacks the proposed IDS achieved an accuracy of 66% only. While using the Pearson correlation coefficient technique the proposed IDS achieved an accuracy of 91.01% with an attack detection rate of 92.93 % for spoofing attacks. The detection time is surprisingly less than fuzzy and DoS attack detection and it is 73.61 ms. Additionally, our IDS is able to classify a single block of CAN messages from an attack-free dataset within 186.86 ms on average.

$$\textit{Attack Detection Rate} = \frac{TP}{TP + TN} \tag{8.3}$$

$$\textit{Attack-Free Detection Rate} = \frac{FP}{FP + FN} \tag{8.4}$$

$$Accuracy = \frac{TP + FP}{TP + FP + TN + FN} \tag{8.5}$$

$$Precision = \frac{TP}{TP + FP} \tag{8.6}$$

$$Recall = \frac{TP}{TP + FN} \tag{8.7}$$

$$\textit{F1-Score} = 2 \times \frac{precision \times recall}{precision + recall} \tag{8.8}$$

Where TP (True Positive) is the number of the correctly identified attacked blocks, TN (True Negative) is the number of the incorrectly identified attacked blocks, FP (False Positive) is the number of the correctly identified attack-free blocks and FN (False Negative) is the number of the incorrectly identified attack-free blocks.

8.3.4 Comparison with state-of-the-art

Finally, the proposed IDS is compared with one of the state-of-the-art works (3) that converts CAN bus data into graphs and detects intrusion by performing statistical analysis. Therefore, the effectiveness of our proposed approach is evaluated by comparing with (3) using the same real vehicular CAN message datasets (87). The experiment is done in the same computer environment also.

The experimental result shows, unlike (3), our IDS can detect if a single graph (200 CAN bus messages) is malicious or benign , while (3) requires a distribution of graphs (300,000 CAN bus messages) to detect attack. Apart from that, the comparison has also done in terms of accuracy of combined attacks (DoS & fuzzy attacks and DoS, fuzzy & spooging attacks) shown in table 8.4. According to our experiment, the proposed IDS achieved a combined accuracy of 98.53% for DoS & fuzzy attack, while (3) achieved 86.84% combined accuracy while using the chosen dataset. When defending against a combination of DoS, fuzzy & spoofing attack the proposed IDS showed 5.85% lower misclassification rate than the state-of-the-art (3).

Table 8.4: Comparison with the state-of-the-art (3) in terms of attack detection accuracy against combined attacks

Approach	**DoS & fuzzy attack (%)**	**DoS, fuzzy & spoofing attack (%)**
Propose IDS	98.53	96.01
(3)	86.84	90.16

8.4 Discussion

The need for a computationally efficient IDS for modern vehicles is becoming one of the most essential security components as these vehicles are exposed to a huge number of threats. In this chapter, we introduced a lightweight IDS for the CAN bus which can detect message injection attacks on the CAN bus based on the analysis of the graph neighborhood similarity. We developed a novel algorithm to detect anomalies on transmitted CAN bus by comparing any single block of CAN messages with blocks of attack-free CAN messages. We evaluated our approach against three types of message injection attacks which are spoofing, fuzzy and DoS attacks on real vehicular

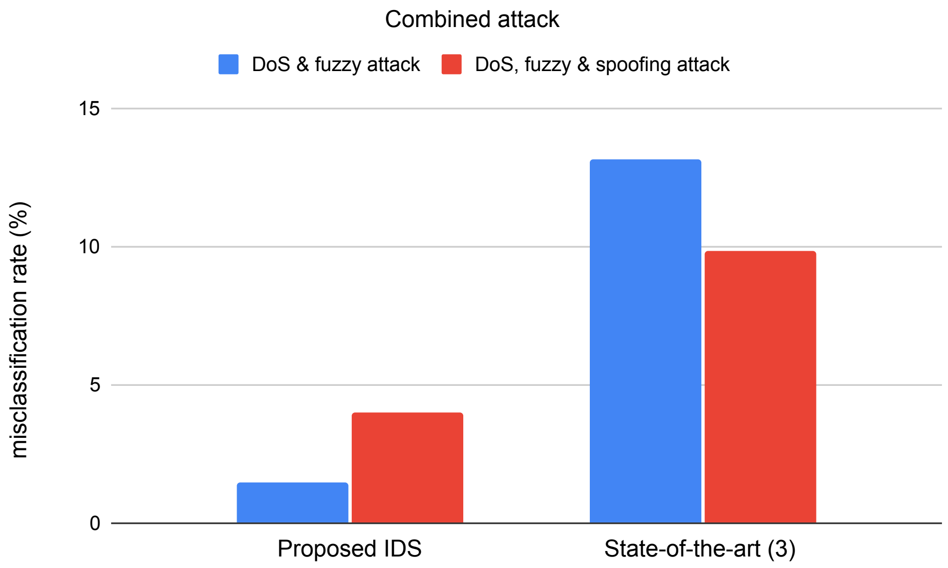


Figure 8.7: Comparison between the proposed IDS with the state-of-the-art(3) using combined attack datasets.

CAN bus data. The results demonstrated that the proposed IDS can successfully detect spoofing attacks, fuzzy attacks and a DoS attack which can be the most perilous attack for vehicles in minimal time with an accuracy of 96.01%. We believe that our IDS contributes to improving vehicle security without making any change in the CAN protocol. As future work, we plan to evaluate our system against other attacks. Finally, to confirm practicality, an empirical experiment needs to be carried out to test our approach under various real-world scenarios.

CHAPTER 9

Physical Fingerprinting via Deep Learning

To solve the above-mentioned IVN security vulnerability, different approaches have been implemented by security researchers (148; 149; 150; 151). These solutions can be broadly categorized into two categories. (1) cryptography based solutions (152; 153; 154), (155), (2) intrusion detection system based solution (156; 157; 158). The traditional cryptography-based solutions can provide some degree of security but they are computationally expensive and uses the network bandwidth which is critical for CAN-based vehicle networks (7). Moreover, these cryptography-based solutions are vulnerable to replay attack (156). Recently, researchers have proposed intrusion detection system-based solutions for detecting CAN cyberattacks by implementing the famous physical layer identification (99) techniques (6),(7),(159). The fundamental idea of this approach is, the analog signal behaviors of data transmitters have slight variations which are introduced in the design, fabrication and manufacturing process. Researchers show that even manufactured in the same production lot, two same digital devices have unique artifacts in their signaling behavior which is difficult to control and duplicate (160). Avatefipour et al. were able to extract those unique artifacts and proposed a framework based on a neural network for CAN sender identification by utilizing the extracted distortions (6). Likewise, in the last 5 years researchers (7; 98; 161; 162) have proposed a lot of frameworks that are effective in CAN transmitter identification. While, the frameworks offer a high percentage of accuracy, but the core architecture of these methods depends on handcrafted feature engineering. As the approaches rely on neural network-based methods, feature engineering remains an essential step in the testing phase of the framework. In some cases, the feature engineering becomes computationally so expensive, that the real-time sender identification remains a challenge. Additionally, a significant amount of data is required for training neural networks, and obtaining sufficient data poses a challenge for automotive platforms with limited resources (7). Therefore, the high cost of feature engineering and the limited availability of training data, specifically physical voltage signals, are the primary factors hindering the application of deep learning in physical fingerprinting research for in-vehicle automotive purposes. Hence, this chapter aims to answer the following research question: ”is it feasible to identify the sender of the CAN message using an affordable approach that employs deep neural networks?”

In order to integrate a transmitter identification strategy to the existing CAN protocol, this chapter proposes a framework that is based on the intersection between physical layer identification (99) and computer vision technique. To fingerprint, the proposed framework first extracts distortions from the analog signals sent by the ECUs. Then the distortions are converted into visual representation (images) by using recurrence plot technique (163) which are are distinctive in human eyes. To automate the process of ECU identification, the images are fed into deep neural networks (Mobilenetv2 (164) and EfficientNet (165)) to learn patterns of the signals from those generated images. To eliminate the requirement of large amount of training data, the framework utilizes transfer learning (166) approach and retrains pretrained models to solve sender identification. Finally, the trained model is tested to evaluate the performance of the proposed framework. According to the evaluation, it achieves better accuracy in identifying the CAN senders and is lightweight in terms of computational cost.

The main contribution of the chapter is as follows:

- To the best of our knowledge, this is the first CAN sender identification framework that utilizes the concept of deep learning based computer vision task and transfer learning.
- The framework takes advantage of recurrence plot to visualize the dynamics of the senders to fingerprint ECUs.
- Based on the experimental settings with 8 ECUs, the framework identifies senders with an accuracy of 98.34% where 0.05 ms is needed to process a feature of a single observation for identification.
- The framework does not change the underlying architecture of basic CAN protocol, thus making it applicable to all CAN protocol based vehicles.

9.1 Spoofing attack model

Due to the absence of sender or receiver address as discussed above, CAN network is susceptible to spoofing attack (6; 7; 167; 101). The attack can be defined as when a compromised ECU tries to send CAN data by impersonating an authorized ECU with same or different CAN AID. In a modern vehicle, this can happen by two different scenarios. One is when an attacker takes control of an authorized ECU utilizing its code vulnerabilities. The compromised ECU can impersonate any other authorized ECU connected to the network. And the other scenario is when an attacker gains access to the vehicle by external connectivity (e.g. via OBD-II port or using WiFi or bluetooth). It is an feasible attack example as having OBD-II on every vehicle is considered an automotive standard, with pin 6 and pin 14 representing the CAN interface that can be used to connect external

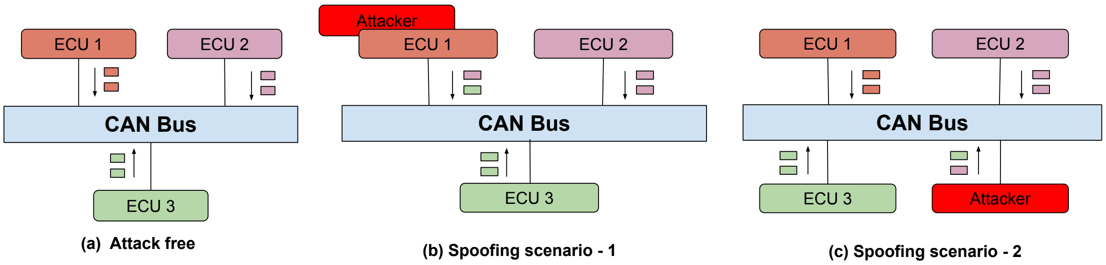


Figure 9.1: Spoofing attack in CAN network

devices (168). Lets consider a CAN network with 3 ECUs as shown as Figure 9.1 (a), where the color of the message represents the unique arbitration ID and data payload for a corresponding ECU. subfigure (b) and (c) represents two scenarios where, subfigure (b) shows where an attacker took control of an existing ECU and try to spoof ECU 2 and ECU 3. On the other hand, subfigure (c) shows one spoofing attack where the attacker gains access to the bus as an external entity and impersonate ECU1 and ECU 2. As the CAN protocol lacks authentication capability by default, it is an challenge for receiving ECU to determine whether the message is benign or malicious. This kind of attack is easily implementable in today's vehicle as the chapter performed in section 9.3.6.

9.2 Methodology

In this section the proposed framework for identifying CAN message senders is described. The phases of this methodology is described in a bottom up fashion, where the core idea of physical layer identification in the subsection A is presented first. Then the technique of image generation using the physical characteristics is describes in subsection B and finally in subsection C, the entire proposed framework is explained.

9.2.1 Linking CAN signal to the transmitter

Physical layer identification is a popular concept for identification of senders in connected networks for so many years (99; 169). The fundamental idea of this approach is, the behavior of senders in terms of analog signal has slight variations. The differences are introduced in the design, fabrication and manufacturing process, even two identical digital devices that are manufactured in the same production lot, have unique artifacts which is difficult to control and duplicate (6; 7; 101). In a practical world, although it can be reproduced by reverse-engineering, but the process is difficult if not impossible for a determined attacker. Fig. 9.2 illustrates the amount of inherent variation between two different CAN transmitters.

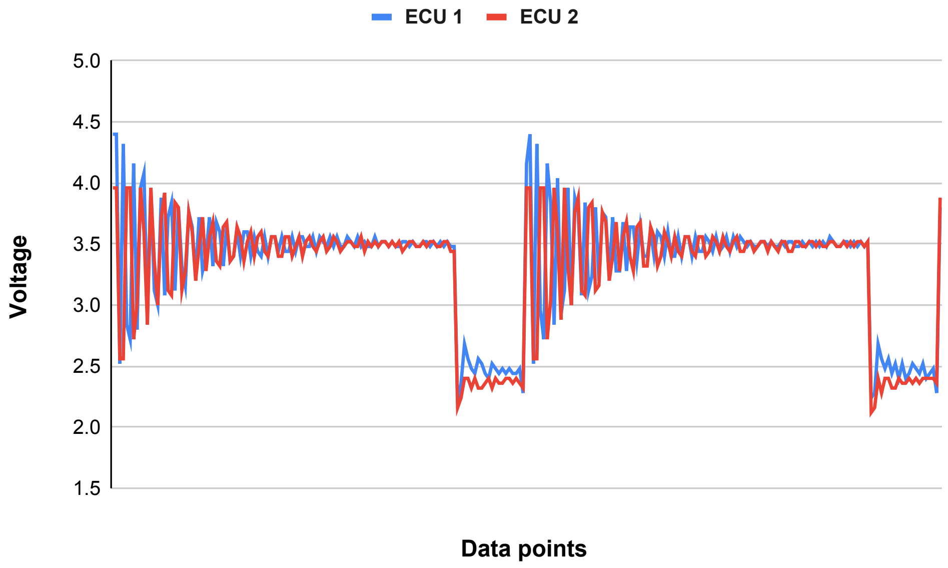


Figure 9.2: Analog signal difference of two ECUs

The framework uses the above mentioned inherent variation of the CAN transmitter and uses them to fingerprint the transmitter as it is unique. The Figure 9.3 shows how a CAN signal stays in an idea condition and how it distorts in practical world. The spikes from the idea line is considered as the impurity of each CAN transmitter which is identical to it. The proposed work uses it to create a unique signal characteristics profiling for transmitters.

Again the question remains how to extract the tiny variations? Which is called distortions of the analog voltage. Lets assume, V is a collection of analog voltage signal captured from the CAN-H wire where,

$$V = (V_1, V_2, V_3....V_n) \tag{9.1}$$

Ideally, V_i should be 3.5 when it is a dominant bit and 2.5 when it is a recessive bit. In real world, the unique artifacts add noise to the ideal value and creates spikes (see 9.3). In order to extract the unique variations, the spiking points needs to be subtracted from 3.5 or 2.5 depending on it is a dominant or recessive bit. So, the unique artifacts (Distortions, D_i) of an ECU is,

$$D_i = (V_i - T_j) \tag{9.2}$$

where T_j is either 3.5 or 2.5 depending on if the bit is dominant or recessive.

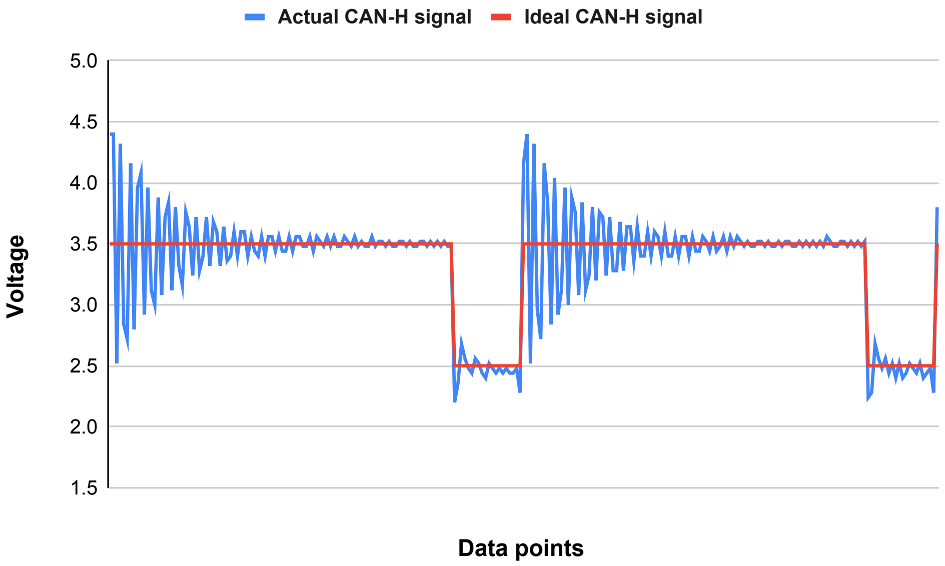


Figure 9.3: CAN-H signals with unique artifacts

9.2.2 Representing signal profiling by recurrence plots

In this phase of the proposed method, the extracted unique slight variations of CAN senders are used to create images for each transmitter. The variations are turned into images via the recurrence plot (RP) (163) technique where each image represents the pattern of a sender. The initial purpose of recurrence plots was to create a visualisation of the recurrences of a system's states in a phase-space (with dimension n) within a small deviation ϵ. That means, a recurrence of a state at time i and at a different time j is marked within a two-dimensional squared matrix with ones and zeros (black and white points in a plot), where both axis represent time. The RP can be formally expressed by the following matrix in equation 3 (163).

$$R_{ij} = \Theta(\epsilon - |\overrightarrow{D_i} - \overrightarrow{D_j}|) \qquad i, j = 1....n \tag{9.3}$$

Where D_i and D_j is the distortions extracted in equation 2. The matrix can be used to create an image which is actually a correlation plot. The image is the representation of a CAN transmitter as it is created form the unique physical characteristics. The proposed work uses the images in identifying the source of CAN signals by considering it as an image classification problem.

9.2.3 Source identification of ECUs

Finally, the framework proposes a framework that uses the above mentioned steps to identify the source of CAN ECUs. Figure 9.4 shows overall architecture of our source identification framework. The proposed architecture first extracts the unique artifacts of the CAN transmitters. Then the recurrence plots are created from the extracted distortions which are a representation of uniqueness of CAN transmitters. Finally, the recurrence plots are used to train and test a deep learning model. In Figure 9.4 the green line shows the training phase and the red line represents the testing phase of the system. The data processing needed for the framework is to extract the unique artifacts and the creation of recurrence plot, which is same for both the training and testing phase. For the selection of deep learning architecture, we have chosen two popular network MobileNetV2 (170) and EfficientNet (165) for the experiment.

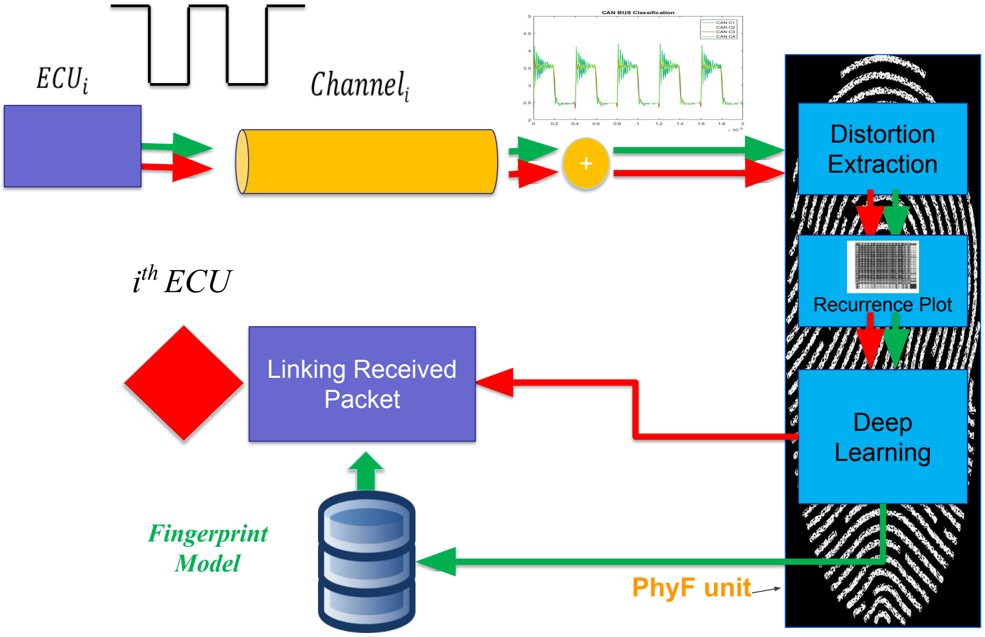


Figure 9.4: Source identification of CAN message senders

9.3 Experimental result

In this section the effectiveness of the proposed methodology is evaluated by conducting experiments in the laboratory where subsection *A* presents the experimental setup. It is followed by subsection *B*, *C* i.e. the generation of recurrence plot of CAN senders and the performance of the proposed framework consecutively. Then the effect of environmental factors over the proposed framework is discussed in subsection *D* and effect of information aware downsampling is presented in subsection *E*. Finally, the performance of spoof detection in a vehicle test bench and the comparison with the state-of-the-art is presented in subsection *F* and *G* respectively to conclude the section.

9.3.1 Experimental setup

To verify the effectiveness of the proposed framework, an experiment is designed (Figure 9.5) on a CAN protocol based test bed that has total 8 ECU built by Arduino Uno microcontrollers connected to CAN transceivers where each ECU has 1 meter channel length. Physical signal for each ECU is captured using a DSO1012A oscilloscope with a sampling rate of 2GSa/s, 100MHz bandwidth, and 8-bit vertical resolution. The data was collected in a laboratory environment from the CAN-H pin which ideally ranges from 3.5v to 2.5v. Multiple programming languages are used in this experiment as the microcontrollers are programmed using C programming language and Python is used for training & testing deep learning models and result analysis. The experimental testbed is set up in a plug and play mode, because some of the experiments were done using 4 ECUs and some of the experiments were conducted using 8 ECUs. To check the performance of the proposed framework on a real vehicle, an experiment is designed on a vehicle test bench that is based on the GM Sierra 2020 model for spoof detection also (elaborated extensively in subsection *F*).

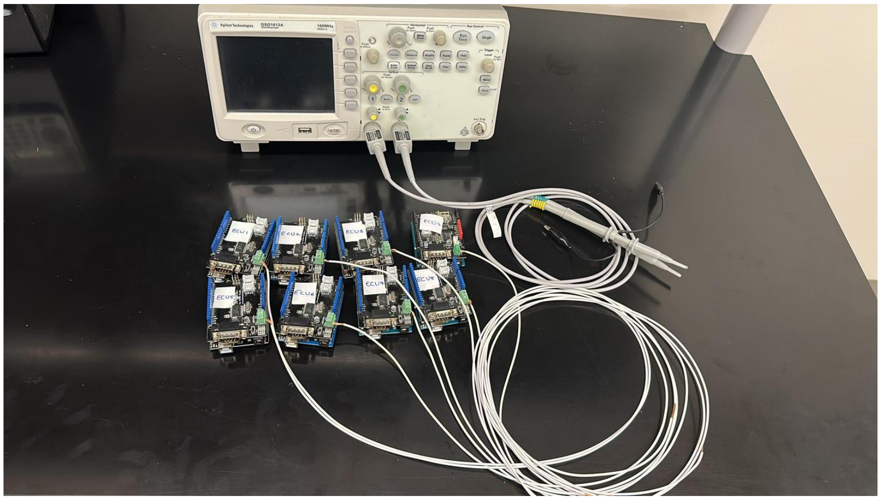


Figure 9.5: Experimental settings

9.3.2 Generation of recurrence plot of CAN senders

The goal of this subsection was to create recurrence plot by using distortions captured from the CAN transmitters and visualize them in human eyes. To perform that experiment we collected analog signals from 4 ECUs using the testbed described in subsection *A* and extracted the distortions of the ECUs. The distortions are mapped to create recurrence plot and saved as images for visualization. Each image was generated from 96 voltage data points (length of CAN-H dominant bit) started from the peak of the voltage signals. Figure 9.6 shows the generated plots of 4 ECUs where each row has images for each ECU. It indicates that, the images has their own patterns and they are different to each other visually.

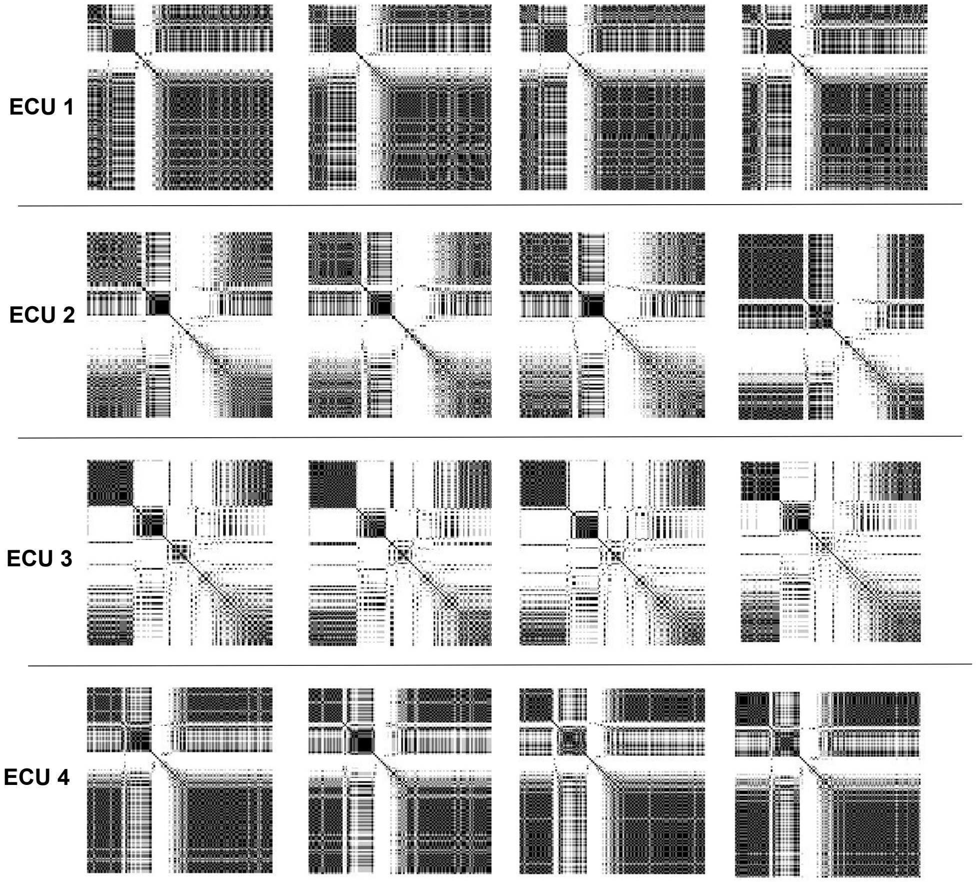


Figure 9.6: Recurrence plot representing 4 ECUs

Visually RPs provide some useful insights about the sender ECUs. But the question arises how much information the RPs contain to distinguish the CAN transmitters. In order to do so, an experiment was deigned to quantify the RPs generated from the CAN ECU signals. To achieve that, a recurrence quantification analysis was performed to quantify the RPs by extracting recurrence properties from the generated RPs. We used recurrence parameters (171) such as recurrence rate, entropy diagonal lines, longest diagonal line length, laminarity, divergence, additional diagonal line length to perform data analysis. To do so, a Python program is written to generate RPs from CAN high analog voltages collected from 8 ECUs and then the images are used to perform recur-

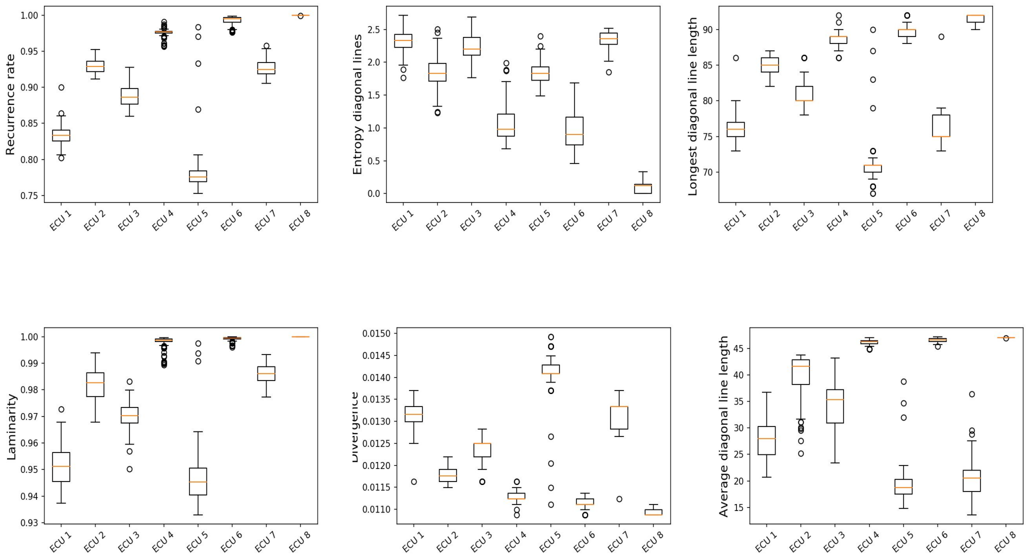


Figure 9.7: Feature differences of recurrence quantification parameters

rence quantification analysis (RQA) in an Apple M1 chip computer with 8 GB RAM. For RQA, the images are fed into python library PyRQA (172) and the parameters are extracted for rigorous analysis. To see the feature differences of the ECUs in terms of RQ parameters the data is plotted in a box plot shown in Figure 9.7. The figure shows that the 8 ECUs have notable variations when compared against the 6 recurrence parameters.

9.3.3 Performance of the proposed framework

This experiment evaluates the accuracy of sender identification in a CAN network using the proposed framework. The main goal of this subsection is to demonstrate the applicability of image classification via deep learning models in CAN physical fingerprinting. To achieve this, the proposed framework is validated against deep learning networks using (Mobilenetv2 (164) and EfficientNet (165)) architecture. First the data from 8 ECUs are collected from the testbed described in subsection *A*, then data processing which involves extraction of distortion & image generation is done in an Apple M1 computer with 8 GB RAM and finally, the training and testing of deep learning architecture is performed in a Google-Colab environment. The code for the data processing, model training and model testing is written in Python programming language.

To conduct the experiment 131,760 analog voltage data points are gathered in total and 1098 images were created as described in subsection *B* where each ECU has 250 images and each image has a dimension of 192X192 pixels. To handle the smaller number of images we used transfer learning (173), where a trained model is tuned to solve a problem which is unknown to

Table 9.1: CAN sender identification using deep learning models

Algorithm	Accuracy(%)	Feature processing time (ms)
Efficientnet	95.04	0.07
Mobilenetv2	97.52	0.07

the trained model. In order to do so, a pre-trained MobileNetV2, trained with a public dataset (imagenet (174)) with 1,700,505 parameters is selected and it's weights are used to retrain the model to solve the CAN sender identification problem. For retraining the model, 70% data was used, while the retrained model is tested with 15% and validated with remaining 15% data. Table 9.1 shows the result of the simulation. It indicates, Mobilenetv2 architecture achieves a maximum validation accuracy of 97.52%. To check the performance of the proposed methodology while using a different deep learning algorithm, the same experiment was repeated using an EfficientNet model pre-trained on imagenet dataset (174) with 5,338,572 parameters. It was re-trained using 70% data and tested using 15% data where the image dimension was 224X224 pixels. Finally, the performance of the methodology was measured by validating the trained model with 15% remaining data. The experimental result shows that the EfficientNet achieves a validation accuracy of 95.04%. While using both MobileNetv2 and EfficientNet it takes 0.07 ms per image on average for data processing task which involves noise extraction and image creation.

9.3.4 Effect of environmental factors on the proposed IDS

This subsection represents the analysis of the effect of environmental conditions on the performance of the proposed framework. It is important because, the foundation of the proposed methodology is image classification where the images are created from the distortions present in electrical signals and the signal characteristics are sensitive to environmental factors like temperature, amount of moisture contamination, aging, etc. (175). These factors, if not accounted for, could lead to incorrect identification of the senders in real-world scenarios. In order to verify their effect, data is collected from a setup testbed with 8 ECUs and then analysis is performed to measure performance of the proposed framework under the presence of noise that may be produced by environmental factors. To add the noise, a simulation is created by adding Additive White Gaussian Noise (AWGN) (176) of different percentage of the voltage distortions (1%, 2%, 3%, 4%, 5% & 10%) to the voltage signals. The overall experiment is divided into two different steps, first one is adding AWGN to the electrical signals and creating the images by using the noisy distortions. Figure 9.8 shows the distorted images that are generated from different level of noisy voltages. in the subfigure (a) represents an image generated without AWGN, while subfigure b,c,d,e,f,g represents images with 1%, 2%, 3%, 4%, 5% & 10% added AWGN. The noiseless and noisy figures clearly indicates that the noises caused by environmental factors has significant effect on the images generated by the voltage distortions while there is deviation from the original image increases with the addition of level of noises to the voltages.

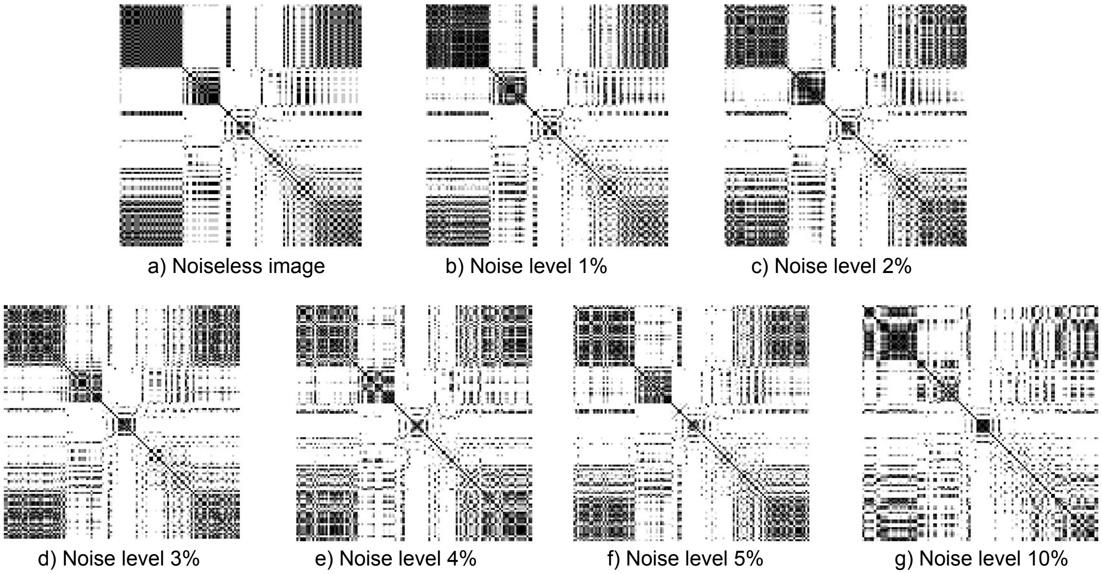

a) Noiseless image
b) Noise level 1%
c) Noise level 2%
d) Noise level 3%
e) Noise level 4%
f) Noise level 5%
g) Noise level 10%

Figure 9.8: Images generated from voltages under different environmental conditions

In the second step of the experiment, a deep learning model is trained using the noiseless original noise, while the images with noise is tested against the trained model and the model performance is evaluated in terms of sender identification accuracy (shown in Table 9.2). While introducing 1% noise in the testing data the performance of the proposed framework degrades by a 30.23% so the proposed model is sensitive to environmental noise. To check the performance of the proposed approach when the model is retrained, again the trained model is retrained by adding images with 1% AWGN and tested against noisy images. Later the trained model was retrained with 2% noise and model performance against noisy images was evaluated again. The experimental result is summarized and shown in Table 9.2. According to it, when the generated images with 1% AWGN are introduced during the model training with noiseless images, testing accuracy improves significantly for noisy images (1%, 2%, 3%, 4%, 5% and 10% GN). Although the training data had only noisy images with low AWGN (1%), the trained model was able to classify noisy images with an improvement of maximum 34.47% and minimum 19.37%. Again, when the model was again retrained by introducing noisy images with 2% AWGN and the model can identify senders with an maximum upgrade of 9.2% in terms of accuracy. So, it can be concluded that, the proposed framework is performs better if the model is retrained with noisy images.

Table 9.2: Effect of environmental factors over the proposed framework

Training images			**Testing images**	**Model performance**
Noiseless	1% AWGN	2% AWGN	AWGN (%)	Accuracy (%)
Yes	No	No	0	98.34
Yes	No	No	1	68.11
Yes	No	No	2	61.09
Yes	No	No	3	55.81
Yes	No	No	4	51.05
Yes	No	No	5	47.00
Yes	No	No	10	33.47
Yes	Yes	No	1	94.77
Yes	Yes	No	2	95.56
Yes	Yes	No	3	92.31
Yes	Yes	No	4	85.45
Yes	Yes	No	5	80.94
Yes	Yes	No	10	52.84
Yes	Yes	Yes	2	96.82
Yes	Yes	Yes	3	95.99
Yes	Yes	Yes	4	94.65
Yes	Yes	Yes	5	89.13
Yes	Yes	Yes	10	61.52

9.3.5 Effect of selective (information aware down sampling) sampling on the proposed framework

Since, the amount of data to be processed for generating each image has a larger influence on the required computing power, a major goal is to reduce the required amount of sampling points. To reduce the sampling points considered to create the image, an experiment with rigorous analysis is conducted. If we look carefully, the backbone of the methodology is the images which are created from the distortions of the ECUs. Again, the distortions are created from analog voltage signal of the CAN signals. Figure 9.9 shows the plot of analog signal captured form an ECU and it is clearly visible that, the signals has spikes at the beginning and gradually it settles down in terms of voltage. From that we can infer that, the distortions which is extracted from the overshoot portion of analog signals (marked as a red box in Figure 9.9) holds significant unique information which is vital in sender identification. and after that we have data points that are less informative. Based on that observation, an experiment was conducted where images were generated by varying the informative and uninformative data points. Then the images are fed into a MobileNetV2 model for evaluation. So, in order to verify that an simulation is designed where images generated by three approaches are tested against MobileNetV2 model and the validation accuracy are evaluated. They are,

- **Truncated sampling**: images generated using all the informative points .
- **Custom odd sampling**: images generated using all informative points and the odd sampling points of uninformative portions aka .
- **5th sequence sampling**: images generated using all the informative points and the (0,5,10, 15 ...nth) sampling points in uninformative portions.

Table 9.3: Performance analysis on information aware down sampling

Sampling method	Accuracy(%)	Processing time (ms)
Truncated sampling	94.21	0.04
Custom odd sampling	95.05	0.06
5th sequence sampling	98.34	0.05

To conduct the above mentioned experiments, analog voltage samples for 8 ECUs are collected using the experimental setup described in subsection A and images are generated using the truncated sampling, the custom odd sampling and the 5th sequence sampling methods. The images are

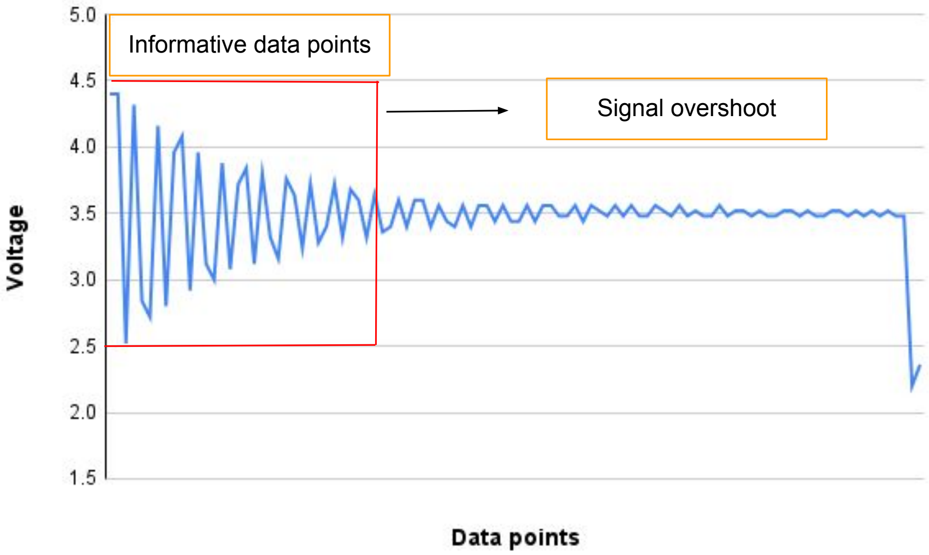


Figure 9.9: Information aware down sampling technique

then trained, tested and validated with the MobileNetV2 deep learning architecture and evaluated based on validation accuracy that is shown in Table 9.3. According to the analysis, 5th sequence sampling achieved a validation accuracy of 97.93% which is better than the accuracy achieved while using truncated sampling and custom odd sampling. Where the truncated sampling is faster than the other two approaches in data processing by at least .01 ms.

9.3.6 Spoof detection on vehicle test bench

To evaluate the performance on an actual vehicle test bench, an experiment is conducted where the goal is to detect spoofing attack using the proposed methodology. First an experiment is setup on a laboratory vehicle test bench (Model: GM Sierra, Year: 2020) shown as Figure 9.10 to perform spoofing attack by changing the vehicle gear from park to drive mode using a raspberry pi 4 model B, where analog voltage data is captured using a picoscope 2205 A with a sampling rate of 25 MS/s. After that, analog voltage data is captured again with the same sampling rate of 25 MS/s when the vehicle is put to drive mode from park mode using the vehicle gearshift. Although the attacker ECU was sending the same data we can see form Figure 9.11 the data send by the attacker has different analog voltage profile than the authorized ECU. The Figure 9.11 shows that the two sets of benign CAN-H signal data from authorized ECU, marked in blue & orange and captured at different times, differ from the two sets of CAN-H data from the attacker, marked in green & purple and captured at different times. However, when sent from the same CAN ECU, the data from both

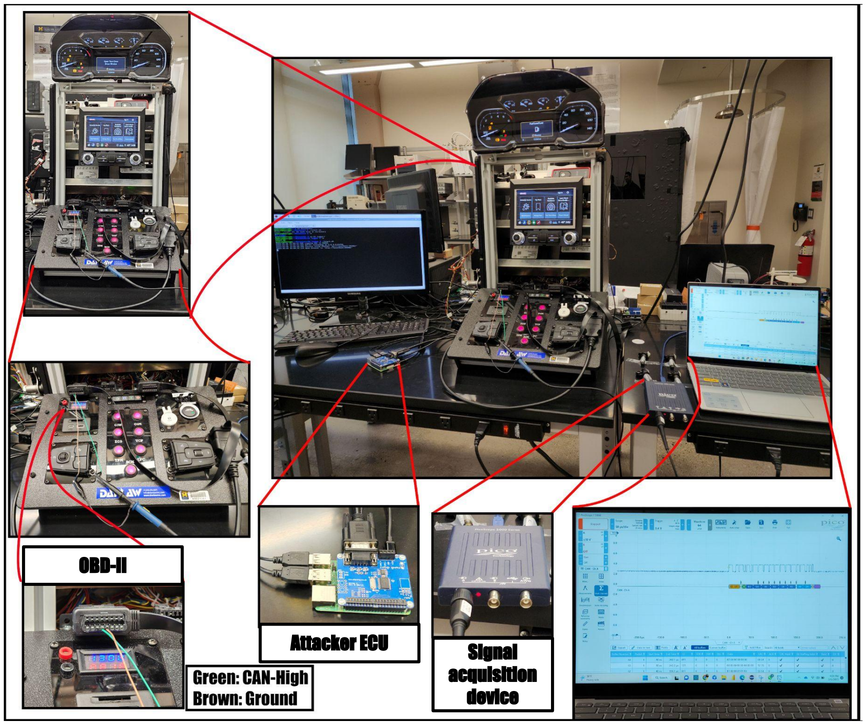


Figure 9.10: Spoofing attack on vehicle test bench

the authorized ECU and the attacker appear almost identical. It is clear form the figure that, the ECUs has their own fingerprint in their CAN-H dominant bit analog data which is different from 3.5v (marked as red in Figure 9.11).

Table 9.4: Performance under spoofing attack on laboratory vehicle bench

Data Sample	**Accuracy(%)**	**Precision(%)**	**Recall(%)**	**F1 score(%)**
No attack	97.78	100	97.78	98.87
Spoofing attack	100	96.72	100	98.33
Combined (spoof attack + no attack)	98.89	98.36	98.89	98.60

The analog voltages then prepossessed in a computer with 8 GM RAM and 416 images are generated from the distortions for both ECUs using Python programming language, where 240 were from authorized ECU and 176 were from attacker ECU. As the dataset is comparatively small we did data augmentation technique to flip each images from left to right and prepared a data size of 832 images. The images are then retrained using a pre-trained MobileNetV2 deep learning architecture where 70% data are used as training, 15% for tuning the model and 15% for testing the trained model. The performance of the proposed methodology is evaluated against metrics such

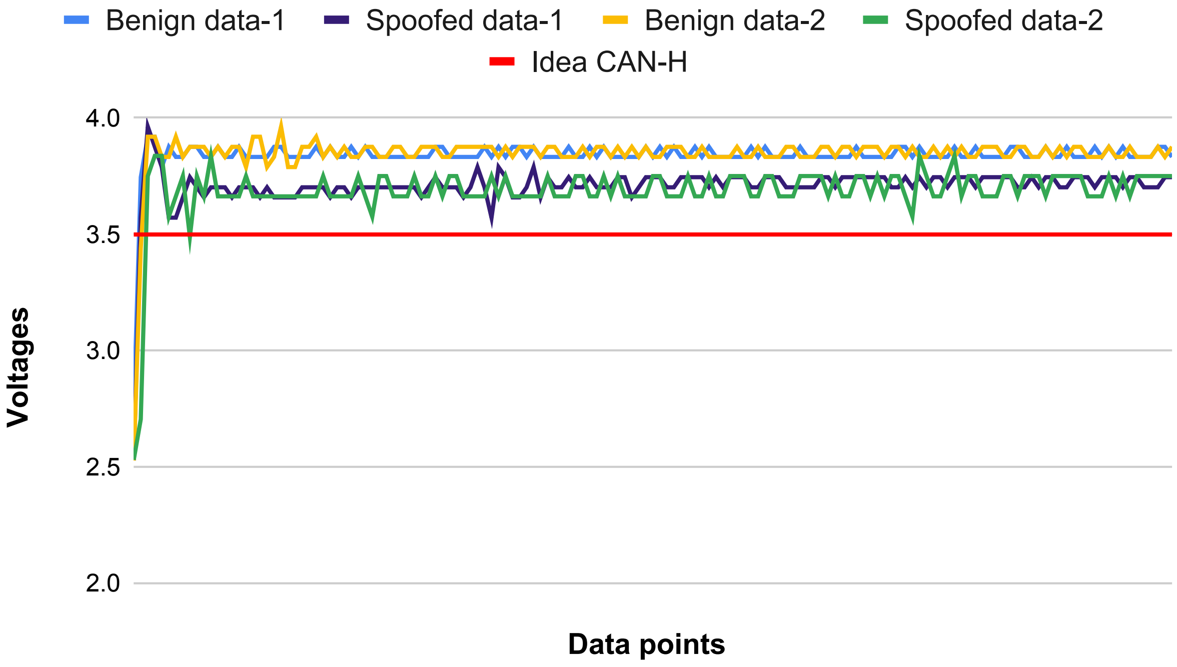


Figure 9.11: Analog voltage data for an authorized ECU and an attacker ECU

as (1) attack detection accuracy, precision, recall, f1-score and (2) benign data detection accuracy, precision, recall and f1-score. The simulation result is summarized in Table 9.4 and it shows that the proposed methodology detects spoofing attack with 100% accuracy while it achieves a precision of 96.72%, recall of 100% and f1-score of 98.33%. On the other hand, it detects benign data with an accuracy of 97.78%, precision of 100%, recall of 100% and f1-score of 98.33%. So, finally, it can be concluded that, the proposed methodology is efficient and applicable to today's vehicles as it achieved an combined accuracy of 98.89% in detecting spoofed and benign data in the vehicle test bench.

Table 9.5: Comparison with the state-of-the-art (6; 7)

Approach	**Accuracy(%)**	**Precision (%))**	**Recall(%)**	**F1 (%))**	**time (ms)**
(6)	97.13	97.00	97.00	95.75	1.35
(7)	89.24	89.63	89.63	89.38	0.95
KNN	94.97	95.00	95.38	95.13	238
SVM	95.82	95.75	96.13	95.75	238
Proposed IDS	98.34	98.63	98.25	98.38	0.05

9.3.7 Performance comparison with the state-of-the-art

Finally the proposed methodology is compared against the state-of-the-art sender identification methods (6) and (7) where (6) trains a Neural Network and (7) builds a Support Vector Machine (SVM) model using statistical features. In order to do so, the state-of-the-art (6) methodology extracts the distortion of the ECUs at first and handcrafts 11 statistical analysis based features including 6 time domain features i.e. maximum, minimum, mean, variance, skewness, kurtosis and 5 frequency domain features i.e. spectral standard deviation, spectral kurtosis, spectral skewness, spectral centroid, irregularity k to feed into an artificial neural network to evaluate the performance of their approach. While simulating their approach with data gathered from 8 ECU using the experimental setup described in subsection *A*, an validation accuracy of 97.13% as achieved while the proposed framework achieved 98.34%. But in terms of data processing (feature engineering), the state-of-the-art takes 27 times more than the proposed framework. The proposed methodology creates the a single recurrence plot for testing the model in 0.05 ms time using the 5th sequence information aware down sampling while, the state-of-the-art (6) generates 11 statistical features in 1.35 ms for identifying a single CAN sender which is computationally more expensive (see Table 9.5). Again the proposed approach is compared against (7) where the authors use signal characteristics by extracting 12 features such as maximum, mean, variance, skewness, kurtosis, centroid, flatness, power, irregularity, plateau, max plateau ratio and overshoot height to train machine learning model. While training & testing a SVM model, the state-of-the-art degrades by 9.1% in terms of sender identification accuracy than the proposed methodology and needs 0.95 ms to process features which is 19 times slower than the proposed approach. In addition to that, the proposed approach is compared with traditional machine learning based approaches i.e. K-Nearest Neighbors(KNN) and SVM where recurrence quantification parameters are used as features to fit machine learning models. Based on the data collected form 8 CAN ECUs, KNN achieved an accuracy of 94.97% and SVM achieved an accuracy of 95.82% while the feature generation took around 238 ms. It clearly shows that the proposed methodology is better both in terms of accuracy of identifying senders and feature processing time.

9.3.8 Discussion

Like other physical fingerprinting based approaches (6; 7; 167; 103), the proposed methodology is also sensitive to environmental effects such as temperature, aging, moisture, etc. (175) as electrical signal characteristics exhibit variations during car operation. According to (91), these deviations are monotonic and non-uniform, so it is hard to predict if not impossible. The effective solution in this scenario is to re-train the model to capture the updated deviations of this signal parameters (101; 177). Our experimental analysis (subsection *D* under section V) also proves that the effect of environmental conditions can be handled by updating the fingerprinting through re-training the deep learning model. Additionally, in terms of the attacks that are fed via the gateway of the vehicle, can remain undetected as the unique characteristics of attacker can be overridden by the unique characteristics of the gateway (7). Additional security measures like behavioral fingerprinting based approaches like (178; 179) are recommended to safeguard in the scenario.

CHAPTER 10

Conclusions

In conclusion, this dissertation successfully addresses the current state-of-the-art IVN message injection attacks and highlights that developing robust and reliable solutions to identify, localize, and mitigate cybersecurity threats to CAVs is of societal importance. As a gist, it successfully mitigates cyberattack in a fashion by proposing a multilayered framework that is reliable and computationally efficient. The performance of this framework is validated against a series of experiments on public dataset and collected dataset inside the UM-Dearborn ISSF laboratory. This has translated into a series of 7 academic publications and 1 under review article respectively[as of july 2023].

The first layer of the proposed framework aims to protect in-vehicle networks by developing real-time message authentication, intrusion detection, and localization based on unclonable signal attributes for physical fingerprinting of electronic control units (ECUs). The approach exploits uniqueness in physical signal attributes, leverages statistical signal processing and parameter modeling techniques for physical fingerprint estimation, and uses statistical machine learning methods for transmitting ECU identification and localization. The second layer aims to protect in-vehicle networks against firmware/software-level attacks using ECU behavioral fingerprinting through data-driven statistical graph analytics. The approach targeted by the research team here is the transformation of sequential in-vehicle network data into a directed-graph to leverage statistical graph analytics for ECU behavior modeling and intrusion detection.

The proposed physical fingerprinting framework solves the CAN protocol's inability to identify the sender by modeling the problem as an image classification problem. It introduces a novel approach for creating images utilizing uniqueness of the analog signal of CAN senders and it classifies the images using deep learning model to identify CAN ECUs. As, the proposed method only requires to generate an image for the identification of ECUs, it can be a better alternative than the handcrafted feature engineering process in CAN physical fingerprinting. As per contributing to the state-of-the-art, to the best of our knowledge the proposed methodology is the first ever work that utilizes the concept of computer vision in CAN sender identification problem. The experimental result shows that it is effective as it achieved an accuracy of 100% in modern vehicle test bench and efficient as it only requires an image to identify sender. The potential of this proposed methodology is beyond CAN because the methodology is not protocol dependent. Hence, it has a potential to identify sender in in-vehicle networking protocols such as FlexRay, Ethernet, MOST, etc. In the future, we would like to investigate, the effectiveness of the methodology in other in-vehicle networking protocols and would evaluate the performance of other deep learning architectures than MobileNetV2 & EfficientNet. The need for a computationally efficient IDS for modern vehicles is becoming one of the most essential security components as these vehicles are exposed to a huge number of threats and this work opens a new dimension of mitigating cyberattack in real time.

The behavioral fingerprinting framework models the graph theory to detect IVN intrusion by generalizing the network traffic. The proposed framework examines the performance of the IDS by focusing on two different categories, (i) based on detection algorithm, (ii) based on graph features. The effectiveness of the idea of learning natural behavior and deviation of nature of IVN networks is evaluated by statistical approach, general machine learning approach and a novel ensemble based approach. To check the effectiveness of graph theory we analyzed the characteristics of all kinds of CAN monitoring-based attacks and proposed a four-stage IDS with the help of graph theory, statistical analysis, and the chi-square method.To the best of our knowledge, this is the first graph based IDS for CAN bus communication. Our experimental results show that we have a very low misclassification rate in detecting attacks or attack free data. Moreover, a CAN intrusion detection system is proposed that uses graph based features to detect CAN attack using traditional machine learning algorithms (SVM and KNN). This novel and pragmatic approach uses graph based features to classify authentic and malicious CAN messages for in-vehicle communication. The experimental results showed that using graph-based features, an accuracy of 97.92% & 97.99% was achieved using SVM & KNN algorithms respectively. In the course of time, we would like to consider other graph properties to detect intrusion for in-vehicle communication. In addition, we will apply different machine learning algorithms in place of the SVM & KNN to see the robustness of the selected features.

The ensemble based novel approach a lightweight IDS is introduced for the CAN bus which can detect message injection attacks on the CAN bus based on the analysis of the graph neighborhood similarity. I developed a novel algorithm to detect anomalies on transmitted CAN bus by comparing any single block of CAN messages with blocks of attack-free CAN messages. I evaluated our approach against three types of message injection attacks which are spoofing, fuzzy and DoS attacks on real vehicular CAN bus data. The results demonstrated that the proposed IDS can successfully detect spoofing attacks, fuzzy attacks and a DoS attack which can be the most perilous attack for vehicles in minimal time with an accuracy of 96.01%. I believe that our IDS contributes to improving vehicle security without making any change in the CAN protocol. As a future work, I plan to evaluate our system against other attacks. Finally, to confirm practicality, an empirical experiment needs to be carried out to test our approach under various real-world scenarios.

However, the behavioral fingerprinting approach has a limitation and it needs to be taken into account while interpreting the finding of the study. That is the behavioral approaches proposed in the methodology is protocol dependent. In this layer, the foundation of the proposed methodologies are graph theory and the constructed graphs considers the CAN arbitration ID as nodes, so, the proposed methodologies become specific to CAN network. While applying the behavioral approaches to other IVN networks like Automotive Ethernet, LIN, MOST, etc. need additional work. Which means it is essential to find relations between subsequent IVN messages to construct graphs. On the other hand, the physical fingerprinting approach is protocol independent because the underlying kernel relies on distortions in voltages. In the future, a series of experiments needs to be carried out to establish the connection of the number for ECUs over the sender identification accuracy to evaluate the effectiveness of the proposed physical fingerprinting approach.

In summary, the thesis emphasizes the need to prioritize the security of the entire automotive ecosystem to effectively safeguard modern cars. In addition to protecting IVNs, a focus on sensor level security is necessary to guarantee the accuracy and dependability of data gathered by numerous sensors positioned throughout the vehicle. Wireless security is equally crucial, particularly in light of the development of Vehicle-to-Everything (V2X) communication, which allows vehicles to communicate with other vehicles, infrastructure, and people. In addition to defending drivers and passengers from potential dangers, strengthening cybersecurity measures at these crucial levels will also increase public confidence in connected and autonomous vehicles, paving the road for a safer and more secure transportation future.

APPENDIX A

Glossary

This dissertation uses concepts from Computer Network, Machine-Learning and Cyber-Security. Next, relevant terminology are defined for better reading.

A.1 Graph-Theory

- **Assortativity** - a preference for a network's nodes to attach to others that are similar in some way.
- **Density** - the ratio between the number of edges in a graph and the maximum number of edges that the graph can contain.
- **Diameter** - The length of the shortest path between the most distanced nodes.
- **Edge** - A notion that connects multiple nodes in a graph.
- **Graph** - A non linear data structure that is constructed by nodes and edges.
- **Node** - A point that defines a graph.
- **Radius** - the minimum graph eccentricity of any graph vertex in a graph.
- **Reciprocity** - a measure of the likelihood of vertices in a directed network to be mutually linked.

A.2 Machine-Learning

- **Classification** - Process of determining the class of data that has certain pattern.
- **Class/label** - A discrete value that data with certain pattern can be classified with.

- **Convolutional-Neural-Network** - A deep learning model for processing grid pattern data such as image.

- **Deep-Learning** - A subset of machine learning and a type of convolutional neural network that has more than three layers.

- **Ensemble Learning** - Refers to the algorithm that utilizes decision of two or more algorithm to predict.

- **Feature** - The characteristics that can define a class.

- **K-Nearest Neighbors** - Supervised machine learning algorithm that can be used in classification and regression problem.

- **Support-Vector-Machine** - Supervised machine learning algorithm that can be used in classification and regression problem.

A.3 Cyber-Security

- **Availability** - Policy that ensures resources are available consistently only to authorized entities.

- **Confidentiality** - Refers set of rules to protecting information from unauthorized access.

- **Integrity** - Refers to set of rules that ensure information is not tempered while traveling from source to destination.

- **Intrusion-detection-system** - A system that monitors network traffic to detect suspicious behavior of a network.

BIBLIOGRAPHY

[1] "Upstream security's 2021 global automotive cybersecurity report," https://upstream.auto/upstream-security-global-automotive-cybersecurity-report-2021/, accessed: 2021-10-25.

[2] A. Alshammari, M. A. Zohdy, D. Debnath, and G. Corser, "Classification approach for intrusion detection in vehicle systems," *Wireless Engineering and Technology*, vol. 9, no. 4, pp. 79–94, 2018.

[3] R. Islam, R. U. D. Refat, S. M. Yerram, and H. Malik, "Graph-based intrusion detection system for controller area networks," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[4] D. Yang, K. Jiang, D. Zhao, C. Yu, Z. Cao, S. Xie, Z. Xiao, X. Jiao, S. Wang, and K. Zhang, "Intelligent and connected vehicles: Current status and future perspectives," *Science China Technological Sciences*, vol. 61, pp. 1446–1471, 2018.

[5] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1577–1583.

[6] O. Avatefipour, A. Hafeez, M. Tayyab, and H. . Malik, "December)," *Linking received packet to the transmitter through physical-fingerprinting of controller area network*, pp. 1–6, 2017.

[7] M. Kneib, O. Schell, and C. . Huth, *February)*. EASI: Edge-Based Sender Identification on Resource-Constrained Platforms for Automotive Networks. In NDSS.

[8] J. Brockman, *The Greatest Inventions of the Past 2,000 Years*. Simon and Schuster, 2000.

[9] "Number of u.s. automotive industry employees between june 2010 and 2021, by sector," https://www.statista.com/statistics/276474/automotive-industry-employees-in-the-united-states-by-sector/, accessed: 2021-10-25.

[10] "Us auto industry sales analysis," https://www.goodcarbadcar.net/usa-auto-industry-total-sales-figures/, accessed: 2021-10-25.

[11] F. Hatani, "Artificial intelligence in japan: Policy, prospects, and obstacles in the automotive industry," in *Transforming Japanese Business*. Springer, 2020, pp. 211–226.

[12] J. N. Bajpai, “Emerging vehicle technologies & the search for urban mobility solutions,” *Urban, Planning and Transport Research*, vol. 4, no. 1, pp. 83–100, 2016.

[13] D. Elliott, W. Keen, and L. Miao, “Recent advances in connected and automated vehicles,” *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 6, no. 2, pp. 109–131, 2019.

[14] R. N. Charette, “This car runs on code,” *IEEE spectrum*, vol. 46, no. 3, p. 3, 2009.

[15] “Top 25 auto cybersecurity hacks: Too many glass houses to be throwing stones,” https://www.forbes.com/sites/stevetengler/2020/06/30/top-25-auto-cybersecurity-hacks-too-many-glass-houses-to-be-throwing-stones/?sh=21a2ac817f65, accessed: 2021-10-25.

[16] A. Greenberg, “Hackers remotely kill a jeep on the highway—with me in it,” *Wired*, vol. 7, p. 21, 2015.

[17] S. Nie, L. Liu, and Y. Du, “Free-fall: Hacking tesla from wireless to can bus,” *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[18] A. Hafeez, K. Topolovec, C. Zolo, and W. Sarwar, “State of the art survey on comparison of can, flexray, lin protocol and simulation of lin protocol,” 2020.

[19] A. Talic, “Security analysis of ethernet in cars,” 2017.

[20] S. Jadhav and D. Kshirsagar, “A survey on security in automotive networks,” in *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018, pp. 1–6.

[21] H. Muyshondt and C. Sanchez, “Most–the audio/video backbone of the car,” in *Audio Engineering Society Conference: 36th International Conference: Automotive Audio*. Audio Engineering Society, 2009.

[22] K. Johansson, M. T”orngren, and L. Nielsen, “Vehicle applications of controller area network,” *Handbook Of Networked And Embedded Control Systems*, pp. 741–765, 2005.

[23] R. Islam and R. U. D. Refat, “Improving can bus security by assigning dynamic arbitration ids,” *Journal of Transportation Security*, vol. 13, no. 1, pp. 19–31, 2020.

[24] [Online]. Available: https://support.ixiacom.com/sites/default/files/resources/whitepaper/ixia-automotive-ethernet-primer-whitepaper_1.pdf

[25] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, “Comprehensive experimental analyses of automotive attack surfaces.” in *USENIX Security Symposium*, vol. 4, no. 447-462. San Francisco, 2011, p. 2021.

[26] I. Rouf, R. D. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, “Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study.” in *USENIX Security Symposium*, vol. 10, 2010.

[27] O. Nakhila, E. Dondyk, M. F. Amjad, and C. Zou, “User-side wi-fi evil twin attack detection using ssl/tcp protocols,” in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2015, pp. 239–244.

[28] M. Vanhoef and F. Piessens, “Denial-of-service attacks against the 4-way wi-fi handshake,” in *Proceedings of the 9th International Conference on Network and Communications Security*. Academy & Industry Research Collaboration Center (AIRCC), 2017.

[29] M. H. Eiza, Q. Ni, and Q. Shi, “Secure and privacy-aware cloud-assisted video reporting service in 5g-enabled vehicular networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 10, pp. 7868–7881, 2016.

[30] I. Ivanov, C. Maple, T. Watson, and S. Lee, “Cyber security standards and issues in v2x communications for internet of vehicles,” 2018.

[31] J. Cichonski, J. Franklin, and M. Bartock, “Guide to lte security,” National Institute of Standards and Technology, Tech. Rep., 2016.

[32] M. Muhammad and G. A. Safdar, “Survey on existing authentication issues for cellular-assisted v2x communication,” *Vehicular Communications*, vol. 12, pp. 50–65, 2018.

[33] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, p. 91, 2015.

[34] Autoblog, “Vehicle headlight can bus injection theft method update,” https://www.autoblog.com/2023/04/18/vehicle-headlight-can-bus-injection-theft-method-update/, April 18 2023.

[35] ZDNet, “Info of 27.7 million texas drivers exposed in vertafore data breach,” https://www.zdnet.com/article/info-of-27-7-million-texas-drivers-exposed-in-vertafore-data-breach, 2023.

[36] M. Wolf, R. Lambert, T. Enderle, and A. Schmidt, “Wanna drive? feasible attack paths and effective protection against ransomware in modern vehicles,” in *Proc. Embedded Security in Cars Conference (escar) Europe*, 2017.

[37] “The McAfee Security Report,” https://www.mcafee.com/blogs/other-blogs/mcafee-labs/todays-connected-cars-vulnerable-hacking-malware/.

[38] “Boston 25 News Report,” https://www.boston25news.com/news/discovery-of-hidden-gps-tracker-leads-to-mass-supreme-court-case/742286360/.

[39] Y. Wiseman, “Vehicle identification by ocr, rfid and bluetooth for toll roads,” *International Journal of Control and Automation*, vol. 11, no. 9, pp. 67–76, 2018.

[40] J. M. Ernst and A. J. Michaels, “Lin bus security analysis,” in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 2085–2090.

[41] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 528–533.

[42] D. K. Nilsson and U. E. Larson, "Simulated attacks on can buses: vehicle virus," in *IASTED International conference on communication systems and networks (AsiaCSN)*, 2008, pp. 66–72.

[43] D. Püllen, N. A. Anagnostopoulos, T. Arul, and S. Katzenbeisser, "Security and safety co-engineering of the flexray bus in vehicular networks," in *Proceedings of the International Conference on Omni-Layer Intelligent Systems*, 2019, pp. 31–37.

[44] E. Seo, H. M. Song, and H. K. Kim, "Gids: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2018, pp. 1–6.

[45] A. Hafeez, K. Topolovec, and S. Awad, "Ecu fingerprinting through parametric signal modeling and artificial neural networks for in-vehicle security against spoofing attacks," in *2019 15th International Computer Engineering Conference (ICENCO)*. IEEE, 2019, pp. 29–38.

[46] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2017, pp. 57–5709.

[47] S. Woo, H. J. Jo, I. S. Kim, and D. H. Lee, "A practical security architecture for in-vehicle can-fd," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2248–2261, 2016.

[48] Y. Huo, W. Tu, Z. Sheng, and V. C. Leung, "A survey of in-vehicle communications: Requirements, solutions and opportunities in iot," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 132–137.

[49] S. M. Bellovin, "Security problems in the tcp/ip protocol suite," *ACM SIGCOMM Computer Communication Review*, vol. 19, no. 2, pp. 32–48, 1989.

[50] 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 Technical Paper, Tech. Rep., 2017.

[51] S. C. HPL, "Introduction to the controller area network (can)," *Application Report SLOA101*, pp. 1–17, 2002.

[52] M. Alves, M. Pereira, and H. G. Ramos, "Can protocol: A laboratory prototype for field bus applications," in *XIX IMEKO World Congress, Fundamental and Applied Metrology*. Citeseer, 2009, pp. 454–457.

[53] 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)*. IEEE, 2017, pp. 1–6.

[54] A. Hafeez, S. C. Ponnapali, and H. Malik, “Exploiting channel distortion for transmitter identification for in-vehicle network security,” *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 3, no. 11-02-02-0005, pp. 5–17, 2020.

[55] 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 (ICENCO)*. IEEE, 2018, pp. 79–83.

[56] M. Tayyab, A. Hafeez, and H. Malik, “Spoofing attack on clock based intrusion detection system in controller area networks,” in *NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, Michigan*, 2018, pp. 1–13.

[57] A. Hafeez, “A robust, reliable and deployable framework for in-vehicle security,” Ph.D. dissertation, 2020.

[58] A. Hafeez, K. Rehman, and H. Malik. State of the Art Survey on Comparison of Physical Fingerprinting-Based Intrusion Detection Techniques for In-Vehicle Security, 2020.

[59] J. Jung, K. Park, and J.-S. Cha, “Implementation of a network-based distributed system using the can protocol,” in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2005, pp. 1104–1110.

[60] I. Broster and A. Burns, “An analysable bus-guardian for event-triggered communication,” in *RTSS 2003. 24th IEEE Real-Time Systems Symposium, 2003*. IEEE, 2003, pp. 410–419.

[61] M. Di Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and using the controller area network communication protocol: theory and practice*. Springer Science & Business Media, 2012.

[62] H. M. Song, H. R. Kim, and H. K. Kim, “Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network,” in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[63] M. Müter, A. Groll, and F. C. Freiling, “A structured approach to anomaly detection for in-vehicle networks,” in *2010 Sixth International Conference on Information Assurance and Security*. IEEE, 2010, pp. 92–98.

[64] M. Müter and N. Asaj, “Entropy-based anomaly detection for in-vehicle networks,” in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 1110–1115.

[65] H. Li, L. Zhao, M. Juliato, S. Ahmed, M. R. Sastry, and L. L. Yang, “Poster: Intrusion detection system for in-vehicle networks using sensor correlation and integration,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2531–2533.

[66] D. Tian, Y. Li, Y. Wang, X. Duan, C. Wang, W. Wang, R. Hui, and P. Guo, “An intrusion detection system based on machine learning for can-bus,” in *International Conference on Industrial Networks and Intelligent Systems*. Springer, 2017, pp. 285–294.

[67] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[68] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2016, pp. 130–139.

[69] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, 2018.

[70] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection." in *USENIX Security Symposium*, 2016, pp. 911–927.

[71] S. U. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran, "Cloaking the clock: emulating clock skew in controller area networks," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2018, pp. 32–42.

[72] H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka, and H. Imai, "New attestation based security architecture for in-vehicle communication," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*. IEEE, 2008, pp. 1–6.

[73] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2034–2042, 2013.

[74] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (can) communication protocol," in *2012 International Conference on Cyber Security*. IEEE, 2012, pp. 1–7.

[75] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *ECRYPT Workshop on Lightweight Cryptography*, vol. 2011, 2011.

[76] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," *Knowledge-based Systems*, vol. 123, pp. 163–173, 2017.

[77] V. Barletta, D. Caivano, A. Nannavecchia, and M. Scalera, "Intrusion detection for in-vehicle communication networks: An unsupervised kohonen som approach," *Future Internet*, vol. 12, p. 119, 2020.

[78] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown can bus networks," *Vehicular Communications*, vol. 9, pp. 43–52, 2017.

[79] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, 2015, pp. 45–49.

[80] O. Minawi, J. Whelan, A. Almehmadi, and E. khatib K. (202). Machine Learning-Based Intrusion Detection System for Controller Area Networks.

[81] F. Martinelli, F. Mercaldo, V. Nardone, and A. Santone. Car hacking identification through fuzzy logic algorithms, 2017.

[82] O. Avatefipour, A. Al-sumaiti, A. El-sherbeeny, E. Awwad, M. Elmeligy, M. Mohamed, and H. Malik, "An intelligent secured framework for cyberattack detection in electric vehicles' can bus using machine learning," *Ieee Access*, vol. 7, no. 12, pp. 27 580–12 759, 2019.

[83] L. Yang, A. Moubayed, I. Hamieh, and A. Shami. Tree-based intelligent intrusion detection system in internet of vehicles, 2019.

[84] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "Canet: An unsupervised intrusion detection system for high dimensional can bus data," *Ieee Access*, vol. 8, pp. 58 194–58 205, 2020.

[85] M. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "Lstm-based intrusion detection system for in-vehicle can bus communications," *Ieee Access*, vol. 8, no. 12, pp. 85 489–18 550, 2020.

[86] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *Ieee Access*, vol. 6, pp. 3491–3508, 2017.

[87] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020.

[88] L. Zhang, L. Shi, N. Kaja, and D. Ma, "A two-stage deep learning approach for can intrusion detection," in *Proc. Ground Vehicle Syst. Eng. Technol. Symp.(GVSETS)*, 2018, pp. 1–11.

[89] A. A. Elkhail, R. U. D. Refat, R. Habre, A. Hafeez, A. Bacha, and H. Malik, "Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses," *IEEE Access*, vol. 9, no. 17, pp. 62 401–16 243, 2021.

[90] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu, and Z. . Liu, "October)," *LEAP: A lightweight encryption and authentication protocol for in-vehicle communications*, pp. 1158–1164, 2019.

[91] L. Popa, B. Groza, C. Jichici, and P. S. Murvay, "Ecuprint—physical fingerprinting electronic control units on can buses inside cars and sae j1939 compliant vehicles," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1185–1200, 2022.

[92] J. Schmandt, A. T. Sherman, and N. Banerjee, "Mini-mac: Raising the bar for vehicular security with a lightweight message authentication protocol," *Vehicular Communications*, vol. 9, pp. 188–196, 2017.

[93] C. R. O. Hartkopp and R. Schilling, "nd." macan-message authenticated can," in *Proc. 10th Int*, C. E. S. in Cars (escar), Ed.

[94] A. Hazem and H. A. . Fahmy, "November)," *Lcap-a lightweight can authentication protocol for securing in-vehicle networks*, vol. 10.

[95] A. Van Herrewege, D. Singelee, and I. . Verbauwhede, “November),” *CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus*, vol. 20, 2011.

[96] B. Groza, S. Murvay, A. V. Herrewege, and I. . Verbauwhede, “December),” in *LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks. In International Conference on Cryptology and Network Security . , , Heidelberg*. Berlin: Springer, pp. 185–200.

[97] K. G. Shin and K.-T. Cho, “Fingerprinting electronic control units for vehicle intrusion detection,” Jun. 22 2021, uS Patent 11,044,260.

[98] S. Bellaire, M. Bayer, A. Hafeez, R. U. D. Refat, and H. Malik, “Fingerprinting ecus to implement vehicular security for passenger safety using machine learning techniques,” in *Proceedings of SAI Intelligent Systems Conference . , Cham*, 2023, pp. 16–32.

[99] R. M. K. Gerdes, *Physical layer identification: methodology, security, and origin of variation*. Iowa State University, 2011.

[100] M. Tayyab, A. Hafeez, and H. Malik, “Spoofing attack on clock based intrusion detection system in controller area networks,” 2018.

[101] M. Kneib and C. . Huth, “October),” in *Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks*, I. Proceedings, Ed. of theACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 787–800.

[102] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, “Voltageids: Low-level communication characteristics for automotive intrusion detection system,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, 2018.

[103] K. Verma, M. Girdhar, A. Hafeez, and S. S. . Awad, “December). ecu identification using neural network classification and hyperparameter tuning,” *InIEEE International Workshop on Information Forensics and Security (WIFS) (p*, pp. 1–6, 2022.

[104] S. Ahmed, M. Juliato, C. Gutierrez, and M. Sastry, “Two-point voltage fingerprinting: Increasing detectability of ecu masquerading attacks. arxiv,” 2021, preprint.

[105] B. C. H. A. a. Payne, “and exploiting the can bus protocol,” *Journal Of Cybersecurity Education, Research And Practice*, vol. 2019, p. 5, 2019.

[106] A. Saber and D. Troia, “F. & stamp, m. intrusion detection and can vehicle networks,” *Digital Forensic Investigation Of Internet Of Things (IoT) Devices*, pp. 125–154, 2021.

[107] C. Schappin, *Intrusion Detection on the Automotive CAN bus*. Technische Universiteit Eindhoven, 2017.

[108] Y. Yang, Z. Duan, and M. Tehranipoor, “Identify a spoofing attack on an in-vehicle can bus based on the deep features of an ecu fingerprint signal,” *Smart Cities*, vol. 3, pp. 17–30, 2020.

[109] M. Kneib, O. Schell, and C. Huth, "Easi: Edge-based sender identification on resource-constrained platforms for automotive networks." in *NDSS*, 2020.

[110] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ecus using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[111] M. Foruhandeh, Y. Man, R. Gerdes, M. Li, and T. S. Chantem, "Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks," *Proceedings Of The 35th Annual Computer Security Applications Conference*, pp. 229–244, 2019.

[112] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2018, pp. 787–800.

[113] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con*, vol. 21, pp. 260–264, 2013.

[114] M. Hanselmann, T. Strauss, K. Dormann, and H. C. Ulmer, "An unsupervised intrusion detection system for high dimensional can bus data," *IEEE Access*, vol. 8, pp. 58 194–58 205, 2020.

[115] D. Tian, Y. Li, Y. Wang, X. Duan, C. Wang, W. Wang, R. Hui, and P. Guo, "An intrusion detection system based on machine learning for can-bus," *Industrial Networks And Intelligent Systems*, pp. 285–294, 2018.

[116] R. Refat and A. Elkhail, *Hafeez , A. & Malik, H. Detecting CAN bus intrusion by applying machine learning method to graph based features*. Proceedings Of SAI Intelligent Systems Conference, 2021.

[117] D. Yogatama and G. Mann, "Efficient transfer learning method for automatic hyperparameter tuning," *Artificial Intelligence And Statistics*, pp. 1077–1085, 2014.

[118] M. Feurer and F. H. o. Hutter, "Automated machine learning," *pp*, pp. 3–33, 2019.

[119] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," *IEEE Internet Of Things Journal*, vol. 1, pp. 10–21, 2014.

[120] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 49, pp. 1–39, 2016.

[121] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: a survey," *Data mining and knowledge discovery*, vol. 29, no. 3, pp. 626–688, 2015.

[122] P. Mulinka and P. Casas, "Stream-based machine learning for network security and anomaly detection," in *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, 2018, pp. 1–7.

[123] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE symposium on security and privacy*. IEEE, 2010, pp. 305–316.

[124] M. Furdek, C. Natalino, F. Lipp, D. Hock, A. Di Giglio, and M. Schiano, "Machine learning for optical network security monitoring: A practical perspective," *Journal of Lightwave Technology*, vol. 38, no. 11, pp. 2860–2871, 2020.

[125] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks–practical examples and selected short-term countermeasures," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2008, pp. 235–248.

[126] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on intelligent transportation systems*, vol. 16, no. 2, pp. 993–1006, 2014.

[127] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 631–636.

[128] K. Iehira, H. Inoue, and K. Ishida, "Spoofing attack using bus-off attacks against a specific ecu of the can bus," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–4.

[129] A. Tomlinson, J. Bryans, and S. A. Shaikh, "Towards viable intrusion detection methods for the automotive controller area network," in *2nd ACM Computer Science in Cars Symposium*, 2018.

[130] A. Ugoni and B. F. Walker, "The chi square test: an introduction," *COMSIG review*, vol. 4, no. 3, p. 61, 1995.

[131] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median," *Journal of Experimental Social Psychology*, vol. 49, no. 4, pp. 764–766, 2013.

[132] J. Devore, "Making sense of data: A practical guide to exploratory data analysis and data mining," 2007.

[133] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, "Security authentication system for in-vehicle network," *SEI Technical Review*, vol. 81, pp. 5–9, 2015.

[134] W. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24, pp. 1565–1567, 2006.

[135] P. L. . K. nearest neighbor, "Scholarpedia 4:1883."

[136] G. Ducoffe and F. Dragan, *A story of diameter, radius, and (almost) Helly property*. Networks, 2020.

[137] D. Eppstein, “Diameter and treewidth in minor-closed graph families,” *Algorithmica*, vol. 27, pp. 275–291, 2000.

[138] Ł. Kowalik, “Approximation scheme for lowest outdegree orientation and graph density measures,” in *Algorithms and Computation: 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006. Proceedings 17.* Springer, 2006, pp. 557–566.

[139] J. Berg, J. Dickhaut, and K. Mccabe, “Trust, reciprocity, and social history,” *Games And Economic Behavior*, vol. 10, pp. 122–142, 1995.

[140] M. Newman, “Random graphs with clustering,” *Physical Review Letters*, vol. 103, p. 058701, 2009.

[141] R. Noldus and P. Vanmieghem, “Assortativity in complex networks,” *Journal Of Complex Networks*, vol. 3, pp. 507–542, 2015.

[142] S. Suthaharan, *Support vector machine.* Springe, 2016.

[143] F. Salo, M. Injadat, A. B. Nassif, A. Shami, and A. Essex, “Data mining techniques in intrusion detection systems: A systematic literature review,” *IEEE Access*, vol. 6, pp. 56 046–56 058, 2018.

[144] M. Ring, D. Frkat, and M. Schmiedecker, “Cybersecurity evaluation of automotive e/e architectures,” in *ACM Computer Science In Cars Symposium (CSCS 2018)*, 2018.

[145] S. Kosub, “A note on the triangle inequality for the jaccard distance,” *Pattern Recognition Letters*, vol. 120, pp. 36–38, 2019.

[146] J. G. Foster, D. V. Foster, P. Grassberger, and M. Paczuski, “Edge direction and the structure of networks,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 24, pp. 10 815–10 820, 2010.

[147] A. Chapelle and A. Szafarz, “Controlling firms through the majority voting rule,” *Physica A: Statistical Mechanics and its Applications*, vol. 355, no. 2-4, pp. 509–529, 2005.

[148] S. Halder, M. Conti, and S. K. . Das, “January),” in *COIDS: A clock offset based intrusion detection system for controller area networks*, I. Proceedings, Ed. of the 21st International Conference on Distributed Computing and Networking, pp. 1–10.

[149] C. Jichici, B. Groza, R. Ragobete, P. S. Murvay, and T. Andreica, *Effective Intrusion Detection and Prevention for the Commercial Vehicle SAE J1939 CAN Bus.* IEEE Transactions on Intelligent Transportation Systems, 2022.

[150] X. Duan, H. Yan, D. Tian, J. Zhou, J. Su, and W. Hao, *In-vehicle CAN bus tampering attacks detection for connected and autonomous vehicles using an improved isolation forest method.* IEEE Transactions on Intelligent Transportation Systems, 2021.

[151] H. Sun, M. Chen, J. Weng, Z. Liu, and G. Geng, “Anomaly detection for in-vehicle network using cnn-lstm with attention mechanism,” *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 10 880–10 893, 2021.

[152] M. K. Ishak and F. K. Khan, “Unique message authentication security approach based controller area network (can) for anti-lock braking system (abs) in vehicle network,” *Procedia Computer Science*, vol. 160, pp. 93–100, 2019.

[153] S. Nurnberger and C. Rossow, “vatican-vetted authenticated can bus,” *Cryptographic Hardware and Embedded Systems:*, vol. 18, pp. 17–19, August 2016.

[154] K. D. Kang, Y. Baek, S. Lee, and S. H. . Son, “August). lightweight authentication method for controller area network,” in *InIEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA).* IEEE, 2016, pp. 101–101.

[155] Y. Weisglass and Y. Oren, *Authentication method for CAN messages.* ESCAR Europe, 2016.

[156] R. Islam, R. U. D. Refat, S. M. Yerram, and H. Malik, *Graph-based intrusion detection system for controller area networks.* IEEE Transactions on Intelligent Transportation Systems, 2020.

[157] H. Lee, S. H. Jeong, and H. K. . Kim, “August),” in *OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame.* Security and Trust (PST) . IEEE: In15th Annual Conference on Privacy, 2017, pp. 57–5709.

[158] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, “Lstm-based intrusion detection system for in-vehicle can bus communications,” *IEEE Access*, vol. 8, no. 12, pp. 85 489–18 550, 2020.

[159] A. Hafeez, K. Topolovec, and S. . Awad, “December),” *ECU fingerprinting through parametric signal modeling and artificial neural networks for in-vehicle security against spoofing attacks*, vol. 15, pp. 29–38, 2019.

[160] T. Xu, X. Lu, L. Xiao, Y. Tang, and H. . Dai, “May). voltage based authentication for controller area networks with reinforcement learning,” in *International Conference on Communications (ICC) (pp*, I. Ieee, Ed. 1-5). IEEE, pp. 2019–2019.

[161] A. Hafeez, S. C. Ponnapali, and H. Malik, “Exploiting channel distortion for transmitter identification for in-vehicle network security,” *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 3, no. 11, pp. 5–17, 2020.

[162] W. Jeong, S. Han, E. Choi, S. Lee, and J. W. Choi, “Cnn-based adaptive source node identifier for controller area network (can),” *IEEE Transactions on Vehicular technology*, vol. 69, no. 11, pp. 13 916–13 920, 2020.

[163] N. Marwan, J. Kurths, and P. Saparin, “Generalised recurrence plot analysis for spatial data,” *Physics Letters A*, vol. 360, no. 4-5, pp. 545–551, 2007.

[164] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. . Chen, “April),” *Mobilenetv*, vol. 2.

[165] B. Koonce, “Efficientnet,” *In Convolutional neural networks with swift for tensorflow (p*, pp. 109–123, Apr 2021.

[166] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, and H. Zhu, “... & he, q. (2020). a comprehensive survey on transfer learning,” in *Proceedings of the IEEE.* 109(1, pp. 43–76.

[167] Y. Yang, Z. Duan, and M. Tehranipoor, “Identify a spoofing attack on an in-vehicle can bus based on the deep features of an ecu fingerprint signal,” *Smart Cities*, vol. 3, no. 1, pp. 17–30, 2020.

[168] R. U. D. Refat, A. A. Elkhail, and H. Malik, “Machine learning for automotive cybersecurity: Challenges, opportunities and future directions,” *AI-enabled Technologies for Autonomous and Connected Vehicles*, pp. 547–567, 2022.

[169] S. Mathur, A. Reznik, C. Ye, R. Mukherjee, A. Rahman, and Y. Shah, “... & mandayam, n. (2010),” *Exploiting the physical layer for enhanced security [security and privacy in emerging wireless networks]. IEEE Wireless Communications*, vol. 17, no. 5, pp. 63–70.

[170] K. Dong, C. Zhou, Y. Ruan, and Y. . Li, “December),” in *Mobilenetv2 model for image classification.* In2nd International Conference on Information Technology and Computer Application (ITCA) . IEEE, 2020, pp. 476–480.

[171] T. Rawald, M. Sips, and N. Marwan, “Pyrqa—conducting recurrence quantification analysis on very long time series efficiently,” *Computers & Geosciences*, vol. 104, pp. 101–108, 2017.

[172] P. P. 8.0.0, “Feb 21,” vol. 2023, 2021. [Online]. Available: https://pypi.org/project/PyRQA/

[173] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. . Liu, “October). a survey on deep transfer learning,” in *International conference on artificial neural networks . , Cham*, pp. 270–279.

[174] Y. You, Z. Zhang, C. J. Hsieh, J. Demmel, and K. . Keutzer, “August). imagenet training in minutes,” in *Proceedings of the 47th International Conference on Parallel Processing*, pp. 1–10.

[175] A. Sierota and J. Rungis, “Electrical insulating oils. i. characterization and pre-treatment of new transformer oils,” *IEEE Electrical Insulation Magazine*, vol. 11, no. 1, pp. 8–20, 1995.

[176] S. Aja-Fernández and A. Tristán-Vega, “A review on statistical noise models for magnetic resonance imaging,” *LPI, ETSI Telecomunicacion, Universidad de Valladolid, Spain, Tech. Rep*, 2013.

[177] K. T. Cho and K. G. . Shin, “October),” in *Viden: Attacker identification on in-vehicle networks*, I. Proceedings, Ed. of theACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 1109–1123.

[178] R. U. D. Refat, A. A. Elkhail, A. Hafeez, and H. Malik, “Detecting can bus intrusion by applying machine learning method to graph based features,” in *of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3*, I. Systems and A. Proceedings, Eds. International Publishing: Springer, 2022, pp. 730–748.

[179] H. Sun, M. Sun, J. Weng, and Z. Liu, “Analysis of id sequences similarity using dtw in intrusion detection for can bus,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 10 426–10 441, 2022.