Physical-Fingerprinting of Electronic Control Unit (ECU) Based on Machine Learning Algorithm for In-Vehicle Network Communication Protocol "CAN-BUS"

by

Omid Avatefipour

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Computer Engineering)
in the University of Michigan-Dearborn
2017

Master's Thesis Committee:

Associate Professor Hafiz Malik, Chair
Associate Professor Kevin Hua Bai
Assistant Professor Lu Wei

This thesis is dedicated to my beloved parents and my uncle Dr. Shapour Afrashtehfar for their endless love, support, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACK	Acknowledgement
ADAS	Advanced Driver Assistant Systems
ANN	Artificial Neural Network
CAN	Controller Area Network
CAN-FD	Controller Area Network – Flexible Data Rate
CSMA/CD	Carrier Sense Multiple Access / Collision Detection
DoS	Denial of Service
EOF	End of Frame
ECU	Electronic Control Unit
FOTA	Firmware Update Over the Air
GMLAN	General Motors Local Area Network
GPS	Global Positioning System
IDS	Intrusion Detection System
IP	Internet Protocol
LIN	Local Interconnect Network
MOST	Media Oriented Systems Transport
OBD-II	On Board Diagnostics
RDS	Radio Data System
REC	Receive Error Counter
RF	Radio Frequency

RFID Radio Frequency Identification

RKE Remote Keyless Entry

SOF Start of Frame

TCP/IP Transmission Control Protocol / Internet Protocol

TEC Transmit Error Counter

TMC Traffic Message Channel

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

ABSTRACT

The Controller Area Network (CAN) bus serves as a legacy protocol for in-vehicle data communication. Simplicity, robustness, and suitability for real-time systems are the salient features of the CAN bus protocol. However, it lacks the basic security features such as massage authentication, which makes it vulnerable to the spoofing attacks. In a CAN network, linking CAN packet to the sender node is a challenging task. This paper aims to address this issue by developing a framework to link each CAN packet to its source. Physical signal attributes of the received packet consisting of channel and node (or device) which contains specific unique artifacts are considered to achieve this goal. Material and design imperfections in the physical channel and digital device, which are the main contributing factors behind the device-channel specific unique artifacts, are leveraged to link the received electrical signal to the transmitter. Generally, the inimitable patterns of signals from each ECUs exist over the course of time that can manifest the stability of the proposed method. Uniqueness of the channel-device specific attributes are also investigated for time- and frequency-domain. Feature vector is made up of both time and frequency domain physical attributes and then employed to train a neural network-based classifier. Performance of the proposed fingerprinting method is evaluated by using a dataset collected from 16 different channels and four identical ECUs transmitting same message. Experimental results indicate that the proposed method achieves correct detection rates of 95.2% and 98.3% for channel and ECU classification, respectively.

CHAPTER 1: Introduction

Nowadays with the help of advanced technology, modern vehicles are not only made up of mechanical devices but also consist of highly complex electronic devices by adding several forms of external interfaces to other vehicles (V2V and V2I communications) and even to the Internet. These external interfaces still have communication with the internal vehicular networks. Modern vehicles consist of several types of networks namely CAN, LIN, FlexRay, MOST, and recently Ethernet. Among all these communication protocol, the Controller Area Network (CAN) bus is widely used in automotive industry as predominant protocol and in embedded systems networking in general. It finds a wide range of applications from automotive, aerospace, agriculture, medical devices, and even in some of the home and commercial appliances [1]. A modern vehicle contains many different computing devices, known as Electronic Control Unit (ECU), which are responsible for sensing and controlling actuators [2]. Virtually, all functionalities in the modern automobiles ranging from engine control to braking, lighting, driver safety, antilock brake systems (ABS) and the parking assist systems are achieved through these ECUs [3]. These ECUs communicate with each other through different networks. If the communication on these networks are not secured, it can pose a serious threat to the safety of the passengers. The CAN-bus has been a de-facto standard for communication as an in-vehicle network for over 30 years. When CAN-Bus protocol was invented by Robert BOSCH GmbH [1], vehicles was considered as an isolated system which did not have any communication to the outside environment. Therefore, by design, the CAN-bus protocol lacks basic security features such as message authentication option which

makes it vulnerable to a variety of spoofing attacks [4]. For example, in the absence of effective message authentication, a single compromised ECU allows the attacker to take full control of the vehicle by injecting spoofed messages [2,5,6]. Lack of the channel encryption provides the adversary an opportunity to sniff the network traffic by simply plugging in a low-price hardware leading to the replay attacks [7]. Attack surfaces are growing over the course of time which gives rise to develop the effective protection of CAN-bus communication from malicious attackers as a challenging task. The automakers are aiming for a fully-connected intelligent vehicle which makes secure in-vehicle communication problem even more complicated. Recently, researchers have proposed many solutions for in-vehicle networks security at different layers e.g. physical layer and data link layer by using various types of message authentication methods which will be introduced and discussed in related works section.

1.1. Motivation and Aims

Since CAN packets contain no authenticator field, any ECU on the network can impersonate the other ECUs in the network. This provides a broad range of internal as well as external attack surfaces [7]. An adversary can leverage the CAN-Bus protocol vulnerabilities to launch various attacks leading to malfunctioning of the vehicle. Data encryption-based solutions are proven to be inefficient for the CAN-Bus protocol [7]. In this thesis, an intelligent method is proposed to link the received packet to its transmitter based on the unique physical properties of the signal. The proposed physical-fingerprinting-based method exploit unique artifacts both at the digital device (ECU) level and in the physical channel (e.g., CAN-bus). Material and design imperfections in the channel and the transmitter are the main contributing factors behind these unique artifacts. The physical channel unique artifacts, which are used to link received electrical signal to the source (or transmitting) ECU, are considered in this study. More specifically, the proposed method exploits

physical channel dependent attributes for linking received signals (message) to the transmitting device. Therefore, proposed method can be leveraged as an identification method in such a way that if an adversary tries to alter the message content and send a malicious message either from an external ECU or by changing the cables, it can be distinguished that the packet is received from unknown sources and based on the defined safety specifications, proper actions should be performed. Even if an adversary uses the legitimate message identifier (e.g. shut down engine), since he/she is sending that message from an external ECU, the proposed method can detect that the signal has not originated from the legitimate source because the signal patterns will not pair with the ECU that should have generated that message.

CHAPTER 2: Background and Related Works

A modern vehicle consists of different complex embedded devices aka ECU with a wide range of electronic components and interfaces which are communicating both inside the vehicle and outside world. In order to facilitate the communication among these ECUs, several types of internal networks have introduced not only for internal communication but also by utilizing different interfaces they are able to connect to external systems e.g. Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication. In order to have a better understanding for vehicular network protocol, in this chapter more detailed facets of CAN-Bus protocol are explained.

2.1. Internal Networks

In order to establish an extensive communication surface in vehicle, there are several communication protocols available which have been employed for this reason e.g. Local Interconnect Network (LIN), Controller Area Network (CAN), Media Oriented Systems Transport (MOST), and FlexRay. Aforementioned network protocols are different in terms of baud rate, communication protocol, and functionalities. To this end, there is a gateway in vehicle (which is mostly accessible from OBD-II connector) to adapt the transmission speed among these networks. A high level view of how different vehicular networks are transmitting messages is shown in figure 2.1.1 [55].

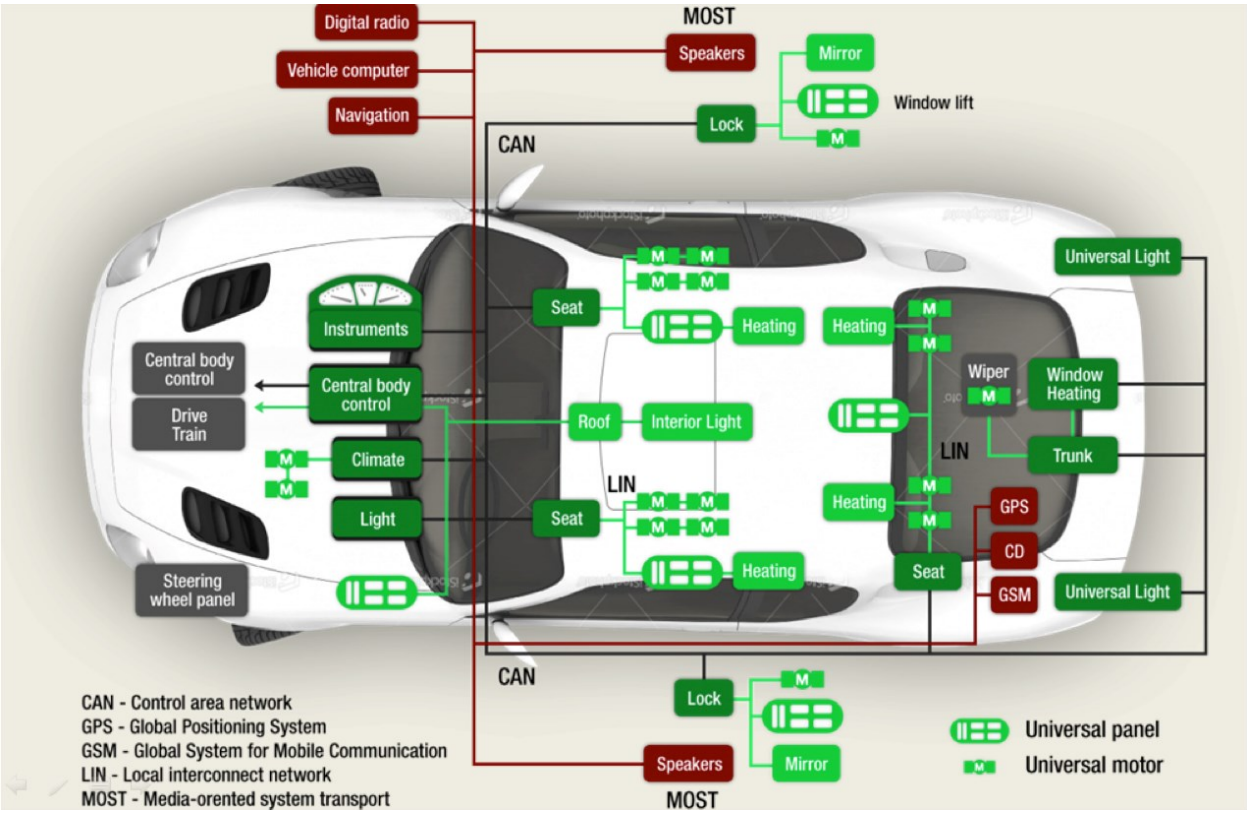

Figure 2.1.1. A high level view of vehicle networks architecture

In this thesis, the main focus is on CAN-Bus protocol since it is the most widely used protocol for critical components in the vehicle e.g. airbag, brake, engine control module, etc. [8] which can jeopardize the driver and passengers if an adversary would able to penetrate into this networks and doing malicious activities. Beside from CAN-Bus protocol, LIN is as a serial and byte oriented communication protocol with master/slave organization which is specially developed to achieve cost-effective communication for intelligent sensors and actuators in vehicle when the bandwidth, fault tolerance, and advanced features of CAN-Bus is not required. Examples of area that LIN is used can be named as: window lift, mirrors, wiper, and rain sensor, etc. [9]. Media Oriented Systems Transport (MOST) is a high-speed multimedia network topology which has been customized for automotive industry. MOST laid out as ring topology and synchronous data

communication for multimedia purposes e.g. audio and video transmission. As a result of plug & play feature which is provided in MOST protocol, adding/removing a MOST device to the existing network. In a given MOST network, one node is assigned as timing master node that continuously feed MOST frames into the ring [10]. The total bandwidth which is available in MOST protocol for transmitting of the stream data e.g. audio and video is around 150MBuad. Since the proposed technique in this thesis is introduced for CAN –Bus protocol, the following part is allocated for the CAN-Bus protocol specification.

2.2. CAN-Bus Protocol

The Controller Area Network (CAN-Bus) protocol was introduced in 1983 by Robert BOSCH GmbH as a common, small area network solution that supports distributed product and distributed system architectures which been widely applied in the automotive communication and even in domestic appliances, building automation, factory automation, military, medical devices, and entertainment domains. [1]. Compared to the TCP/IP protocol in which the origin and destination addresses are defined in each packet, CAN-Bus messages does not have origin and destination address and instead it utilizes the broadcasting communication technique that every message transmitted by a transmitter node are broadcasted to the entire network for all nodes to read and verify. Before CAN-Bus was introduced as a protocol for vehicular network, each electric component in car was required to have pair-to-pair connection to all the other components due to the unavailability of a common bus for nodes which was resulted in too much wire harnesses effort. Therefore, CAN-Bus reduced the harnessing requirement of physical network in a large extend. Since there is no destination address in CAN-Bus, each node can publish and receive particular messages based on the pre-defined node (here ECU) configuration. This communication technique increases the network elasticity [11] which means that if new ECU is supposed to add to the current

network, it will be configured easily and does not require any changes to the network infrastructure and other nodes as well. CAN-Bus is considered as an event-trigger protocol which means a message is generated in reply to the generation of event or request in the network. CAN-Bus is also considered as multi-master protocol that allows any node can publish/receive message on the bus if the communication bus is free. It uses CSMA/CD media access control method in such a way that every node on the network must monitor the bus for a period of no activity before trying to send a message on the bus (carrier sense). Once this period of no activity occurs, every node on the bus has an equal opportunity to transmit a message (multiple access). Bitwise arbitration technique (which will be explained later) is applied as collision avoidance method to listen to the network traffic during transmission and detect the collision occurrence in order to initiate the transmission. Vehicular network has introduced a variety of merits such as reducing harness in large extent, establishing data sharing, remarkably improving the intelligent control level of vehicle e.g. Advanced Driving Assistant Systems (ADAS), improving capabilities of failure diagnosis and repair and so on.

2.2.1. Message Arbitration

To meet the real-time systems deadline requirements, each message has been assigned an identifier frame which is utilized to define the message priority for bus access [12]. Priority is inversely proportional to message ID: the lower number of message identification value, the higher priority it has to gain the bus. This prioritization feature has also solved the bus access conflict in such a way that if two nodes want to send data simultaneously, each ECU which has a lower ID value will publish the message firstly. (Due to the higher priority). This technique is also known as message arbitration [11]. Generally, CAN regulates arbitration in a predictable and efficient manner. It is worth mentioning that 0 bit is considered as dominant bit, meaning that if both 0 and

1 bit (considered as recessive bit) are transmitting on the bus simultaneously by two senders, the 0 bit will gain the bus. The dominant state always wins over the recessive state. For instance, if three nodes (First node: 11001011111 in binary, second node: 110011111111 in binary, and third node 110010110010 in binary) try to transmit message simultaneously, in order to prevent bus collision, a given node with the lowest ID (in this case third node) will transmit the information firstly because it has a lowest value and highest priority than the other two nodes. The two others will stop transmitting and waiting until the bus becomes free again to retransmit the entire packet. Figure 2.2.1.1. depicts the message arbitration for this scenario.

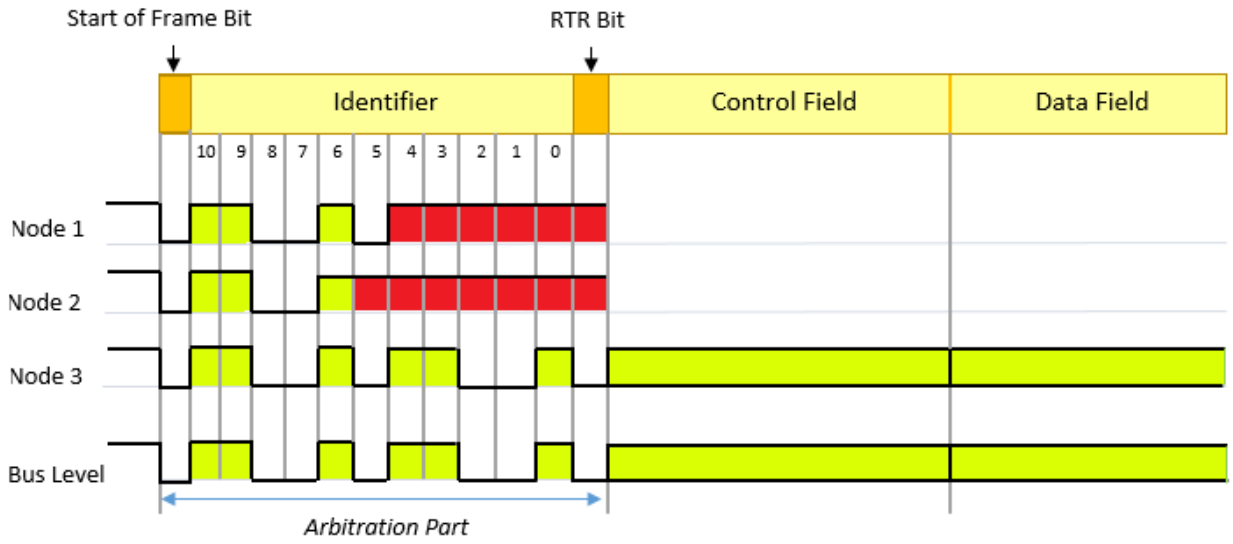

Figure 2.2.1.1. Arbitration condition in CAN-Bus protocol.

In automotive industry, differential signal voltage is mostly used for the physical layer signaling using two communication wires e.g. CAN-High and CAN-Low [13]. Shown in Figure 2.2.1.2 is the bit transition and signal voltages of CAN bus communication which includes series of dominant and recessive bits. When a recessive bit (logical 1) is transmitting both CAN-High and CAN-low are driven to the 2.5 volts which indicates that the voltage difference is zero during the transmission of recessive bit and when a dominant bit (logical 0) is transmitted, CAN-High goes

to 3.5 volts and CAN-Low goes down to the 1.5 that means the voltage difference in the dominant bit is 2 volts [14] which making CAN-Bus very resilient against electric and magnetic interferences.

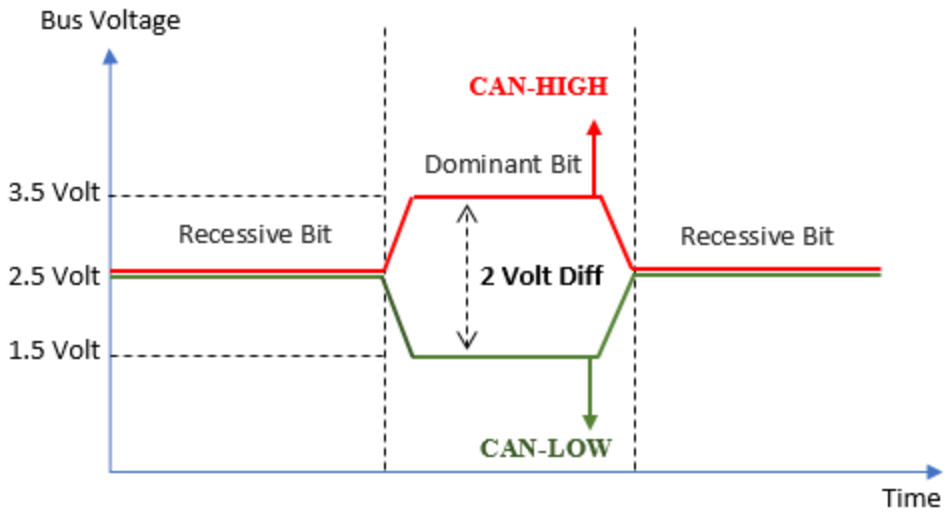

Figure 2.2.1.2. CAN-Bus differential signal illustration.

2.2.2. CAN Data Link Layer

Generally, there are two formats of CAN-Bus namely standard format which has 11 bit for identifier and extended-format which has 29 bit for identifier frame [1]. Data Frame, Remote Frame, overload frame, and error frame are four major frame types in controlled area network (CAN-Bus). If one system has standard format and the another system has extended-format can communicate with each other as long as the extended format is not used.

- **Data Frame:** This frame is used to carry the data from a transmitter to a receiver, which consists of the following bit fields: start of frame (one dominant bit), arbitration field which consists of 12 bits, control field which has 6 bit, and data field (in range of 0 to 64 bits), CRC field (16-bit), ACK field (2-bit), and End of Frame (7-bit). The complete illustration of data frame is shown in figure 2.2.2.1 [1]. Arbitration field defines the priority of each

message and also there is a single bit in this field to define this is a data frame or remote frame. Remote frame is used to enable the receiver to request another data from transmitter. The data frame can be in the length of zero (remote frame) to eight bytes and control field specifies the length of the data frame. CRC frame: CRC frame consists of 16 bits totally which 15 bits are used for Cyclic Redundant Checksum algorithm for error detection and one recessive bit as delimiter. During the message transmission, the message transmitter sends the CRC and all receivers computed a local CRC value. Each receiver compares its computed CRC with the transmitted CRC and if CRC matches, the receiver must acknowledge this and if CRC does not match, the receiver must not acknowledge and should transmit error frame. ACK field: Receiver node re-computes the CRC and if it matches, it reports this to the transmitter that tells the valid message has been received correctly. It is done by overwriting the recessive bit in ACK slot with the dominant bit. Finally, during the end of frame, the transmitter sends 7 recessive bits as a marker that indicates the end of frame. This would consider as time period of idle time between messages to vote "NO" for nodes that do not agree with the transfer.

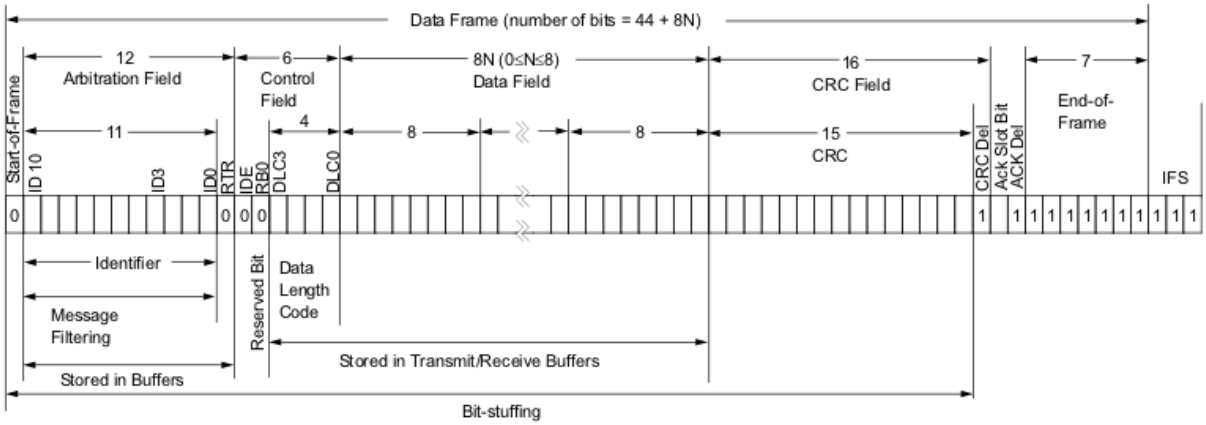

Figure 2.2.2.1. CAN Bus Data frame

- **Remote Frame:** In the case that receiver node needs more data from the sender, with the same identifier, it requests this demand by sending the remote frame [1] Remote frame is almost identical to the data frame except a few differences that the remote frame does not transmitted any data and the objective of the remote is different from data frame. Data frame and remote frame are distinguished by the RTR bit in the arbitration field (Data frame: RTR=0, Remote Frame: RTR=1). Other fields are the same in remote frame.
- **Error Frame:** Whenever each node detects an error during transmission, it will transmit the error frame which consists of two parts namely error flag and error delimiter. Error flag is given by the occurrence of the error frame. All other nodes also detect an error condition and start transmission of an error flag. Each CAN controller has a Transmit Error Counter (TEC) and Receive Error Counter (REC). The value of the error counters in CAN controller determines the error state of the CAN protocol controller (Error Active, Error Passive, Bus off). After the error frame is finished, the node tries to retransmit the message. There are 5 types of errors in CAN-Bus protocol as follows [15]:
    1. **Bit Error:** a node that is sending a bit on the bus also monitors the bus and bit error can be detected at that bit time, when the bit value that is monitored differs from the bit value sent.
    2. **Bit Stuffing Error:** a stuff error is detected when 6 consecutive recessive or 6 consecutive dominant bits are received.
    3. **CRC Error:** The CRC sequence received is not identical to the CRC sequence calculated.
    4. **Format Error:** a format error is detected when a fixed-form bit field (CRC delimiter, ACK delimiter, EOF field) contains one or more illegal bits.
    5. **Acknowledgment Error:** An ACK error is detected by a transmitter whenever it does not monitor a dominant bit during the ACK slot.

Error frame will be transmitted upon detection of error except for CRC error which is transmitted in EOF. A node detecting an error transmits an error flag like the error flag's form violates the rule of bit stuffing or destroys a bit field requiring fixed form. All the other nodes also detect an error condition, start transmitting the error flag. Collective length of all error flags varies between 6 and 12 bits. Additionally, each node increments its error counter. After the error frame is finished, the node tries to retransmit the message and retransmission can be attempted after 17 to 31 bit times. Structure of error frame is shown in figure 2.2.2.2.

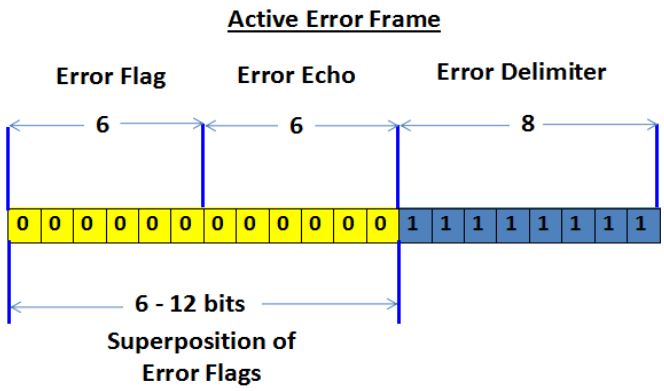

Figure 2.2.2.2. CAN-Bus Error Frame

- **Overload Frame:** The overload frame is used to make a delay whenever the receiver is not able to receive data due to some internal conditions. There are two circumstances in which overload frame can be transmitted one of them as mentioned earlier is due to the internal condition of receiver which require more delay for the next data frame or remote frame and the other reason is due to the detection of dominant bit (bit zero) during the intermission. It should be noted that the overload frame is rarely used in current application of CAN-Bus.

2.2.3. Bit Stuffing

CAN-Bus exclusively uses the recessive/dominant edge for synchronization. To maintain synchronization between all receivers and the transmitter, a sufficient number of recessive/dominant edges is required during the transmission. CAN-Bus uses bit stuffing to achieve synchronization. It is worth mentioning that bit stuffing is entirely handled by the CAN controller, therefore no software intervention is required. Bit stuffing technique is used in CAN-Bus which indicates that if there are six consecutive identical bits transmitted in the bus, it is considered as an error because bit stuffing law is violated [16]. Bit stuffing can be applied in different frames in CAN-Bus e.g. arbitration field, control field, and CRC field which means a complementary bit will be added to the frame when the transmitter finds that there are five identical bits consecutively. Therefore, six consecutive identical bits during the transmission is considered as bit-stuffing violation and error frame will be transmitted by each node which detects this situation. The stuff bit rule is used to indicate local errors. When transmitter detects a bit error and transmits an error flag, at the 6^{th} bit of the error flag all other nodes recognize a violation of the bit stuffing rule and transmit error flags. After error delimiter (8 bits) and intermission (3 bits) the transmitter tries again to access the bus to retransmit the corrupted message. In Figure 2.2.3.1. the CAN frame and how bit stuffing is applied to the frame is shown.

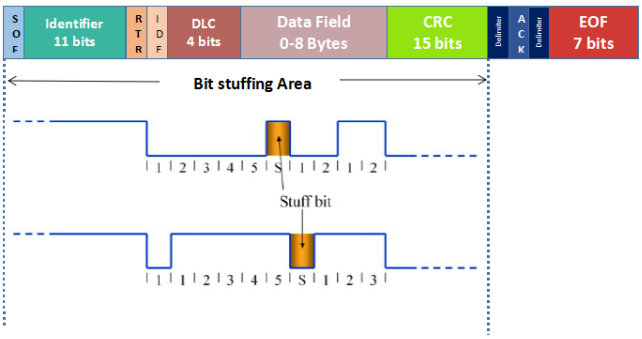

Figure 2.2.3.1. Bit stuffing technique for synchronization in CAN-Bus

2.3. CAN-Bus Limitations and Vulnerabilities

Connecting the vehicular networks to different environments, both internal networks and wireless, creates fantastic services for the automotive industry in terms of efficiency, cost and safety e.g. Vehicle to Vehicle (V2V), and Vehicle to Infrastructure (V2I) communication, Firmware-Update-Over-The-Air (FOTA) and remote diagnostics that enables embedded software components to be re-programed remotely and provides advantages for drivers in a way that they do not need to bring the vehicle to dealer for diagnostic services [17]. However, these features can introduce new challenges because both internal and external communication needs to be secured properly otherwise attackers can take full control of the vehicle and endanger the passengers' life consequently by misusing these features. The CAN-bus protocol was designed to be lightweight, robust, and fast as it should be capable of having satisfactory performance in real-time environment to meet the defined time constraints [18]. However, CAN-Bus contains several vulnerabilities which are included in its design and has paved the way for adversaries to have access to the network and inject malicious message for different purposes. From security solution standpoint, a secure communication should meet these five criteria by protocol or system security designer [19]:

- **Data Integrity:** information which is received by the receiver should be exactly the same as sender has sent in channel without any alternation.
- **Authentication:** all parties (ECUs in CAN-Bus) should be detected that they are authenticated.
- **Confidentiality:** the communication between authorized parties should be protected against unauthorized ones.

- **Nonrepudiation:** the security solution should prove that the parties in the communication cannot deny the authenticity of the message that was organized.
- **Availability:** the security solution should ensure that the system availabilities throughout different circumstances are guaranteed.

One of the inherent limitation of CAN-Bus, which makes the nodes in network to be compromised, is the lack of message authentication within each CAN message. As its name implies, CAN-Bus is a network of different controllers with different functionalities. For instance, Engine Control Unit is sending the RPM data continuously to the bus and it becomes available for all the nodes in CAN-Bus, irrespective of whether nodes in the bus have requested that message or not. The other nodes constantly listen to the bus for their specific message which can be recognized by the message identifier. The CAN-Bus architecture works fine in the normal circumstances. However, it does not provide security facilities by design to prevent unauthorized node from joining the communication and broadcast malicious messages to other nodes. These inherent vulnerabilities give the attacker a potential surface to send spoofed message after understanding the legitimate format of CAN-Bus, and each ECU can impersonate the other ECUs for replay attack which could create harmful consequences for vehicle occupant. Attackers passively listen to the bus to record different legitimate messages content for different functionalities and then he/she can inject their own messages to manipulate the vehicle functionalities [20].

Another vulnerability of the CAN-Bus protocol is the unencrypted traffic during the communication. Encryption techniques never apply during the phase of protocol design since they can make overhead for real-time communication and this would be in contrast with the nature of the protocol (lightweight and fast). This problem makes surface straightforward for adversaries to sniff the traffic by simply buying a low-price hardware which can be connected to the CAN-Bus

and passively sniff data and obviously without some forms of encryptions, message authenticity and integrity would not guarantee and then be able to perform malicious activities. Therefore, it is required to add some security level or plug-in to the current protocol to avoid these incidents [20].

Misuse of protocol is another reason that hacker can take advantage of it. For instance, as mentioned in the earlier part, CAN-Bus uses message arbitration to win the bus for data broadcasting when more than one node tries to send the data. A Denial-of-Service (DoS) attack can be launched by using the message arbitration technique in a way that adversary sends a malicious message with the highest priority (lowest ID) continuously. Therefore, the data-bus will be occupied all the time by the compromised node and could resulted in system failure [21].

2.4. Vehicular Network Interfaces

Recently, modern vehicles are not only considering as close loop system but they also have several types of communication to the outside world. Generally, the vehicular network interfaces can be categorized into four main interfaces: physical interfaces, external interfaces, short-range wireless interfaces, and long-term wireless interfaces.

2.4.1. Physical Interfaces

There are a number of internal physical interfaces inside a vehicle, some of them are directly connected to the internal network e.g. OBD-II is considered as the most well-known physical interface of vehicles because all cars built since January 1, 1996 were required to be OBD-II equipped systems and manufacturers started incorporating OBD-II in various models as early as 1994 for vehicle's self-diagnostic and reporting capability [22]. OBD-II can be used as an entry point into the vehicle network and all the CAN-Bus traffic can be monitored and logged by connecting a cable to the OBD-II port. Recently, OBD-II connector has caught the attention of security researchers as an entry point for car hacking and even the logged data which are captured

from this port, can be analyzed to monitor the normal behavior of CAN traffic and any abnormal behavior can be identified as an attack against CAN-Bus [23]. The OBD-II port has 16-pin layout which is shown and described in figure 2.4.1.1 and Table 2.4.1.1, respectively.

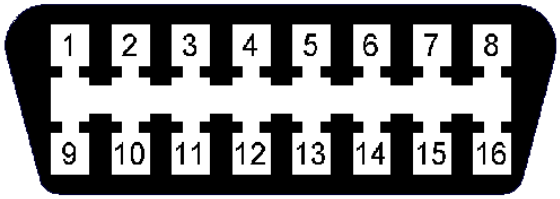

Figure 2.4.1.1 OBD-II pinout

Table 2.4.1.1. 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

As it can be observed from the above table, there are some pins which are allocated for vendor specific functionality. CAN is connected to the OBD-II connector on pins 6 and 14 for CAN High and CAN Low, respectively.

2.4.2. External Interfaces

Some ECUs might require some information from outside world e.g. GPS data, camera, data over internet, etc. to offer a wide range of functionality, comfort, and safety. Shown in figure 2.4.2.1 is the surface of external interfaces that is available in a modern vehicle [2]. Even though these external interfaces have provided interesting features in terms of comfort and safety, it is also considered as a potential risk for driver or passengers if an adversary can penetrate into them. Attacking the external interfaces consider as more dangerous level of attacks because attacker can launch an attack remotely without leaving any trace that driver can be aware of that compared to the physical interfaces entry point in which the attacker has to have physical access to the vehicle to be able to launch attacks.

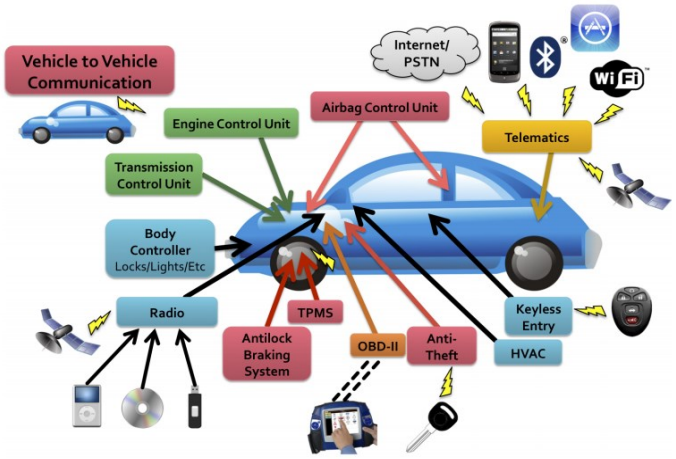

Figure 2.4.2.1 External interfaces in a modern vehicle

2.4.3. Short-Range Wireless Interfaces

Short-range wireless interfaces have been introduced in order to establish a connection between car and different objects within its near surrounding (around 10 meters). Recently, Bluetooth

technology has been widely used recently for different purposes namely Remote Parking, Exploration mode, hands-free calling and other multimedia purposes. Radio Frequency (RF) is another short-rang wireless interface which have been mostly used for Remote Keyless Entry (RKE) has become as a de-facto feature for modern vehicles which can open the doors and trunk remotely by pressing a button on key fob. Manufactures have presented different level of functionality to the driver when he/she is located within the range of RF e.g. opening the door, starting the engine, and RF identification (RFID) inside the key fob in order to avoid unauthorized operation of the car.

2.4.4. Long-Range Wireless Interfaces

Regarding the long-range of wireless interfaces, their range can be go up to infinite distance. Broadcast channels are considered as a main type of long-range which are broadcasting by the transmitter and can be tuned into by a car. Global Positioning System (GPS), Radio Data System (RDS), and Traffic Message Channel (TMC) are considered as main forms of broadcasting long-range wireless technologies which have range up to 10km. Telematics is a method of monitoring a vehicle and by combining the GPS system with OBD (On-board Diagnostic), it is possible to record and also map where the car is and how fast it is traveling and also driving style. These fantastic features that are provided by telematics can report your driving behavior to insurance companies and even if you perform some types of risky maneuvers, it can report them to the police in case a driver tries to blame for an accident. Providing communication over 3G has made it possible for vehicle to send and receive data from management systems. More importantly, with the help of telematics,3G, and 4G communication, Vehicle to Vehicle technology are not too far to be seen in streets. Even though all these features are great in terms of comfort and safety, again it can create more new surfaces for hackers to penetrate into the vehicle network remotely and

jeopardize the life of driver or passengers. Therefore, needless to say that establishing a secure and encrypted communication between vehicular network and these external interfaces is considered as super critical topic to prevent dire consequences. Furthermore, these modern technologies should be very reliable and have minimal delay.

2.5. Automotive Attacks

A practical automotive attacks have been launched by compromising an external interface. As discussed in previous part, there are several external interfaces available in modern vehicle which can be leveraged as an entry point for adversary to launch attack. These external interfaces are connected to the ECUs inside the car and these ECUs are connected to the ones via internal network e.g. CAN, LIN, FlexRay, etc. When an ECU is compromised by an adversary it can send malicious message to the internal network and even take control of the vehicle. As for attacking the physical interfaces in the car, OBD-II is the most common port for attackers to have access to the internal network. For this type of attack, adversary needs to be present in the car during the attack or he/she can remotely communicate with OBD-II. Recently, infotainment systems have become a popular feature in vehicles but it can also pose a new attack surface via USB port or CD drive. At the related work part, this type of attack will be explained as a case scenario that researchers investigated. Physical interfaces are considered as a popular tool for security researcher and penetrating testers to have access to the internal network and log the message traffic. However, this interface is less attractive since they require to be present in the car to launch attack and it not practical. Regarding launching attack via the wireless interfaces, if attacker tries to send malicious message over the Wi-Fi hotspot, he/she has to follow the victim and stay close throughout the attacks. However, if attackers try to launch attack over the cellular connection, there would not be any restriction in terms of distance as long as there is internet connection in both sides. Therefore,

attacks over cellular connection has become an important and dangerous form among the other interfaces since there would not be any trace from attacker during attack launching.

2.5. Literature Review

In this part, the state-of-the-art survey is carried out to discuss different approaches and solutions that researchers have proposed to make in-vehicle communication more secure. Researchers have worked in different CAN-Bus layers to introduce security solutions. Cho and Shin [37] proposed a clock skew based framework for ECU fingerprinting and use it for the development of Clock based Intrusion Detection System (IDS). The proposed clock based fingerprinting method [37] exploited clock characteristic which exists in all digital systems: "*tiny timing error known as clock skew*". The clock skew identification exploits uniqueness of the clock skew and clock offset which is used to identify a given ECU based on clock attributes of the sending ECU. The proposed method measures and leverages the periodic behavior of CAN-Bus messages to fingerprint each ECU in the network and then constructing a reference clock behavior of each ECU by using Recursive Least Square (RLS) algorithm. Based on the developed reference behavior, deviation from the baseline clock behavior would consider as abnormal behavior (ECU is compromised) with low rate of false positive error: 0.055%. Cho and Shin developed a prototype for the proposed IDS and demonstrated effectiveness of the proposed CIDS on three different vehicles e.g. Honda Accord, Toyota Camry, and a Dodge Ram.

Wang et al [39]. propose a practical security framework for vehicular systems (VeCure), which can fundamentally solve the message authentication issue of the CAN bus. They validate the proposed method by developing a proof-of-concept prototype using Fessscale automotive development board. In their method each node which sends a CAN packet needs to send the message authentication code packet (8 bytes) as well. They divided the ECUs into two categories

namely Low-trust group and High-trust group. ECUs which have external interfaces e.g. OBD-II or telematics are put in the low-trust group. High-trust group share a secret symmetric key to authenticate each coming and outgoing messages in a way that an ECU from Low-trust group that does not know the key cannot send message to critical ECUs in high-trust group. Wang et al. used SHA-3 hash function but they improve the system throughput by pre-calculating of the heavy weighted cryptographic function. The proposed method creates 2000 additional clock cycle compared to the system without message authentication technique (equals to the 50 micro second by running on the 40 MHz processor). By offline pre-calculating the hash function their method is 20-fold faster computationally than the other methods which uses message authentication solutions. Figure 2.5. depicts the proposed method, CAN-Bus without message authentication, and classic SHA-3 hash function in terms of number of CPU clock cycles that they consume.

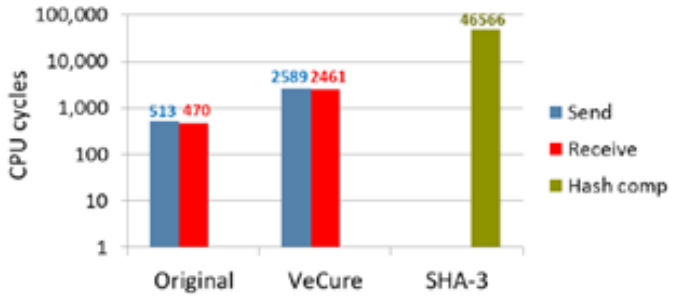

Figure 2.5. No. of CPU cycle of CAN-Bus without message authentication, VeCure, and classic SHA-3 hash function.

Koscher et al. [40] carried out a comprehensive experimental analysis of vehicle attack surfaces. They have analyzed different threat models and vulnerabilities with different range of vectors e.g. diagnostics mechanics sessions in which the adversary has a physical access to the bus via OBD-II port and by running a program on laptop to inject malware to the CAN-Bus. Infotainment systems in modern car have introduced several fascinating features e.g. connecting to the internet,

cellphone, importing all the cellphone log to the infotainment screen like contact lists, etc. These features open a new surface for attackers to inject the malware in an audio file and by playing the modified audio file, the infotainment systems can be comprised and finally the attacker can steal the logged data which have saved at infotainment systems. Koscher et al. also examine both short rage wireless access e.g. Bluetooth, remote key less entry, RFID, and long range wireless e.g. GPS and satellite radio. They perform different attacks with the help of these surfaces. For instance, they manipulated the WMA audio file in a way that it is played perfectly on PC. However, in the background it sends CAN-Bus messages when the CD is played by the victim vehicle. The question which might come up to the mind is that why car manufactures do not consider these vulnerabilities during the CAN-Bus development? Koscher et al. discussed that vehicles had not been targeted for these types of attacks and on that time there were not as diverse surfaces of communication as we have recently. But vehicles nowadays as connected with several short-range and large-range wireless network and by introducing V2V & V2I communication this trend is continuingly growing and consequently the opportunities for attackers would be more provided and in-vehicle network vulnerabilities will be increased as well.

Paar et al. [41] researchers from the Germany presented that the remote keyless entry which is becoming predominant feature for modern vehicles can be comprised and they can break the system based on the Keeloq RFID technology. This vulnerability can be applied to all remote keyless entry or other remote building access control systems which use Keeloq as cipher. They showed that the key less remote access can be compromised from a distance of 100 meters. Theoretically, the car generates random value which will be processed by the remote keyless module and by matching the correct calculation the car door will be open. Replay attacks are not allowed by the security protocol that even an adversary records all communication between two

parties and try to impersonate one of the parties later on, the replay of the log file does not allow him to open the door. However, Paar et al. applied the side channel attack of these systems.

Hoppe et al. [42] performed four different tests on the control of window lift, warning light, airbag control systems and central gateway. They also classified and summarized their result in the CERT taxonomy for the security penetration and vulnerabilities of each part and analyze two selected counter-measures. They provide some short-term and long-term solution and believe the short-term solutions can adopt into the current vehicle electronic systems but for the long-term solution some major alternation in the protocol design is required. For instance, intrusion detection systems (IDS) and data analysis is introduced as short-term security solution. In the first scenario the electric window lift is targeted in the CANoe (simulation software by Vector CANTech company) in which the vehicular network is simulated and when a predefined condition is met (car speed goes beyond 200 km/h) by adding some lines of malicious codes, the electric window lift automatically is opened and will not close until the end of attack. This attack lies down on the "Read" and "Spoof" method to monitor the current traffic and when it reaches the specified condition, it spoofs the command for electric window lift and finally Denial of Service will be performed and does not allow driver to halt the attack when it is running. Hackers use the vulnerabilities of CAN-Bus since the messages are not authenticated during the communication and the malicious code is sent from the unauthorized ECU.

For the second scenario, Hoppe et al. target the warning lights (indicators). In the normal circumstances, when unauthorized opening of a door happens, the corresponding door sensor will send message to the ECU and some events will be triggers e.g. generating light and horn alarm for a couple of seconds. In this scenario when hacker opens vehicle door, the triggered "on" alarm will be set to "off" immediately which leads to turning off the light bulb and horn switched off and

thief can steal the car or the items from the interior without any alarm. Again this vulnerability lies down on the CAN-Bus architecture communication (no message authentication) and this is “read” and “spoof” attack action and Denial of Service (DoS) as well. In the third scenario Hoppe et al. analyze the air bag control system. In this attack scenario, the air bag module will be removed from the system which leads to dire consequence during the car accident (air bog does not work in emergency cases). They believe that the intention of this attack can be monetary goals because after air bag deploys in the accident, its substitution could be costly. This attack scenario can be done by a compromised powertrain subnetwork ECU or by connecting a hardware to the OBD-II port. Additionally, they controlled the air bag controller indicator that does not indicate the air bag failure anymore. In table 2.5.1. the CERT classification of three aforementioned scenarios are summarized:

Table 2.5.1. CERT Classification of three attack scenarios

Scenario	***Attacker***	***Vulnerability***	***Action***	***Target***	***Result***
Electric window system	Hackers By injecting the malicious code	CAN bus protocol no message authentication	Read/ spoof	Control Unit (e.g. right door)	Blocking of the window system (DoS)
Warning lights (indicators)	Thieves by injecting malicious code	CAN bus protocol no message authentication	Read/ spoof	Control Unit (ECU)	Blocking of the warning light system (DoS)
Air bag control system	Re-seller By injecting malicious code (OBD-II) port	CAN bus protocol no message authentication	Read/ Spoof Copy	Air bag ECU	Theft of resources (airbag function)

One of the short-term countermeasure is developing the Intrusion Detection Systems (IDS). When a malicious activity or network pattern is detected by an intelligent detection system, it should create some alarm or warning to limit the consequences of the attack. e.g. stop the car at the next safe position. One capability that an IDS is detecting the message frequency. For instance, in scenario 1 & 2 the corresponding messages send in a constant frequency from a specified identifier. Attacker basically tries to send the exact identifier but with different content. Since removing the existing message is hard to achieve, therefore adversary will try to send the altered message with the same identifier within the significantly higher frequency. Hence, if the IDS can detect the high frequency of suspicious activity, it can create some warning alarm to the driver accordingly.

Hiroshi et al. [43] proposed a security authentication monitoring system for CAN-Bus which uses MAC for protecting CAN bus against spoofing attacks. The role of monitoring node in their proposed method is to authenticate each ECU and verified the authentication code which is defined for each CAN message. The modified CAN controller is required to install for their monitoring node to implement the message authentication which transmits an error frame to overwrite spoofed message. Additionally, if the monitoring node is compromised or removed from the bus, the entire network is compromised.

Hazem et al. [44] proposed a Lightweight CAN Authentication Protocol LCAP. The proposed method requires to append a “magic number” which can be generated on the one-way hash function employed in TESLA protocol [45] for the message to be verified from the receiver side. Handshake technique is used for node synchronization and channel security. It requires 2 bytes of the data field for the authentication code which only creates small overhead for message authentication code exchange among the nodes. However, since the LCAP introduces the new IDs in the network configuration, it requires large address space.

CHAPTER 3: Methodology

In this thesis, a physical-fingerprinting method is proposed to link the received packet to its transmitter based on the unique physical properties of the signal. The proposed physical-fingerprinting-based method exploit unique artifacts both at the digital device (ECU) level and in the physical channel (e.g., CAN-bus). Material and design imperfections in the channel and the transmitter are the main contributing factors behind these unique artifacts. The physical channel unique artifacts, which are used to link received electrical signal to the source (or transmitting) ECU, are considered in this study. More specifically, the proposed method exploits physical channel dependent attributes for linking received signals (message) to the transmitting device. The proposed method can be leveraged as an identification method in such a way that if an adversary tries to send a malicious message either from an external ECU or by changing the cables, it can be distinguished as a malicious activity and based on the defined safety specifications proper actions can be performed. Even if an adversary uses the legitimate message identifier (e.g. shut down engine), since he/she is sending that message from an external ECU, the proposed method can detect that signal has not originated from the legitimate one because the signal will not pair with the ECU that should have generated that message. It has been observed that uniqueness of the physical attributes exists both in time and frequency domain. In this thesis, a feature vector consisting of 11 time and frequency domain statistical signal attributes including higher-order moments, spectral flatness measure, minimum, maximum, and irregularity K are considered to

capture the channel and the transmitter dependent uniqueness. A multi-layer neural network based classifier is trained and tested for source ECU and the source channel. Experimental results indicate that the proposed attributes can be used to classify different channels and ECUs. Performance of the proposed fingerprinting method is evaluated on a dataset collected from 16 different channels and four identical ECUs transmitting the same message. Experimental results demonstrate that the proposed method achieves correct detection rates of 95.2% and 98.3% for channel and ECU classification, respectively.

3.1. Machine Learning Algorithms

Following the definition of (Mitchell, 1997), machine learning techniques are defined as an intelligent algorithm that have the ability to learn from historical data [46]. The task of machine learning can be divided into clustering, classification, and prediction. Nowadays, by emerging the machine learning and intelligent algorithms, several methods are proposed in various engineering application e.g. intelligent controller design for industrial robots [24-26], Intrusion Detection Systems (IDS) [27-29], adaptive optimization algorithms [30-32], etc. Machine learning algorithms have been widely used as a powerful mathematical tool to develop security solution in the area of vehicular networks [33-36]. Each machine learning technique consists of 5 major steps which is illustrated in figure 3.1.1. The first step is data acquisition in which the data should be gathered to import to the machine learning technique. Sometime, the gathered data needs to be pre-processed. For instance, there could be some noise or cleaning the invalid data. Data pre-processing is considered as an important step in order to make sure that machine learning method provides a satisfactory result since dirty data can affect system performance. Following the data pre-processing step, further optional transformation might be required to establish standard data representation in order to provide more efficient processing e.g. transforming the data to frequency

domain by Fourier transformation. Having completed the necessary transformation process, the feature extraction would be the next step to extract the features from the pre-processed input data. Finally, the feature selection is performed in order to provide the most relevant and distinguishable feature. In addition, some data visualization technique might be required to apply for better output representation and interpretation.

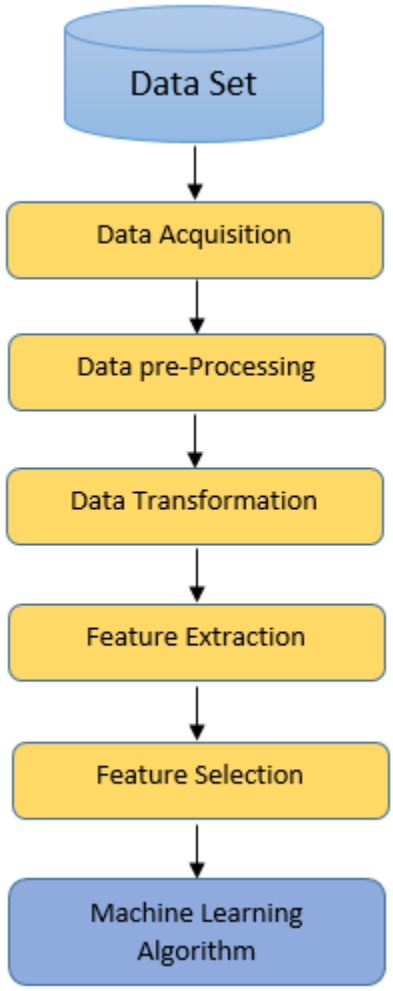

Figure 3.1.1. Machine Learning major steps

All the aforementioned processes are carried out during the training phase to extract knowledge from a given training data set. The final objective would be developing a system that can detect or classify unseen data during the testing phase. It should be noted that training and test phases might

be required to repeat several times in order to optimize the initial parameters of proposed algorithm. In this thesis machine learning algorithm is utilized to identify each ECU that are transmitting messages and based on the inimitable signal characteristics which exists during transmission, a physical- fingerprinting of each ECU can be obtained. In the following parts the proposed algorithm will be explained in more detail. There are several number of machine learning techniques available [47] e.g. linear regression, logistic regression, decision tree, Support Vector Machines (SVM), ensemble method, Artificial Neural Networks (ANN), Naïve Bayes, K-nearest neighbors, Random Forest, bootstrap aggregation, stacked aggregation, Genetic Algorithm (GA), etc. to name a few. In this thesis, Artificial Neural Network (ANN) is selected as a machine learning algorithm to identify each ECU during message transmission. Explanation of the other methods is out of the scope of this thesis.

3.2. Artificial Neural Networks

As mentioned at the previous part, there are several machine learning algorithms available for classification, predication, and identification. In this thesis, Artificial Neural Network is utilized for classification purpose due to its capability for mapping non-linear input data to the output. Generally speaking, classification task is assigned as a procedure of assigning an instance C_v to one of k classes w_c. The number of classes and their labels are known based on the problem statement. Here, there are 4 classes available since the proposed method is supposed to identify four different ECUs in the experimental setup (More explanation is provided in the experimental setup and analysis section). The classification procedure can be also defined as learning a function that can map input variables to a pre-defined set of output variables (class labels in supervised learning). Neural Network is a powerful data modeling method that is capable of capturing and representing complex input/output relationships. A neural network usually contains a large number

of parallel processes in parallel and their powerful feature to learn by providing examples has made it very flexible and popular method in machine learning domain. Artificial Neural Network is considered as an information processing paradigm which is inspired by the way that brain neurons is working. The fundamental processing element in this algorithm is a neuron. In biological neurons system, each neuron receives an input from the other sources, perform some type of combinations, and doing a general nonlinear operation on the results and finally provide the output results. The relation of these parts is illustrated in figure 3.2.1.

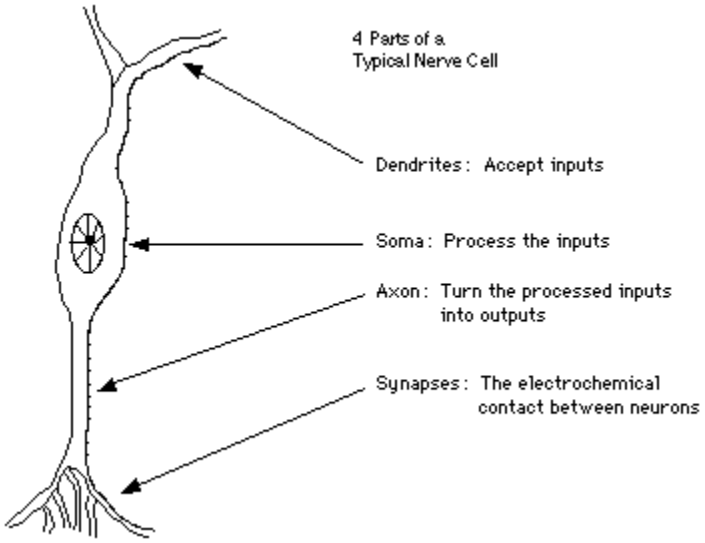

Figure 3.2.1. biological neuron construction

Artificial Neural Network can learn performing tasks (Here classification) automatically by taking examples with task-specific programming. In similar to the biological neural network, artificial neural network is also consisting of connected units (artificial neurons). Each connection between neurons can transmit a signal from one node to other ones. The receiving neurons can process the signal and perform the same transmission to the other nodes. In a common ANN architecture, each node is usually a real number and a non-linear function is used to provide the output of each node

by doing summation of the input neurons. Each neuron might also have weights which can be different as learning proceed which can manipulate the transmitting signals. Generally speaking, neurons are grouped into different layers that each layer might conducting different forms of transformation function on their inputs. Signals starts traversing from input layer to one or more level of hidden layers and finally to the output layer. Figure 3.2.2. shows the high level architecture of artificial neural network.

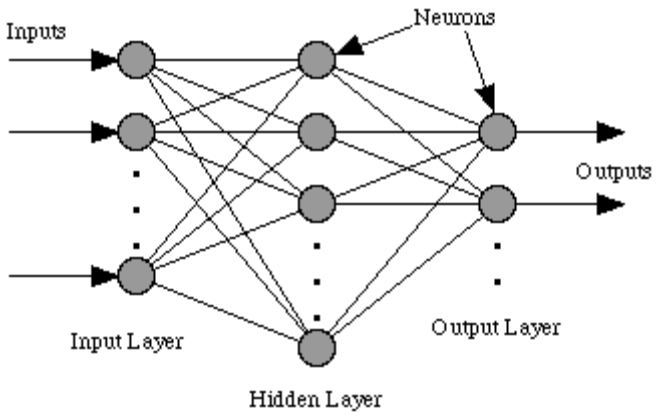

Figure 3.2.2 Neural Network Architecture

More detail view of artificial neural network with transfer function is shown in figure 3.2.3.

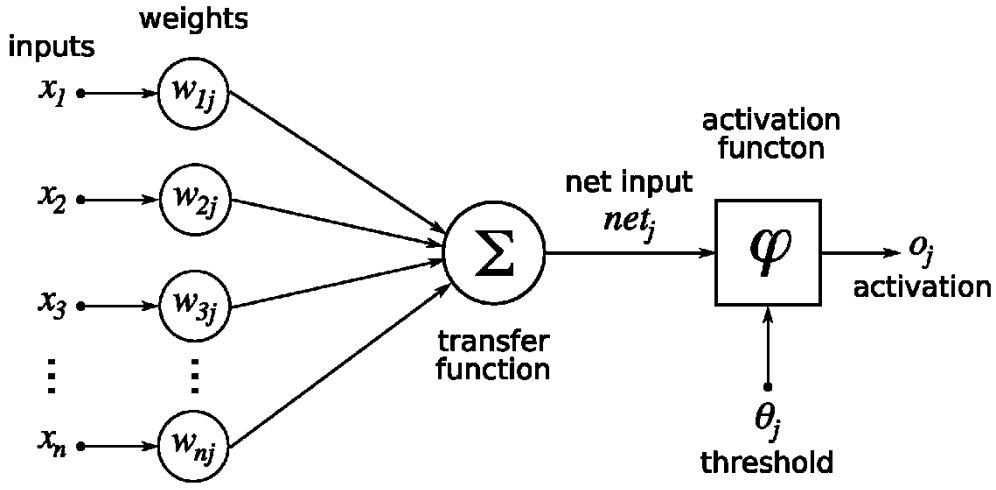

Figure 3.2.2. Artificial Neural Network construction with transfer function.

In this thesis the artificial neural network has been employed as classification algorithm in supervised learning environment. In this mode the actual output of a neural network will be compared to the desired output. In the supervised learning environment, each neuron's weight is initialized randomly in the beginning and by each iteration the weights will be adjusted accordingly in order to produce closer match between the actual output and desired one. The main goal of training is to minimize the error of each processing element by tuning the weights to reach the acceptable network accuracy. The training phase might take a lot of time and would be completed when the network achieves the desired accuracy to provide outputs based on the given set of inputs. When no further learning is required, the weights parameter will be frozen for the application. In the supervised learning environment, the output classes are labeled and neural network tries to classify every data point to its corresponding output class with minimum miss-classification. The task of classification is also considered as learning function to map input variables to a set of pre-defined output variables. In this topic an example of classification would be classify each ECU that are sending message on CAN-Bus based on the features mean, standard deviation, minimum, maximum values, etc. The whole feature set for this purpose is introduced in the physical-fingerprinting part. The accuracy of ANN for classification can be represented as confusion matrix based on the classification results which have achieved from classifying data with known class labels (supervised learning). Confusion matrix for the ECU physical-fingerprinting method will be explained later on. Several adaptive learning algorithms for feed-forward neural networks are introduced by researchers. Many of these training algorithm are based on the gradient decent algorithm as a famous method in optimization theory. From an optimization standpoint, learning an artificial neural network is equal to minimizing a global error function which depends upon the neuron weights. Since learning phase in the real neural network applications may require

adjustment of several thousand weights, only optimization methods that are applicable to large-scale problems, are relevant as alternative learning algorithms. The backpropagation algorithm is defined as a powerful method to minimize the error function in weight domain by using the gradient decent. Since in the feedforward artificial neural network all the neurons are connected to each other, the combination of weights to minimize the error function would be considered as the optimal solution of the learning problem. The basic approach in learning is to start with an untrained network, present an input training pattern and determine the output. The error or criterion function is some scalar function of the weights that is minimized when the network outputs match the desired outputs. The backpropagation learning rule is based on gradient descent. The weights are initialized with random values, and are changed in a direction that will reduce the error. In figure 3.2.3 the more detail of each neuron in terms of its activation function is illustrated.

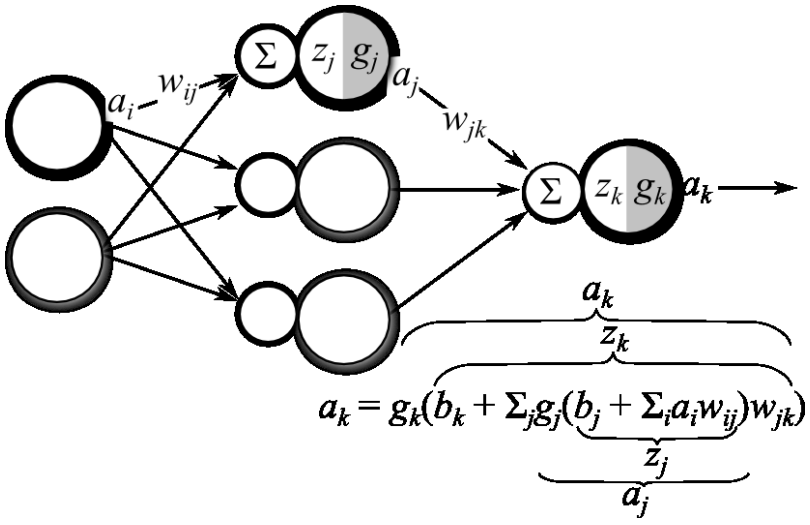

Figure 3.2.3. Artificial Neural Network diagram with transformation function

As it is depicted in figure 3.2.3, the values of input layer a_i are multiplied by a set of fully-connected weights w_{ij} which establish a connection between input and hidden layer. These weights values are then summed with the basing factor b_j and the output provides pre-activation value for

hidden layer (z_j). Then the pre-activation value will be transformed by the hidden layer activation function g_i to construct the feed-forward activation value a_j which is transforming to the next level. Similarly, the hidden layer activation values a_j are multiplied by the weights connecting the hidden layer to the output layer, a biasing factor b_k is also added, and the resulting value is transformed by the output activation function g_k to build the network output a_k. Finally, the output is compared to a desired target value t_k and the error between the output value and desired target value will be calculated.

3.3. CAN-Bus Physical-Fingerprinting Method

In this thesis, a physical fingerprinting method to link the received packet to its transmitter based on the unique physical properties of the signal in introduced. The proposed physical-fingerprinting-based method exploit unique artifacts both at the digital device (ECU) level and in the physical channel (e.g., CAN-bus). Material and design imperfections in the channel and the transmitter are the main contributing factors behind these unique artifacts. The physical channel unique artifacts, which are used to link received electrical signal to the source (or transmitting) ECU, are considered in this study. More specifically, the proposed method exploits physical channel dependent attributes for linking received signals (message) to the transmitting device. The proposed method can be leveraged as an identification method in such a way that if an adversary tries to send a malicious message either from an external ECU or by changing the cables, it can be distinguished as a malicious activity and based on the defined safety specifications proper actions can be performed. Even if an adversary uses the legitimate message identifier (e.g. shut down engine), since he/she is sending that message from an external ECU, the proposed method can detect that signal has not originated from the legitimate one because the signal will not pair with the ECU that should have generated that message. It has been observed that uniqueness of the

physical attributes exists both in time and frequency domain. In this study, a feature vector consisting of 11 time and frequency domain statistical signal attributes including higher-order moments, spectral flatness measure, minimum, maximum, and irregularity K are considered to capture the channel and the transmitter dependent uniqueness. A multi-layer neural network based classifier is trained and tested for source ECU and the source channel. Experimental results indicate that the proposed attributes can be used to classify different channels and ECUs. Performance of the proposed fingerprinting method is evaluated on a dataset collected from 16 different channels and four identical ECUs transmitting the same message.

The proposed transmitted identification method relies on the fact that each electronic device (e.g. ECU) and channel impulse response of the physical channel (e.g., CAN-Bus) exhibit unique artifacts which can be used for linking received signal to the sending ECU. More specifically, by extracting the distinguishable statistical features of transmitting signals, the source of the coming message is identified.

Let $S_i(t)$ be the output of the i^{th} ECU and $h_j(t)$ be the impulse response of the j^{th} physical channel between i^{th} ECU and the physical fingerprinting (PhyFin) unit. The physical signal at the input of the PhyFin unit, $y_{ij}(t)$, can be expressed as Equation 1 and Figure 3.3.1, respectively.

$$y_{ij}(t) = h_j(t) * S_i(t)$$

where, * denotes convolution operator.

Convolution is a formal mathematical operation, just as multiplication, addition, and integration. Addition takes two numbers and produces a third number, while convolution takes two signals and produces a third signal. Convolution is used in the mathematics of many fields, such as probability

and statistics. In linear systems, convolution is used to describe the relationship between three signals of interest: the input signal, the impulse response, and the output signal.

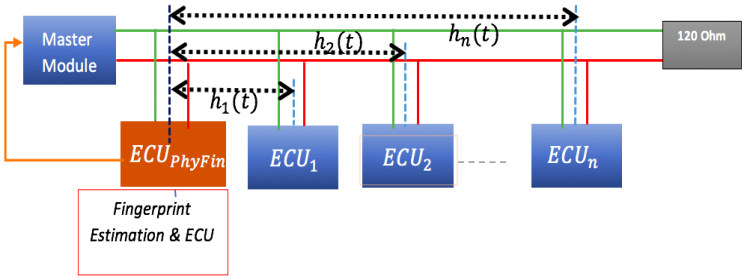

Figure 3.3.1. Physical input signal and channel response

Physical signal at the input of PhyFin unit, $y_{ij}(t)$ is used for linking $y_{ij}(t)$ to its source. Shown in Figure 3.3.2 are plots of four waveforms at the output of four different channels when identical message is applied at the input of these channels. It can be observed from Figure 3.3.2 that channel impulse response is different for all four channels, which validates our claim of channel specific uniqueness.

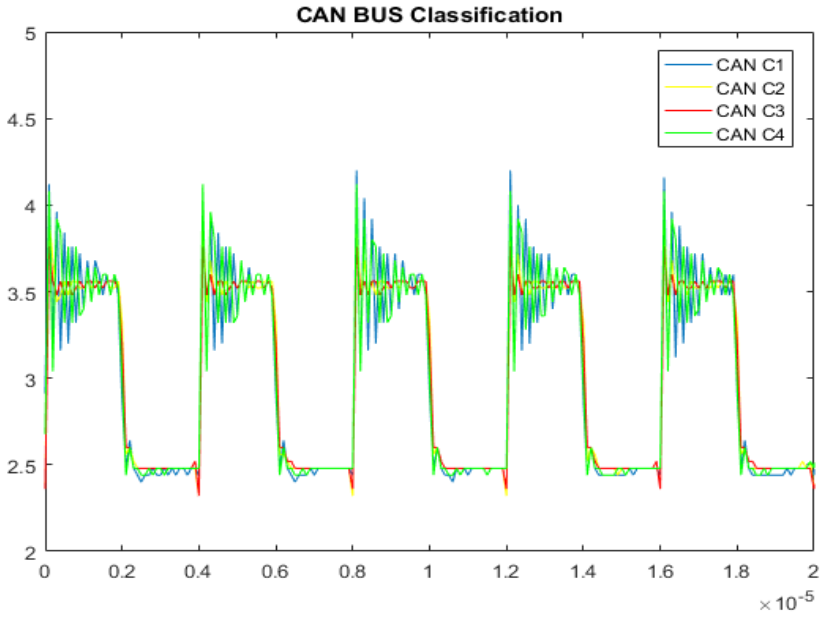

Figure 3.3.2. Waveforms of the received signals from four different CAN-bus channels with identical channel input message.

3.3.1. Feature Extraction and Selection

Feature extraction is considered as an attribute reduction process [48]. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Models built on extracted features may be of higher quality, because the data is described by fewer, more meaningful attributes. Feature extraction is also employed to improve the speed and efficiency of supervised machine learning algorithm. In contrast to dimensionality reduction methods such as projection (PCA) or compression, feature selection methods do not alter the original representation of the variables but it is considered as a process in which the number of features can be decreased by identifying and removing non-informative features. Since they preserve the original representation of the variables, accuracy of classifier will not be reduced after removing those features and selecting only a subset of informative features. Furthermore, determining an appropriate feature selection can reduce complexity and dimensionality of the feature space which leads to processing rate acceleration. Therefore, feature selection is one of the most important processes in machine learning systems particularly for the signals which are acquired from each ECU. Various feature extraction methods, both in time and spectral domain are evaluated in this study. To validate effectiveness of the proposed method here, feature extraction method presented in [49] is considered. To this end, 40-dimensioanl scalar features both in time and spectral domain are extracted using LibXtract - *a library for feature extraction* [50]. 40 scalar features both in time and frequency domain are summarized in Table 3.3.1.

Table 3.1.1. 40 scalar features both in time and frequency domain

Num.	Feature Name	Description
1	xtract_mean	Extract the mean of an input vector
2	xtract_variance	Extract the variance of an input vector
3	xtract_standard_deviation	Extract the deviation of an input vector.
4	xtract_average_deviation	Extract the average deviation of an input vector.
5	xtract_skewness	Extract the skewness of an input vector.
6	xtract_kurtosis	Extract the kurtosis of an input vector.
7	xtract_spectral_mean	Extract the mean of an input spectrum.
8	xtract_spectral_variance	Extract the variance of an input spectrum.
9	xtract_spectral_standard_deviation	Extract the deviation of an input spectrum.
10	xtract_spectral_skewness	Extract the average deviation of an input spectrum.
11	xtract_spectral_kurtosis	Extract the kurtosis of an input spectrum.
12	xtract_spectral_centroid	Extract the centroid of an input vector.
13	xtract_irregularity_k	Calculate the Irregularity of an input vector using a method described by Krimphoff (1994)
14	xtract_irregularity_j	Calculate the Irregularity of an input vector using a method described by Jensen (1999)
15	xtract_tristimulus_1	Calculate the Tristimulus of an input vector using a method described by Pollard and Jansson (1982)
16	xtract_smoothness	Extract the smoothness of an input vector using a method described by McAdams (1999)
17	xtract_spread	Extract the spectral spread of an input vector using a method described by Casagrande(2005)
18	xtract_zcr	Extract the zero crossing rate of an input vector.
19	xtract_rolloff	Extract the spectral rolloff of an input vector using a method described by Bee Suan Ong (2005)
20	xtract_loudness	Extract the 'total loudness' of an input vector using a method described by Moore, Glasberg et al (2005)
21	xtract_flatness	Extract the spectral flatness measure of an input vector, where the flatness measure (SFM) is defined as the ratio of the geometric mean to the arithmetic mean of a magnitude spectrum.
22	xtract_flatness_db	Extract the LOG spectral flatness measure of an input vector.
23	xtract_tonality	Extract the tonality factor of an input vector using a method described by Peeters 2003.
24	xtract_noisiness	Extract the noisiness of an input vector using a method described by Tae Hong Park (2000)

25	xtract_rms_amplitude	Extract the RMS amplitude of an input vector using a method described by Tae Hong Park (2000)
26	xtract_spectral_inharmonicity	Extract the Inharmonicity of an input vector.
27	xtract_crest	Extract the spectral crest of an input vector using a method described by Peeters (2003)
28	xtract_power	Extract the Spectral Power of an input vector using a method described by Bee Suan Ong (2005)
29	xtract_odd_even_ratio	Extract the Odd to even harmonic ratio of an input vector.
30	xtract_sharpness	Extract the Sharpness of an input vector.
31	xtract_spectral_slope	Extract the Slope of an input vector using a method described by Peeters(2003)
32	xtract_lowest_value	Extract the value of the lowest value in an input vector.
33	xtract_highest_value	Extract the value of the highest value in an input vector.
34	xtract_sum	Extract the sum of the values in an input vector.
35	xtract_hps	Extract the Pitch of an input vector using Harmonic Product Spectrum (HPS) analysis.
36	xtract_f0	Extract the fundamental frequency of an input vector.
37	xtract_failsafe_f0	Extract the fundamental frequency of an input vector.
38	xtract_wavelet_f0	Extract the fundamental frequency of an input vector using wavelet-based method.
39	xtract_midicent	Convenience function to convert a frequency in Hertz to a "pitch" value in MIDI cents
40	xtract_nonzero_count	Extract the number of non-zero elements in an input vector.

The extracted feature set is then analyzed further to select relevant features. FEAST Toolbox is applied [51] which utilizes the joint mutual information criterion, for ranking the features in order to select the most informative and relevant features among the features. FEAST Toolbox can be added in MATLAB toolbox. To this end, 11 features both in time and frequency domain have been achieved and summarized in Table 3.1.2, and Table 3.1.3, respectively.

Table 3.1.2. Time-domain feature set

Feature name	Equation
Maximum	$m_{ij} = (\mathrm{Min}(y_{ij}(i)) \mid i=1\ldots N)$
Minimum	$M_{ij}=(\mathrm{Max}(y_{ij}(i)) \mid i=1\ldots N)$
Mean	$\mu_{ij}= \frac{1}{N} \sum_{i=1}^{N} y_{ij}(i)$
Variance	$\sigma_{ij}^2=\sqrt{\frac{1}{N-1} \sum_{i=1}^{N} y_{ij}(i)-\mu_{ij}}$
Skewness	$\rho_{ij}= \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_{ij}(i)-\mu_{ij}}{\sigma_{ij}}\right)^3$
Kurtosis	$\kappa_{ij}= \frac{1}{N} \sum_{i=1}^{N} \left(\frac{y_{ij}(i)-\mu_{ij}}{\sigma_{ij}}\right)^4 -3$

Table 3.1.3. Frequency-domain feature set

Feature Name	Equation
Spectral Std-Dev	$\sigma_s=\sqrt{(\sum_{i=1}^{N} (y_f(i))^2*(y_m(i))) / \sum_{i=1}^{N} (y_m(i))}$
Spectral Skewness	$\rho_s= (\sum_{i=1}^{N} y_f(i)(y_m(i)) / \sigma_s^3$
Spectral Kurtosis	$\kappa_s=(\sum_{i=1}^{N} (y_m(i)-C_s)^4*y_m(i)) / \sigma_s^4-3$
Spectrum Centroid	$C_s=(\sum_{i=1}^{N} y_f(i)y_m(i)) /(\sum_{i=1}^{N} y_m(i))$
Irregularity-K	$IK_s=\sum_{i=2}^{N-1} \mid y_m(i) - \frac{y_m(i-1)+y_m(i)+y_m(i+1)}{3} \mid$

Note: y_m and y_f are the magnitude and the frequency vectors respectively

To this end, the proposed method will extract those aforementioned features from feature set and then by using Artificial Neural Network, the system would identify individual ECUs, each with its own inimitable signal characteristics during the message transmission. For this purpose, the

concept of using ANN is to train a classifier with lot of CAN-Bus messages that each ECU is emitting and the message that malicious or compromised ECU sends can be identified by the trained classifier (Here ANN). It is worth mentioning that classification algorithms have been widely employed as a powerful method for security solutions. For instance, Intrusion Detection System (IDS) system can be designed along with a classification algorithm to learn the normal behavior of CAN-Bus traffic and any deviation from that would be identify as an abnormal behavior of CAN. In order to generate each ECU fingerprint patterns, designing a classifier to receive its input from each ECU by observation, which are 11 aforementioned features both in time and frequency domain and then train the classifier to distinguish and recognize each signal's features to use as reference and later on match the upcoming message for the ECU which is supposed to send this message. Hence, if the signal's patterns do not match that ECU, it can be concluded that the coming message has been received from an external source.

3.3.2. Attack Taxonomy

Generally speaking, the objective of attackers who target the vehicles are to penetrate into the CAN-Bus network to transmit malicious messages to be able to take partial or full control of vehicle. Since the CAN-Bus protocol does not include message authentication in its transmitting packets, attackers can simply launch attacks by establishing a connection to the CAN-bus network and performing the replay attack. There are two major scenarios that attackers could be able to penetrate into the network.

Attack Type 1: In this scenario, an adversary would have physical access to the CAN-Bus network through an external device (ECU) that is connected to the vehicle. The most common entry point for this scenario would be On-Board-Diagnostics (OBD-II) connector which is located under the steering wheel. As discussed earlier, OBD-II cable is utilized to give the vehicle owner or dealers

to have access to the network and perform diagnostics tasks for different subsystems in the car. If an adversary could be able to have access to this connector, he/she can log the whole CAN-Bus traffic and since the CAN-bus traffic is not encrypted, messages would be interpreted by doing some reverse engineering effort. This type of attack can be identified by the proposed method if a signal fingerprint does not match with any signal patterns that was applied for ANN training. The monitoring system (trained by ANN) needs to determine whether there is sufficient match between the transmitting signals and reference ones. If there is not enough match with new coming signals, the system can conclude that attacker is trying to transmit that signal form an unknown external device.

Attack Type 2: this attack scenario is considered as a situation in which the attacker compromises an existing ECU in the vehicle network. An adversary tries to transmit with correct and legitimate signals but from another ECU which has been hostage. Even though the attackers believe they are injecting correct messages and there would not be any avoidance for their activity in this situation, they cannot change this fact that each ECU leaves an inimitable signal patterns that is unique for each ECU. Material and design imperfections in the channel and the transmitter are the main contributing factors behind these unique artifacts that attackers cannot change it. Due to the effect of hardware inconsistencies, the subtle difference among different ECU can be observed. Therefore, this signal uniqueness has been leveraged as ECU fingerprinting method and attacker cannot hide this fact that a malicious message is being sent from an alien external source. In addition to the ECU signal uniqueness, channels are also introducing new fingerprinting in such a way that even if an identical message is transmitted by different channels, the inimitable signals have been observed in the receiver side. This can be also leveraged to combined with the ECU fingerprinting to introduce more uniqueness for each ECU with its corresponding channel.

CHAPTER 4: Experimental Results and Evaluation

The proposed method has been evaluated by conducting a series of experiments. The success- rate of proposed method has been assessed both for ECU and channel identification by using common metrics available for machine learning algorithm. The experimental evaluation is divided into two major categories namely channel identification and ECU identification.

4.1. Experimental Setup

The experimental setup has been established to implement the proposed method as an identification technique for both channel and source (ECU) identification. Three different type of channels, GXL, TXL, SAE J1939-15, are used for CAN-Bus. These channels are being used actively in real vehicles. Details of the channel types and channel lengths are outlined as follows and technical specification has been provided in table 4.1.1:

- GXL primary automotive cable is used for engine compartment where high resistance is required according to SAE J1128. [52]
- TXL is also primary automotive cable used for applications requiring smaller diameters and minimal weight.
- CAN-bus data cables SAE J1939-15 which is used for connecting different ECUs to network.

Table 4.1.1. The technical specification of three different channel families.

Type	AWG	Conductor	Insulation	No. of Strands	Temperature	Compliances
GXL	18	Bare copper	Cross-linked Polyethylene (XLP)	16x30	-40°C -125°C	Ford ESB-(M1L85-A), Chrysler (MS8900), SAE-J-1128.
TXL	18	Bare copper	Cross-linked Polyethylene (XLP)	19x30	-40°C -125°C	Ford (M1L-123A), Chrysler (MS-8288), SAE-J-1560
CAN-bus Data cable	18	Bare copper	Cross-Linked Polyolefin (XLPO), Thermoplastic Polyurethane (TPU)	19x31	-45°C -125°C	SA J1939-11 Physical Media, RoHS, SAE J1128 performance (fluid, flame propagation)

Six (6) channel lengths are considered to realize CAN-Bus with pairs of twisted wires from same manufacturer and gauge. Overall, the experimental setup contains following hardware and software components:

- Four (4) Arduino Uno R2 microcontroller kits
- Four (4) CAN-Bus shield board with MCP2515 CAN-bus controller and MPC2551 CAN transceiver.
- Three (3) different types of Cables (GXL, TXL, and CAN-bus data cable) with multiple lengths: 0.5 meter, 1 meter, 2 meter, 3 meter, 4 meter, and 5 meter.
- Oscilloscope DSO1012A for the voltage samples recording with Sampling Rate of 2GSa/s, 100MHz bandwidth, and 8-bit vertical resolution.
- Script for sending an identical message continuously from different channels and ECUs to observe the unique patterns of signals from each channel and ECU.

- MATLAB R2016a software for statistical data analysis of sampled signals.

The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button [53]. Figure 4.1.1 shows the Arduino board which is being used as ECU.

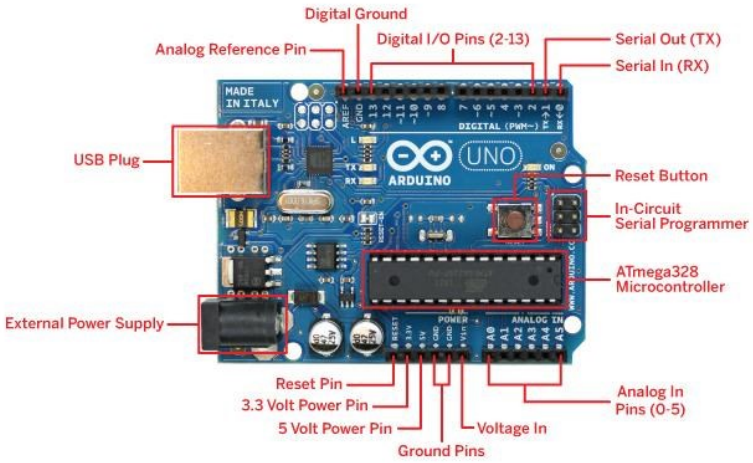

Figure 4.1.1. Arduino Uno Board

In order to establish the CAN-Bus communication in the Arduino board, there is another board available which is connected to the Arduino to make this board as ECU with CAN-Bus communication stack on it. This CAN-BUS Shield adopts **MCP2515** CAN Bus controller with SPI interface and **MCP2551** CAN transceiver to give your Arduino/Seeeduino CAN-BUS capability. With an **OBD-II** converter cable added on and the OBD-II library imported, the Arduino board is capable of performing functionality of on board diagnostics and data logger. It implements the CAN V2.0 B with the baud rate up to 1Mb/s with two screw terminal to easily connect CAN-High and CAN-Low. Figure 4.1.2. shows the CAN-Bus shield which will be attached to the Arduino board. Figure 4.1.3 shows the complete package which acts as ECU for CAN-Bus data communication.

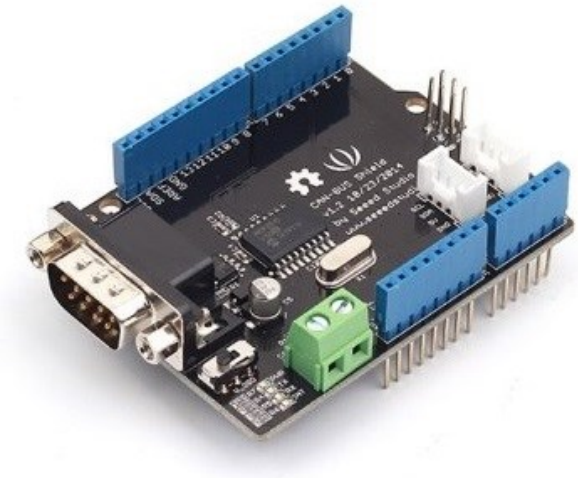

Figure 4.1.2. CAN-Bus shield for Arduino

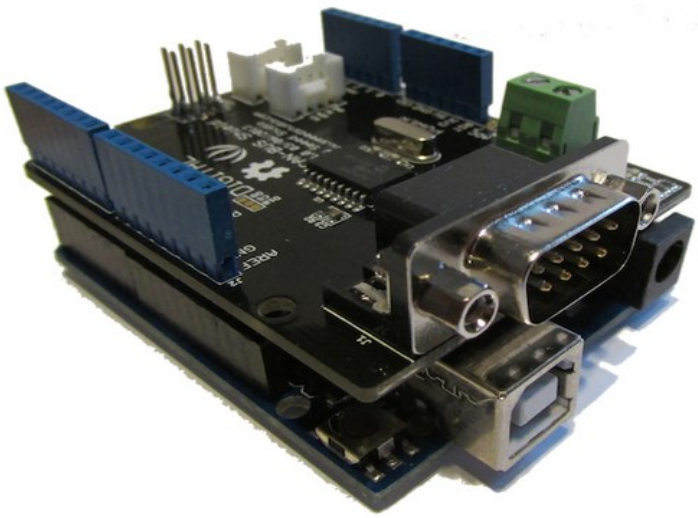

Figure 4.1.3. Arduino Board with attached CAN-Bus shield

For the experimental setup, three major CAN-Bus cables which are used in automotive industry have been employed as communication channel. GXL cable is suitable for use as low voltage primary wire intended for use at nominal system voltages. Type GXL is ideal for high temperature applications with limited exposure to fluids and physical abuse experienced during the operation of cars, boats, trucks, buses, tractors, trailers, etc. TXL wire is an extra-thin wall, stranded, single-conductor automotive primary wire. It is rated to SAE J-1128, Ford (M1L-123A) and Chrysler

(MS-8288) specifications [54]. TXL is also utilized in wiring harnesses in passenger cars and light trucks, agricultural tractors, construction, locomotive and off-the-road vehicles. TXL may be used in automotive applications where small diameter and minimal weight are desirable. It is intended for use at 50 volts or less in surface vehicle electrical systems. OEMs and truck manufacturers continue to add complex and additional functions to on and off-road vehicles. CAN-Bus Data Cable J1939/11 reduces wiring, electronic interference, and offers high-speed network communication. It is resistant to abrasions and cuts, while also has an excellent resistance to oil and chemicals. The SAE J1939/15 is unshielded with no drain and the SAE J1939/11 CAN-Bus cable is shielded with drain wire. The CAN-Bus J1939/11 cable is a suitable channel for sensors and actuators and ECUs. It has also capability of transmitting signals and conduct power even for heavy trucks, buses, and agricultural vehicles namely combines, tractors, and sprayers.

Each ECU in experimental setup has been programmed to send identical message over the CAN-Bus and oscilloscope has been used to gather the transmission signals for data analysis. DSO1002A Oscilloscope with the following specification has been employed for this purpose. It provides 200MHz bandwidth with up to 2GSa/s sample rate, 20kpts memory, powerful triggering (edge, pulse width) with adjustable sensitivity to filter noise and prevent false triggers.

MATLB script has been developed to create an interface between oscilloscope and each ECU in order to capture the transmission signals. Performance of the proposed algorithm is evaluated for both CAN-Bus channel and ECU classification. To this end, physical signal is captured at the output of three different cable families with multiple lengths (0.5 meter, 1 meter, 2 meters, 3 meters, 4 meters, and 5 meters) and twelve identical ECUs with same input CAN-bus message. For this reason, a dataset for the 18 channels and four identical ECUs is collected. For each data collection setting, 144000 (3600 cycles *40 samples) samples are collected for channel

identification. 1 dataset element (1 row) consists of 40 samples of the recorded data (40 samples= 1 cycle of waveform). For performance evaluation, the collected data has divided into training and testing data. (Training set: 65% and Test set: 35%, respectively). Firstly, the training data is used for the ANN to be trained to classify different channels and ECUs and then the tasting data is applied to evaluate the performance of the classifier. The dataset used here is collected in the same environment i.e. under the same temperature and using an identical message to observe the minute and unique variation of the digital signals. The sampled signals were at 500K bit rate which is commonly used in High-speed CAN communication. There are some other bit rates standard available for automotive communication like 100K, 125K, or even 33K bit (GMLAN). Furthermore, MATLAB ANN toolbox is used to implement the classification algorithm both for channel and ECU identification.

4.2. Experimental Results and Discussion

Performance of the proposed method is evaluated through a series of experiments for channel as well as ECU identification. To achieve this goal, a multilayer neural network based classifier is trained on randomly selected 65% data for each channel and ECU. The trained classifier is then employed to test performance of the proposed methods on remaining 35% data. Classification accuracy is used to measure performance of the proposed method. The first experimental evaluation part is allocated to the channel identification.

Experiment 1: Channel Identification

The main objective of this experiment is to validate uniqueness of channel specific features. Material and design imperfections for each specific physical channel is the leading factors behind the channel specific unique artifacts. To validate this claim, data is recorded for each cable family and each channel length with identical channel input, transmitted using the same ECU.

Specifically, for this experiment *'cable type'* and *'length'* are the only variables. During the training phase, the neural network is trained for classifying three different cable family and six corresponding channel lengths (e.g., GXL: 0.5 meter, GXL: 1 meter, GXL: 2 meter, GXL: 3 meter, GXL: 4 meter, and GXL: 5 meter and so on). A multilayer neural network is trained with "scaled conjugate gradient back propagation" training algorithm, 11 inputs variables (time and frequency domain), 6 outputs which corresponds to different lengths of GXL cable, stopping criteria of Epochs = 2000, gradient = 1e-7, and three hidden layers with 50,40, and 40 hidden nodes respectively. Shown in Figure 4.2.1 is the architecture of the multilayer neural network trained for channel classification.

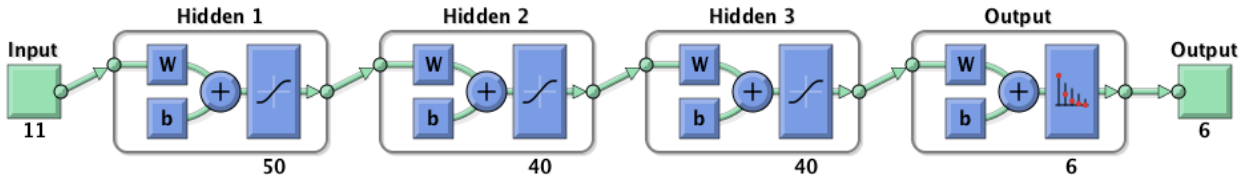

Fig. 4.2.1. Neural Network architecture of channel classifier.

Confusion matrix is a specific table layout that represent the performance of the classification algorithm, typically in the supervised learning environment. Shown in Table 4.2.1 and 4.2.2 are the confusion matrices of the channel (C) classification averaged over all cable types for the training and test phase. It can be observed from aforementioned tables that that the proposed method for channel classification achieves overall correct detection rate of 97.6% and 95.2% for the training and test phase, respectively. It can also be noticed that 0.5 meter and 1 meter channels exhibit relatively higher false rates for both training and testing, these false rates can be attributed to the fact that both channel lengths are not very different. The signal characteristics uniqueness exists for each family type cable and the corresponding lengths.

Table 4.2.1. Training confusion matrix for channel classifier

Predicted Class								
	C1	365 15.6%	4 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.9% 1.1%
	C2	30 1.3%	378 16.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.6% 7.4%
	C3	2 0.1%	0 0.0%	376 16.1%	12 0.5%	0 0.0%	0 0.0%	96.4% 3.6%
	C4	1 0.0%	0 0.0%	8 0.3%	382 16.3%	0 0.0%	0 0.0%	97.7% 2.3%
	C5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	388 16.6%	0 0.0%	100% 0.0%
	C6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	394 16.8%	100% 0.0%
Class label		91.7% 8.3%	99.0% 1.0%	97.9% 2.1%	97.0% 3.0%	100% 0.0%	100% 0.0%	**97.6%** **2.4%**
		C1	C2	C3	C4	C5	C6	
		Target Class						

In above table, the first six diagonal cells represent the number of percentage of the correct classification by ANN during the training phase. For instance, 365 sample points are correctly classified as channel 1 (C1) which corresponds to 15.6% of all data points. In similar, 378 data samples are correctly classified as channel 2 and so on. 4 data points are incorrectly classified as channel 1 which corresponds to 0.2% of all data points. Similarly, 30 data points (1.3%) are incorrectly classified as C2. Out of 369 C1 classification, 98.9% are correct and 1.1% are wrong. The other channels classification results for training phase is represented in the above table a well. Overall, 97.6% of all predictions are correct with only 2.4% of misclassification rate during the training phase.

Table 4.2.2. Test confusion matrix for channel classifier

Predicted Class								
	C1	176 14.0%	10 0.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	94.6% 5.4%
	C2	22 1.7%	205 16.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.3% 9.7%
	C3	3 0.2%	3 0.2%	203 16.3%	9 0.7%	0 0.0%	0 0.0%	93.1% 6.9%
	C4	1 0.1%	0 0.0%	13 1.0%	197 15.6%	0 0.0%	0 0.0%	93.4% 6.6%
	C5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	212 16.8%	0 0.0%	100% 0.0%
	C6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	206 16.3%	100% 0.0%
Class Label		87.1% 12.9%	94.0% 6.0%	94.0% 6.0%	95.6% 4.4%	100% 0.0%	100% 0.0%	**95.2%** **4.8%**
		C1	C2	C3	C4	C5	C6	
		Target Class						

In similar to the training phase the first six diagonal cells represent the number of percentage of the correct classification by ANN during the test phase as well. For instance, 176 sample points are correctly classified as channel 1 (C1) which corresponds to 14% of all data points. In similar, 205 data samples are correctly classified as channel 2 and so on. 10 data points are incorrectly classified as channel 1 which corresponds to 0.8% of all data points. Similarly, 22 data points (1.7%) are incorrectly classified as C2. Out of 186 C1 classification, 94.6% are correct and 5.4% are wrong. The other channels classification results for testing phase is represented in the above table a well. Overall, 95.2% of all predictions are correct with only 4.8% of misclassification rate during the testing phase.

Experiment 2: ECU Identification

The purpose of this experiment is to validate that different ECUs even from the same make and model introduce different artifacts while transmitting an identical message. To achieve this goal, dataset for all four ECUs transmitting same messages over the same channel is used. In this experiment, ECU is the only variable while other variables are kept constant. To this end, data for all four ECUs transmitting same messages over the 2-meter unshielded CAN-Bus data cable is used for training and testing. 2400 rows {2400 cycles, 2400*40 samples} A multilayer neural network classifier is trained with "scaled conjugate gradient back propagation" training algorithm, 11 input variables (both time and frequency domain), 4 outputs which pertains to each ECU, stopping criteria of Epochs = 2000, gradient = 1e-7, and one hidden layer with 20 hidden nodes included. Shown in Figure 4.2.2 is the architecture of the multilayer NN trained for channel classification.

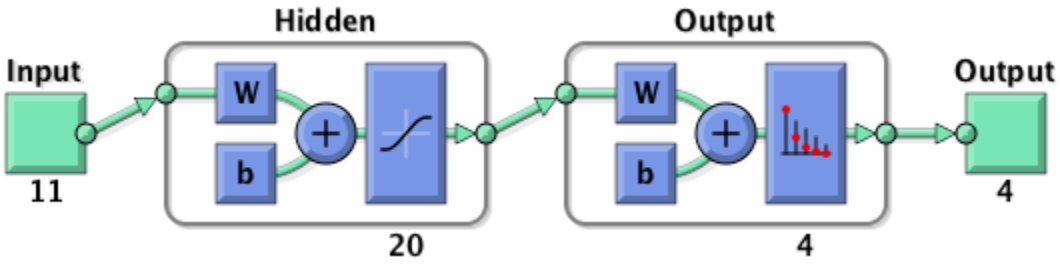

Figure 4.2.2 Neural Network architecture for ECU classification

Table 4.2.3 and Table 4.2.4 summarized the classification performance of the proposed system in terms of confusion matrices of the ECU (E) classification for the training and test phases, respectively. It can be observed from these tables that the proposed method for ECU classification achieves overall success detection rate of 99.6% and 98.3% during the training and test phase, respectively.

Table 4.2.3 Training confusion matrix for ECU classifier

Predicted Class						
	E1	389 24.9%	0 0.0%	3 0.2%	0 0.0%	99.2% 0.8%
	E2	0 0.0%	398 25.5%	0 0.0%	0 0.0%	100% 0.0%
	E3	3 0.2%	0 0.0%	379 24.3%	0 0.0%	99.2% 0.8%
	E4	0 0.0%	0 0.0%	0 0.0%	398 24.9%	100% 0.0%
	Class label	99.2% 0.8%	100.% 0.0%	99.2% 0.8%	100% 0.0%	**99.6%** **0.4%**
		E1	E2	E3	E4	
		Target Class				

Table 4.2.4. Testing confusion matrix for ECU classifier

Predicted Class						
	E1	200 23.8%	0 0.0%	6 0.7%	0 0.0%	97.1% 2.9%
	E2	0 0.0%	202 24.0%	0 0.0%	0 0.0%	100% 0.0%
	E3	7 0.8%	0 0.0%	212 25.2%	0 0.0%	96.8% 3.2%
	E4	1 0.1%	0 0.0%	0 0.0%	212 25.2%	99.5% 0.5%
	Class label	96.2% 3.8%	100.% 0.0%	97.2% 2.8%	100% 0.0%	**98.3%** **1.7%**
		E1	E2	E3	E4	
		Target Class				

CHAPTER 5: Conclusion and Future Work

In this thesis, physical-fingerprinting model is introduced for both channel and ECU identification. It has been demonstrated that for an identical CAN-Bus message, underlying physical channel leaves inimitable characteristic artifacts in the signals at the channel output. These artifacts are unique to different channel lengths and ECUs. The received physical signal therefore can be used for linking received CAN packet to actual transmitter. Statistical attributes in time and frequency domain are utilized for channel and device identification. The performance of the Artificial Neural Network as a classification method is evaluated by carrying out the experimental setup for three different CAN-Bus channels with six multiple lengths (0.5 meter, 1 meter, 2 meter, 3 meter, 4 meter, and 5 meter) and also four ECUs from the same manufacturer. The experimental results and analysis indicate that the proposed method achieves the satisfactory CAN-Bus channel and ECU identification performance with the overall correction rate of 95.2% and 98.3%, respectively. For the future work, development of an identification platform for security purposes will be investigated to determine whether the received message is from the compromised ECU or legitimate one by leveraging these unique signal characteristics. In addition, developing a security solution will be investigated based on CAN-Bus traffic analysis to identify abnormal traffic behavior.

Bibliography

[1] CAN-Bus Specifications Rep. Robert Bosch GmbH. Postfach 50, D-7000. Stuttgart 1Print.

[2] Checkoway, Stephen, et al. "Comprehensive Experimental Analyses of Automotive Attack Surfaces." USENIX Security Symposium. 2011.

[3] J. M. Flores-Arias, M. Ortiz-Lopez, F. J. Quiles-Latorre, V. Pallares and A. Chen, "Complete hardware and software bench for the CAN bus," 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2016, pp. 211-212. doi: 10.1109/ICCE.2016.7430584.

[4] Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaaniche, et al. Survey on security threats and protection mechanisms in embedded automotive networks. 2nd Workshop on Open Resilient Human-aware Cyber-Physical Systems (WORCS-2013).

[5] Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway{with me in it} (2015), http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[6] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy,D., Kantor, B., Anderson, D., Shacham, H. “Experimental Security Analysis of a Modern Automobile”. In: 31st IEEE Symposium on Security & Privacy (S &P 2010), on. pp. 447:462

[7] Kleberger, Pierre, Tomas Olovsson, and Erland Jonsson. "Security Aspects of the In-Vehicle Network in the Connected Car." IEEE Intelligent Vehicles Symposium (2011). Web. 5 June 2011.

[8] Wu, J., LI, Y. B., Li, J., LI, Y. D., YU, H. Y., & SONG, L. M. (2009). CAN bus of automotive driving force control system based on SAE protocol J1939 [J]. *Journal of Jilin University (Engineering and Technology Edition)*, *4*, 005.

[9] Wang, J., Wang, X., Zhai, X., Wang, H., & Lampe, B. (2005, October). CAN/LIN hybrid network for automobile. In *Vehicular Electronics and Safety, 2005. IEEE International Conference on* (pp. 348-352). IEEE.

[10] Lee, M. Y., Chung, S. M., & Jin, H. W. (2010, January). Automotive network gateway to control electronic units through MOST network. In *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*(pp. 309-310). IEEE.

[11] Kaiser, J., & Mock, M. (1999). Implementing the real-time publisher/subscriber model on the controller area network (CAN). In Object-Oriented Real-Time Distributed Computing, 1999.(ISORC'99) Proceedings. 2nd IEEE International Symposium on (pp. 172-181). IEEE.

[12] Hafeez, A., Malik, H., Avatefipour, O., Rongali, P. et al., "Comparative Study of CAN-Bus and FlexRay Protocols for In-Vehicle Communication," SAE Technical Paper 2017-01-0017, 2017.

[13] Ran, P., Wang, B., & Wang, W. (2008, April). The design of communication convertor based on CAN bus. In Industrial Technology, 2008. ICIT 2008. IEEE International Conference on (pp. 1-5). IEEE.

[14] K. Zdeněk and S. Jiří, "Simulation of CAN bus physical layer using SPICE," 2013 International Conference on Applied Electronics, Pilsen, 2013, pp. 1-4.

[15] Farsi, M., Ratcliff, K., & Barbosa, M. (1999). An overview of controller area network. *Computing & Control Engineering Journal*, *10*(3), 113-120.

[16] Nolte, T., Hansson, H., Norström, C., & Punnekkat, S. (2001, December). Using bit-stuffing distributions in CAN analysis. In *IEEE Real-Time Embedded Systems Workshop at the Real-Time Systems Symposium*.

[17] Shavit, M., Gryc, A., & Miucic, R. (2007). Firmware update over the air (FOTA) for automotive industry (No. 2007-01-3523). SAE Technical Paper.

[18] Tindell, K., & Burns, A. (1994, September). Guaranteeing message latencies on control area network (CAN). In Proceedings of the 1st International CAN Conference. Citeseer.

[19] Stallings, W., & Tahiliani, M. P. (2014). Cryptography and network security: principles and practice (Vol. 6). London: Pearson.

[20] Nilsson, D. K., Larson, U. E., Picasso, F., & Jonsson, E. (2009). A first simulation of attacks in the automotive network communications protocol flexray. In Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08 (pp. 84-91). Springer, Berlin, Heidelberg.

[21] Lin, C. W., & Sangiovanni-Vincentelli, A. (2012, December). Cyber-security for the Controller Area Network (CAN) communication protocol. In Cyber Security (CyberSecurity), 2012 International Conference on (pp. 1-7). IEEE.

[22] You, S., Krage, M., & Jalics, L. (2005). *Overview of remote diagnosis and maintenance for automotive systems* (No. 2005-01-1428). SAE Technical Paper.

[23] Theissler, A. (2014). Anomaly detection in recordings from in-vehicle net-works. *BIG DATA AND APPLICATIONS*, 23.

[24] Khalilian, A., Sahamijoo, G., Avatefipour, O., Piltan, F., & Nasrabad, M. R. S. (2014). Design high efficiency-minimum rule base PID like fuzzy computed torque controller. International Journal of Information Technology and Computer Science (IJITCS), 6(7), 77.

[25] Khalilian, A., Piltan, F., Avatefipour, O., Nasrabad, M. R. S., & Sahamijoo, G. (2014). Design New Online Tuning Intelligent Chattering Free Fuzzy Compensator. International Journal of Intelligent Systems and Applications, 6(9), 75.

[26] Sahamijoo, G., Avatefipour, O., Nasrabad, M. R. S., Taghavi, M., & Piltan, F. (2015). Research on minimum intelligent unit for flexible robot. International Journal of Advanced Science and Technology, 80, 79-104.

[27] Sinclair, C., Pierce, L., & Matzner, S. (1999). An application of machine learning to network intrusion detection. In Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual (pp. 371-377). IEEE.

[28] Avatefipour, O., Hafeez, A., Tayyab, M., & Malik, H. (2017). Linking Received Packet to the Transmitter Through Physical-Fingerprinting of Controller Area Network . Information Forensics and Security (WIFS Conference, Rennes, France).

[29] Shon, T., Kim, Y., Lee, C., & Moon, J. (2005, June). A machine learning framework for network anomaly detection using SVM and GA. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC* (pp. 176-183). IEEE.

[30] Mokhtar, M., Piltan, F., Mirshekari, M., Khalilian, A., & Avatefipour, O. (2014). Design minimum rule-base fuzzy inference nonlinear controller for second order nonlinear system. International Journal of Intelligent Systems and Applications, 6(7), 79.

[31] Avatefipour, O., Piltan, F., Nasrabad, M. R. S., Sahamijoo, G., & Khalilian, A. (2014). Design New Robust Self Tuning Fuzzy Backstopping Methodology. International Journal of Information Engineering and Electronic Business, 6(1), 49.

[32] Shahcheraghi, A., Piltan, F., Mokhtar, M., Avatefipour, O., & Khalilian, A. (2014). Design a Novel SISO Off-line Tuning of Modified PID Fuzzy Sliding Mode Controller. International Journal of Information Technology and Computer Science (IJITCS), 6(2), 72.

[33] Ramadan, M. N., Al-Khedher, M. A., & Al-Kheder, S. A. (2012). Intelligent anti-theft and tracking system for automobiles. International Journal of Machine Learning and Computing, 2(1), 83.

[34] Müter, M., & Asaj, N. (2011, June). Entropy-based anomaly detection for in-vehicle networks. In Intelligent Vehicles Symposium (IV), 2011 IEEE (pp. 1110-1115). IEEE.

[35] Müter, M., Groll, A., & Freiling, F. C. (2010, August). A structured approach to anomaly detection for in-vehicle networks. In Information Assurance and Security (IAS), 2010 Sixth International Conference on (pp. 92-98). IEEE.

[36] Narayanan, S. N., Mittal, S., & Joshi, A. (2015). Using data analytics to detect anomalous states in vehicles. arXiv preprint arXiv:1512.08048.

[37] Cho, Kyong-Tak, and Kang G. Shin. "Fingerprinting electronic control units for vehicle intrusion detection." 25th USENIX Security Symposium (USENIX Security 16). USENIX Association, 2016.

[38] Murvay, Pal-Stefan, and Bogdan Groza. "Source identification using signal characteristics in controller area networks." IEEE Signal Processing Letters 21.4 (2014): 395-399.

[39] Q. Wang and S. Sawhney, "VeCure: A practical security framework to protect the CAN bus of vehicles," *2014 International Conference on the Internet of Things (IOT)*, Cambridge, MA, 2014, pp.13-18.doi: 10.1109/IOT.2014.7030108

[40] Experimental security analysis of a modern automobile K Koscher, A Czeskis, F Roesner, S Patel, T Kohno - 2010 IEEE Symposium on Security and Privacy, 2010

[41] Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh and M. Manzuri Shalmani. On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme. In D. Wagner, editor, Proceedings of Crypto 2008, volume 5157 of LNCS, pages 203–20. Springer-Verlag, Aug. 2008.

[42] Hoppe, T., Kiltz, S., & Dittmann, J. (2008). Security threats to automotive CAN networks–practical examples and selected short-term countermeasures. *Computer Safety, Reliability, and Security*, 235-248.

[43] Ueda Hiroshi, Ryo Kurachi, Hiroaki Takada, Tomohiro Mizutani, Masayuki Inoue, and Satoshi Horihata. "Security Authentication System for In-Vehicle Network." SEI Technical Review 81 (2015).

[44] Hazem, A., Fahmy, H.A.: LCAP – "A Lightweight CAN Authentication Protocol for securing in-vehicle networks." In: 10th Int. Conf. on Embedded Security in Cars (ESCAR 2012), Berlin, Germany. vol. 6 (2012).

[45] Perrig, A., Canetti, R., Tygar, J. D., & Song, D. (2005). The TESLA broadcast authentication protocol. *Rsa Cryptobytes*, *5*.

[46] Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (Eds.). (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

[47] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[48] Guyon, I., & Elisseeff, A. (2006). An introduction to feature extraction. *Feature extraction*, 1-25.

[49] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi,"Accelprint: Imperfections of accelerometers make smartphones trackable," in 21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26,2014, 2014. [Online]. Available: http://www.internetsociety.org/doc/accelprint-imperfections-accelerometers-make-smartphones-trackable

[50] "LibXtract: Feature Extraction Library Documentation," http://jamiebullock.github.io/LibXtract/documentation

[51] Brown, G., Pocock, A., Zhao, M. J., & Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, *13*(Jan), 27

[52] SAE J1128Standard - Low Voltage Primary Cable . (2013, September 9). SAE Ineternational: http://standards.sae.org/j1128_201310/

[53] Badamasi, Y. A. (2014, September). The working principle of an Arduino. In *Electronics, Computer and Computation (ICECCO), 2014 11th International Conference on* (pp. 1-4). IEEE.

[54] Tonyali, K. (1995). U.S. Patent No. 5,401,787. Washington, DC: U.S. Patent and Trademark Office.

[55] http://electric-cloud.com/blog/2014/12/continuous-delivery-puts-automotive-software-high-gear/