An Integrated Approach to Securing In-Vehicle CAN Bus Network Using ECU Fingerprinting and Image Classification Techniques

by

Janani Mohan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Science
(Computer and Information Science)
in the University of Michigan-Dearborn
2023

Master's Thesis Committee:

Professor Selim S. Awad, Co-Chair
Lecturer Azeem Hafeez, Co-Chair
Associate Professor Xuan Zhou

Janani Mohan

janmohan@umich.edu

ORCID iD: 0009-0009-5503-1967

DEDICATION

This is humbly dedicated to those who have been unwavering in their support and encouragement throughout my academic odyssey.

ACKNOWLEDGEMENTS

Completing my master's degree was an immensely fulfilling experience that has been pivotal in shaping my academic journey. The success of this thesis owes much to the guidance and support of my esteemed supervisor, Sir Azeem Hafeez, whose invaluable insights, constructive feedback, and encouragement helped me navigate the rigors of academic research. I am also deeply grateful to Mansi Girdhar, who provided me with critical proofreading support that helped refine the quality of this work. My academic advisor, Kimberly LaPere, deserves special recognition for her unwavering support, timely guidance, and diligent follow-ups throughout my academic journey.

I would also like to express my sincere appreciation to my esteemed professor, Selim Awad, for his invaluable mentorship and trust in my abilities. My family, especially my father, Mohan Gangadharan, and mother, Jayanthi Mohan, have been an unwavering source of support and encouragement, without which this achievement would not have been possible. I would like to dedicate this work to my brother, Karthikeyan Mohan, whose constant support and motivation have been a driving force throughout my academic career.

Completing this thesis has been an enriching experience, and I am grateful to everyone who has been a part of this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADAS Advanced Driver Assistance Systems

AV Autonomous Vehicles

CAN Controller Area Network

CANH CAN High

CANL CAN Low

CNN Convolutional Neural Networks

ECU Electronic Control Unit

HIL Hardware-in-Loop

K-NN k Nearest Neighbor

LR Logistic Regression

ML Machine Learning

PCA Principal Component Analysis

SVN Support Vector Machine

VRMS Root-Mean-Square Voltage

ABSTRACT

This paper presents two novel techniques for securing electronic control units (ECUs) in the controller area network (CAN) bus network of autonomous vehicles (AVs).

Method 1 proposes an ECU fingerprinting technique to detect malicious ECUs in the in-vehicle CAN bus network. The technique extracts unique digital signatures based on intrinsic characteristics of the ECUs to identify the ECU responsible for broadcasting counterfeit messages received on the CAN bus. The proposed work employs three machine learning (ML) algorithms, namely k-Nearest Neighbors (k-NN), Support Vector Machine (SVM), and Logistic Regression (LR), to analyze the data from seven distinct ECUs. The performance of the proposed cybersecurity framework is evaluated and compared using these algorithms.

Method 2 proposes a solution for efficient ECU classification to protect against multiple threats and attacks, including hacking and spoofing attacks, that can be perilous for the vehicle and its occupants. This technique visualizes signal loss and distortion in the ECU voltage signals caused by the ECU position in the CAN bus and utilizes Euclidean distance-based image classification on the principal components. This methodology is based on the commercially feasible eigenface-based algorithm, which has found extensive application in real-time scenarios like face recognition, fingerprint recognition, image recognition in photo-based applications, and real-time object identification. By profiling the ECU using this method, we can make the signal analyzer commercially viable. In this paper, we analyze different ECU configurations in the daisy chain network to evaluate the effectiveness of our proposed method. Our approach achieves a high accuracy rate of 97.14%.

The proposed techniques address the security concerns related to the CAN bus network of AVs and provide efficient and effective ways to secure ECUs against malicious activities. The use of machine learning algorithms and visualization techniques in these methods not only enhances the accuracy of ECU detection and classification but also provides a better understanding of the underlying data. These techniques can be implemented in AVs to improve the security of the CAN bus network and ensure safe and secure operation of the vehicles.

CHAPTER 1

Introduction

Automation and the application of artificial intelligence are modernizing the automotive industry and changing the way people perceive it. The ECU (Electronic Control Unit) is pivotal in facilitating vehicle automation such as elevating the user experience by monitoring and making dynamic decisions based on the sensor data for ADAS. Cars that offer semi-autonomous driving features like Tesla Autopilot hinge heavily on the signal processing of the ECU. Automotive researchers are experimenting with vehicle-to-vehicle interactions, and ECUs will play an additional role in communicating with the nearest ECUs during transit. This versatile component is subjected to a multitude of threats and attacks, such as remote hacking into a Jeep Cherokee's ECU in 2015 and the 2017 Tesla Model S ECU breach using a physical custom-designed system. These threats pose a perilous situation for the vehicle and its occupants. In a wave of the increasingly connected and automated automobile era, it is paramount to ensure that ECUs are resilient to hacking and spoofing attacks.

As the automotive industry progresses toward extensive robotization, the use of diverse sensors and actuators is becoming increasingly common. These sensors and various computation units are controlled by embedded ECUs, which are integrated and designed for optimizing a wide variety of functions. Modern electric vehicles contain hundreds of ECUs, and this number is expected to increase in the coming years. CAN is used as a legacy standard protocol for in-vehicle communication, owing to its reliability, robustness, and simplicity.

However, despite the many built-in functional safety features, the unencrypted nature of CAN messages and lack of authentication of message sources render the CAN network vulnerable to multiple cyber-attacks, such as spoofing and modification attacks. These attacks can cause severe implications, such as data breaches and jeopardizing the safety and security concerns of the vehicle industry. The CAN data link connects multiple ECUs together as nodes to send or receive messages, enabling engine operations. It consists of two wires twisted as a pair, namely CAN high and CAN low, and is terminated with a resistor on each end. The CAN bus can have one of two logic states, logical or recessive, where a logical 0 corresponds to a dominant bus level, and a logical 1 is termed as the recessive level. When the bus is idle, i.e., when there is no transmission of ECU information, the voltage level on the bus is recessive (2.5V), and once a message is transmitted, it goes into a dominant state (3.5V). This whole system works as a multi-master system, where every device within the system sends or receives information. However, only a single device or ECU is allowed to send a message at a given time.

The vulnerability of the CAN bus to cyber-attacks is a critical concern for the automotive industry. Several research studies have proposed a range of preventive methods such as message authentication-based approaches, which implement security at the data link layer, and intrusion detection-based approaches, which implement security at the physical layer. However, these approaches have limitations, and there is a need for a more intelligent solution for ECU identification.

The primary contribution of this thesis is to develop a cybersecurity model called ECU fingerprinting to solve the problem of identifying the source ECU of a CAN packet transmitted over the communication channel. The proposed method exploits the undesirable material and design defects in both the physical communication medium and ECUs to relate the received signal to the ECU responsible for sending it. Generally, the distinctive or unique patterns of the transmitted ECU signals over time are explicitly used as digital signatures for the proposed method. Feature sets comprising time and frequency domain-based physical signal attributes are employed to train machine learning-based classification algorithms to determine the malicious ECU.The significance of the proposed ECU fingerprinting model is to improve the CAN efficiency by identifying the malicious CAN packet and its sender node.

CHAPTER 2

Machine Learning Based ECU Detection for Automotive Security

2.1 Background and Motivation

The automotive industry has progressed significantly over the recent years into extensive robotization by using diverse sensors and actuators. These sensors and various computation units are controlled by embedded ECUs which are integrated and designed for optimizing of a wide variety of functions. There are hundreds of ECUs fitted in the modern electric vehicles, and this number is anticipated to amplify in the subsequent years. CAN is used as a legacy standard protocol, owing to its reliability, robustness and simplicity for in-vehicle communication. However, despite many built-in functional safety features, unencrypted nature of the CAN messages and lack of authentication of message sources render CAN network vulnerable to multiple cyber-attacks, e.g., spoofing and modification attacks [1]. As a result, these attacks can cause severe implications, for instance, data breach, and jeopardize the safety and security concerns of the vehicle industry[2].

As shown in Fig. 2.1, CAN data link connects multiple ECUs together as nodes to send or receive messages, enabling engine operations. It consists of two wires twisted as a pair, namely CAN high and CAN low, and is terminated with a resistor on each end. CAN bus can have one of the two logic states, logical or recessive, where a logical 0 corresponds to dominant bus level, and a logical 1 is termed as the recessive level. When the bus is idle, i.e., when there is no transmission of ECU information, voltage level on the bus is recessive (2.5V), and once a message is transmitted it goes into dominant state (3.5V) [3]. This whole system works as a multi-master system, where every device within the system sends or receives information [4]. However, only a single device or ECU is allowed to send a message at a given time.

There has been an extensive research carried out in seeking the possible vulnerabilities, detection and mitigation of the communication attacks on CAN bus. Literature has proposed a number of preventive methods like, message authentication based approaches which implement security at data link layer [5, 6, 7, 8, 9, 10, 11, 12, 13], and intrusion detection based approaches which implement security at physical layer [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36].

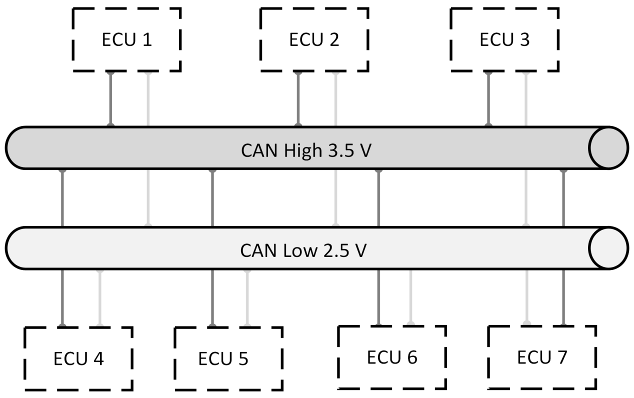


Figure 2.1: ECUs connected across serial CAN Bus.

However, due to the inefficiency of the data encryption based methodologies, an intelligent solution of ECU identification is considered. In a CAN network, the source ECU of a packet transmitted over the communication channel is possibly unmapped[4]. Hence, in case of abnormal behaviors, it is critical to associate malicious CAN packet to its sender node to improve the CAN efficiency, which is a grueling task. The primary contribution of this paper is to solve this problem by developing a cybersecurity model called ECU fingerprinting. Undesirable material and design defects, both in the physical communication medium and ECUs, are exploited to relate the received signal to the ECU responsible for sending it. Generally, the distinctive or unique patterns of the transmitted ECU signals over the course of time are explicitly used as digital signatures for the proposed method. Feature set comprising time and frequency domain based physical signal attributes, are employed to train ML-based classification algorithms to determine the malicious ECU. In the end, performance of the proposed ECU fingerprinting model is analyzed by testing a dataset accumulated from seven different ECUs.

The remainder of the paper is organized as: Section II discusses experimental set up for the proposed approach. Section III the data acquisition and pre-processing stages of the ML algorithms, followed by explaining the feature set comprising of peak-to-peak voltage, mean distortion, percentage overshoot, etc. in section IV. Further, section V plots the probability density functions of all the features listed in previous section. Section VI and section VII outline different ML algorithms which are used for classifying the ECU data in this paper and feature separation respectively. Finally, the paper illustrates the experimental results in section VIII.

2.2 Hardware-In-The-Loop (HIL) Experimental Set Up

An HIL platform is set up to develop and test the proposed approach. In the propounded ML technique, channel variability is evaluated to identify a specific ECU for message authentication. The experiment uses seven channels governed by Society of Automotive Engineers (SAE) and

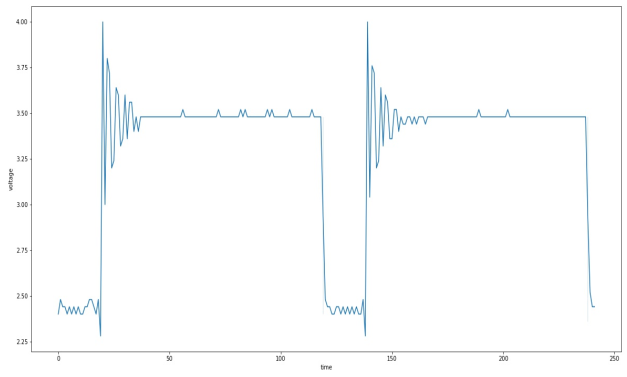


Figure 2.2: Square waves of original data from ECU 1.

CANH pin is used to record the data. The hardware uses seven Arduino UNO-R2 micro-controller kits, seven CAN-Bus shield boards with MCP 2515 CAN-bus controllers, MPC2551 high speed CAN transceivers, and PicoScope 6 Beta tool using a 2 GSa/s maximum real-time sampling rate and 100 MHz bandwidth to capture the measured voltage signals. An interactive software called Jupyter Notebook 6.4.0 is further employed for the study of the captured voltage samples.

2.3 Data Acquisition AND Pre-Processing

First and the foremost step of the ML algorithm is data acquisition, in which the data is accumulated and imported to the specific ML technique. This paper uses a dataset of CAN message packets from seven distinct ECUs, which is recorded using PicoScope. There are seven CAN channels used for transmitting each ECU data. Each ECU has 30 data files comprising of both the dominant and recessive bit voltages and 789 records in total comprising the dataset. The collected data often contains some invalid or inaccurate data points, so data pre-processing is done to clean or remove the incomplete or corrupted data from the dataset. Therefore, the complete ECU data is cleaned to get individual square waves from the distorted waveforms. The graphical plot shown in 2.2 illustrates the result of the data cleaning. After data cleaning, 674 sample records are linked to their appropriate ECU classes. Further, this entire dataset is split into training data (70%) and testing data (30%) for model evaluation. This individual square wave data is further used for feature extraction.

2.4 Feature Set

A collection of control theory and signal processing parameters are considered as the promenient features for the next stage, which is feature extraction. Moreover, feature scaling is done by using min-max normalization, where the features are bound to be in the fixed range of [0,1] to increase the convergence speed. The features utilized for the paper include:

- Peak-to-peak voltage
- Mean distortion
- Percentage overshoot
- Variance
- Volt Root Mean Square (VRMS)
- Peakedness
- Total Energy

2.4.1 Peak-to-Peak Voltage

Peak-to-peak voltage (V_{pp}) is the distance from the lowest amplitude, or trough (V_{min}) [37], to the highest positive amplitude, or the crest (V_{max}) as defined in equation 2.1.In the dataset, it is observed that the lowest and highest voltages of CAN for each ECU are unique. Hence, V_{pp} must be separable as shown in Fig. 2.3, where peak-to-peak voltages are plotted across multiple ECUs.

$$V_{pp} = V_{max} - V_{min} \tag{2.1}$$

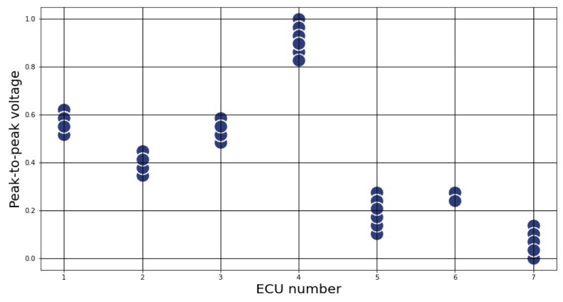


Figure 2.3: Peak-to-peak voltage plotted across ECUs.

2.4.2 Mean Distortion

Distortion is a common problem in signals transmitted through communication channels. It might occur due to both impurities in the medium or any inimitable features of the signals [38]. In the proposed work, each ECU shows a specific pattern of distortion, which is calculated by getting the difference between the actual and the expected value of each voltage. Further, mean of this distortion is calculated for each square wave, dominant bit and recessive bit. For n number of values, the $MeanDistortion$ is computed as,

$$MeanDistortion = \sum_{i=1}^{n} \frac{Actual_(i) - Expected_(i)}{n} \tag{2.2}$$

The graph in Fig. 2.4 shows the mean distortion plotted across the seven ECUs. Many outliers are observed and hence, a good accuracy can be achieved even without this feature. Also ECU 1, ECU 2, ECU 4, and ECU 5 show visible distinction in ranges. The outliers are observed in ECU 1, ECU 2,ECU 3, ECU 4, and ECU 6.

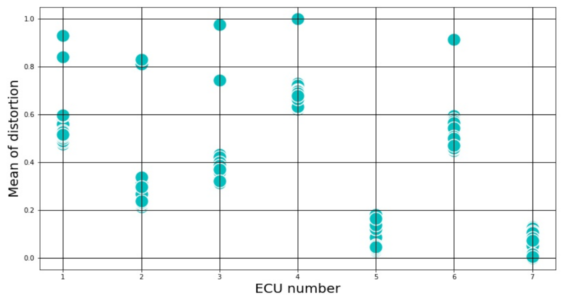


Figure 2.4: Mean of distortion plotted across ECUs.

2.4.3 Percentage Overshoot

Overshoot, also known as percentage of the steady-state value is defined as the maximum amount by which the response overshoots the final, steady-state value [39]. It is used to calculate the step rise in a waveform. Fig. 2.5 plots percentage overshoot across the seven ECUs. It shows a distinct separation between all the ECUs, except ECU 1 and ECU 3. The steady state voltage V_{ss}, is when the voltage reaches the final values and remains steady. It is equated as,

$$\%Overshoot = \frac{V_{pp} - V_{ss}}{V_{ss}} \tag{2.3}$$

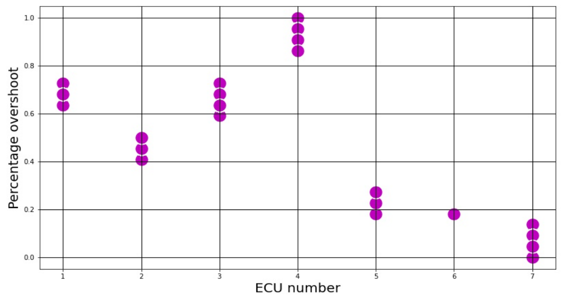


Figure 2.5: Percentage overshoot plotted across ECUs.

2.4.4 Variance

The graph in Fig. 2.6 plots variance of the voltage across the seven ECUs. By the observation, there is a complete overlap between ECU 2 and ECU 3 and with the combined feature set, they can be distinguished, and complete distinct range can be observed between ECU 1, ECU 3, ECU 4 and ECU 5. In general, variance is the statistical measurement of the spread between different values in a given dataset [40]. Variance for X_i term is calculated as,

$$Variance = \sum_{i=1}^{N} \frac{(X_i - M)^2}{N^2} \tag{2.4}$$

Where N is sample size, and M is mean.

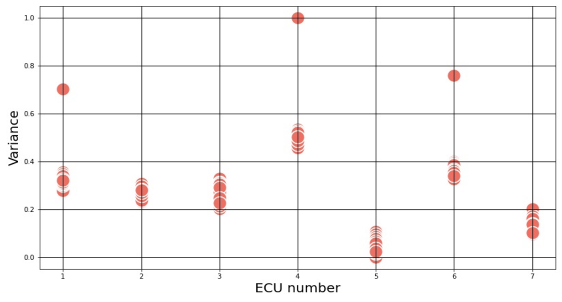


Figure 2.6: Variance of the voltage plotted across ECUs.

2.4.5 Volt Root Mean Square (VRMS)

The graph in Fig. 2.7 plots VRMS values across the seven ECUs. It is observed that there is a complete overlap between ECU 1 and ECU 3. Also with a combined feature set, they can be distinguished, and a complete distinct range can be observed between ECU 1, ECU 2, ECU 4, ECU 5 and ECU 7. VRMS voltage (V_{RMS}) is calculated by dividing peak voltage (V_{pp}) by square root of 2 as shown [41] ,

$$V_{RMS} = \frac{V_{pp}}{\sqrt{2}} \tag{2.5}$$

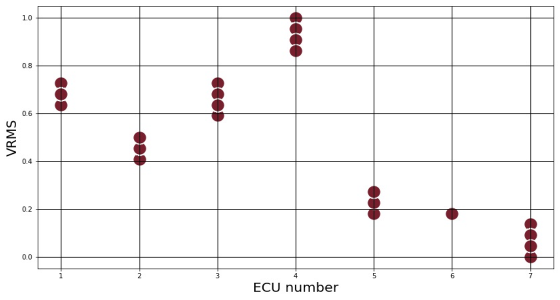


Figure 2.7: VRMS plotted across ECUs.

2.4.6 Peakedness

Since the voltage values of a single ECU are closely related and show similar patterns, so another parameter called peakedness can be used to determine the extent to which voltage values are concentrated around the central value, i.e., mean of the distribution [42]. Peakedness for X_i term is calculated as,

$$Peakedness = \frac{\sum_{i=1}^{N} \frac{X_i - M}{N}}{S^4} \tag{2.6}$$

Where N is sample size, M is mean, and S is standard deviation. The graph in Fig. 2.8 plots peakedness values across the seven ECUs. It states that there is a partial overlap between ECU 3 and ECU 4 and with the combined feature set, they can be distinguished. In addition, a complete distinct range can be observed between ECU 1, ECU 2 and ECU 7.

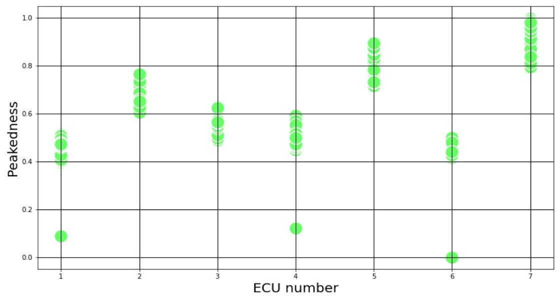


Figure 2.8: Peakedness plotted across ECUs.

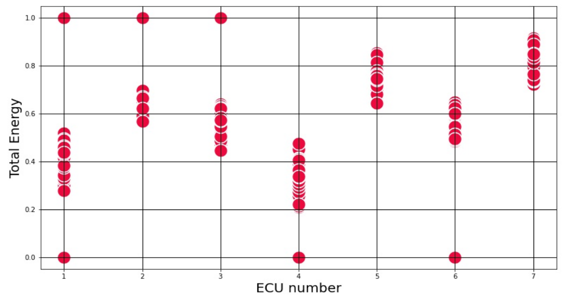


Figure 2.9: Total energy plotted across multiple ECUs.

2.4.7 Total Energy

Fig. 2.9 indicates the total energy plotted across the seven ECUs. Total area under the curve gives the total energy by that waveform [43]. By looking at the graph, it is observed that that there is a partial overlap between ECU 2 and ECU 3 and with the combined feature set, they can be distinguished. Further, a complete distinct range might be observed between ECU 1 and ECU 5.

2.5 Density Of The Features

Density plots shown in Fig. 2.10, visualize the probability density functions of all the features, using kernel density estimation [44]. The features' density values are plotted separately and the point with higher peak is the region with maximum data points occupying between those values. By this plot, separation of ECUs with different features can be visualized.

2.6 Machine Learning Approaches

A ML algorithm is a dynamic mathematical tool to enable immunity from malicious attacks in the vehicular networks. For this paper, several intelligent algorithms are implemented using sklearn library in Python.

2.6.1 k-Nearest Neighbour

k-NN, a non-parametric, supervised algorithm, is used to classify a data point into its available classes. As the name goes, the new data point is placed in the training dataset, followed by the computation of distance metrics, e.g., Euclidean distance to evaluate the correlation between each of the training sample point and the new data point [45].

In the proposed classification algorithm, Euclidean distance (d_E) is the accepted default measure for classification of feature points. It is the geometric distance between two points in a domain, (i) an unknown point (pt1), (ii) another point from the example training set (pt2). Let n is the sample size

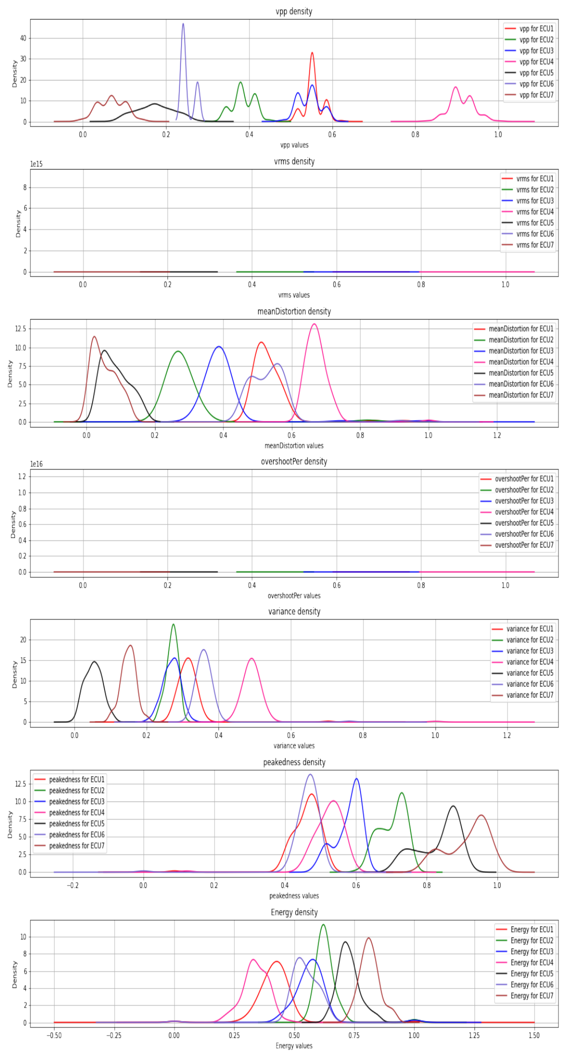


Figure 2.10: Densities of features for the specific ECUs.

in k-dimensional feature space, then (d_E) is measured using Pythagoras theorem applied between the two vectors,

$$d_E\left(pt1_i, pt2_i\right) = \sqrt{\Sigma_{i=1}^{n}\left(pt1_i - pt2_i\right)^2} \tag{2.7}$$

A more generalized formula is used when the training set has both continuous and discrete attributes,

$$d_E\left(pt1_i, pt2_i\right) = \sqrt{\Sigma_{i=1}^{n} d_E\left(pt1_i, pt2_i\right)} \tag{2.8}$$

Further, d_E between the given test $(pt1)$ and training data point $(pt2)$ is formulated as,

$$d_E\left(pt1_i, pt2_i\right) = \sum_{s=1}^{n} \left|pt1_{is} - pt2_{is}\right| \tag{2.9}$$

Another distance metric called as Hamming distance (d_H) is the direct difference between the Boolean attribute values between an example training set point and the unknown vector [46]. If $set_1 = (t, t, t, f)$ and $set_2 = (f, f, t, f)$, then $d_H(set_1, set_2) = 2$. The value of k is chosen as an odd number, i.e., 5, 7, 9.

There is one other method of measuring the distance and it is called Minkowski distance (d_M) [47] which is defined as ,

$$d_M\left(pt1_i, pt2_i\right) = \left(\sum_{s=1}^{n} \left|pt1_{is} - pt2_{is}\right|^q\right)^{1/q} \tag{2.10}$$

Hyper-parameter (k) is an important parameter in the classification algorithm as it might cause under-fitting or over-fitting of the values. For example, if it is too small, then the results might not be accurate, or if the value is too large, then too many neighbors might result in mis-classification of the sample point. Therefore, weighted k-NN is used to avert the undesired results. In weighted k-NN, weight is assigned to the 'k' nearest samples using linear kernel function [37], which is defined as,

$$K(d) = \frac{1}{2} \cdot 1(|d| \leq 1) \tag{2.11}$$

Where K(d) is rectangular kernel function, and d is the distance metric.

The classifier estimates the ECU class based on the nearest neighbors and their votes. In this method, a unique separation between the class of ECUs is observed, and hence a high accuracy is obtained.

2.6.2 Support Vector Machine

SVM is an important statistical supervised learning algorithm that implements both classification and regression analysis. It works on the concept of finding an optimal hyper-plane, also called a decision boundary which linearly separates members of one class from the other by using support vectors and margins [48]. The hyper-plane corresponds to the the maximum margin, i.e., maximum distance between data points of the classes in the hyper-space. SVM is solely based on data points or support vectors nearest to the hyper-plane.

A hyper-plane can be optimized by maximizing the distance between the two distinct support vector clusters, also called street width. Maximum street width (W) can be computed as the below equation, as the boundary of class 1 is x_1 and the boundary of class 2 is x_2 .

$$\text{W} \; = \hat{a\omega} \cdot (x_1 - x_2) \tag{2.12}$$

Since,

$$\hat{a\omega} = \frac{\vec{\omega}}{|\omega|} \tag{2.13}$$

So,

$$\text{W} \; = \frac{\vec{\omega}}{|\omega|} \cdot (x_1 - x_2) \tag{2.14}$$

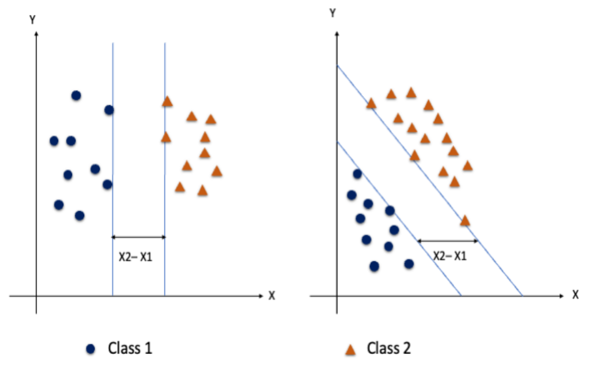


Figure 2.11: Datapoints separation using hyperplane.

Further, decision rule (y_i) is defined to determine on which side of the boundary line, an unknown sample lies. For example, when sample points lie on the boundary line, y_i is,

$$y_i(\omega \cdot x_i + b) - 1 = 0 \tag{2.15}$$

For class 1 (y_i = $+ve$), i.e., when the sample points are on right of the boundary line, decision rule is given by,

$$\vec{w} - x_1 = 1 - b \tag{2.16}$$

For class 2 (y_i = $-ve$), i.e., when the sample points are on left of the boundary line, decision rule is given by,

$$\vec{w} \cdot \vec{x}_2 = -1 - b \tag{2.17}$$

Therefore, street width (W) can be computed as,

$$\text{W} = \frac{\vec{\omega}x_1 - \vec{w}x_2}{|\omega|} \tag{2.18}$$

Also it can be represented as,

$$\text{W} = \frac{1 - b - (-1 - b)}{|\omega|} \tag{2.19}$$

or

$$\text{W} = \frac{2}{|\omega|} \tag{2.20}$$

Now, maximum W can be computed either by maximizing $2/|\omega|$ or by minimizing $|\omega|/2$ or $\frac{1}{2}\|\omega\|^2$. However, minimizing $\frac{1}{2}\|\omega\|^2$ is adopted, since it is the most mathematically convenient method. Further, a simpler Lagrange multiplier methodology is used to compute it.

Let's apply the method of Lagrange multiplier (L),

$$L = \frac{1}{2}\|\omega\|^2 - \sum \alpha_i \left[(y_i \left(\vec{w}_z\vec{x}_i + b\right) - 1\right] \tag{2.21}$$

$$\frac{\partial L}{\partial \vec{w}} = \vec{\omega} - \sum \alpha_i y_i x_i = 0 \tag{2.22}$$

$$\vec{\omega} = \varepsilon \alpha_i y_i x_i \tag{2.23}$$

It is evident that, $\vec{w}$ is a linear sum of support vectors.

Now, calculating $\partial L/\partial b$,

$$\frac{\partial L}{\partial b} = -\varepsilon \alpha_i y_i = 0 \tag{2.24}$$

$$\sum \alpha_i y_i = 0 \tag{2.25}$$

$$L = \frac{1}{2}\|\omega\|^2 - \sum_i \alpha_i \left[y_i \left(\vec{w}x_i + b\right) - 1\right] \tag{2.26}$$

$$\begin{aligned} L = \tfrac{1}{2} \textstyle\sum_i \alpha_i y_i x_i \sum_j \alpha_j y_j x_j - \\ \textstyle\sum_i \alpha_i y_i x_i \cdot \sum \alpha_j y_i x_j - \sum \alpha_i y_i b + \varepsilon \alpha_i \end{aligned} \tag{2.27}$$

$$L = \sum \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \left(\vec{x}_i \cdot \vec{x}_j\right) \tag{2.28}$$

As it is evident from the above equation that optimization depends upon the dot product of x_i and x_j.

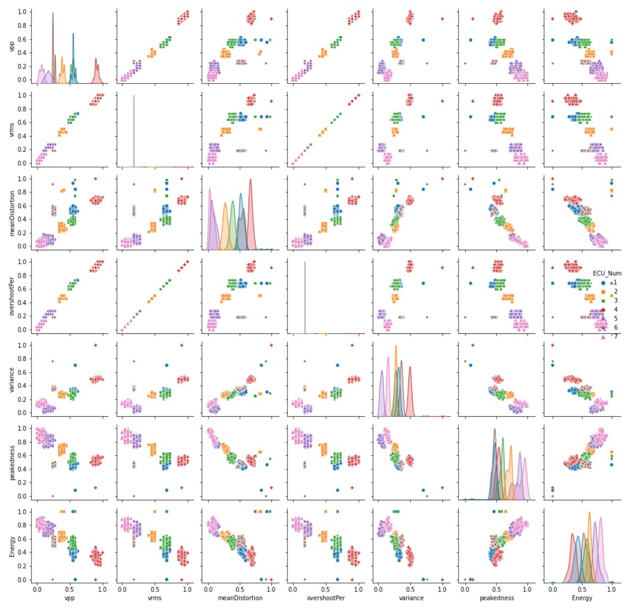


Figure 2.12: Separation of ECUs under various features.

2.6.3 Logistic Regression

LR is a predictive analysis method where the classification of the data is predicted based on the prior collected dataset [49]. It works on structured data only which are linearly separable. It separates the class based on decision boundaries and weights. In other words, it models the probability distribution of a target variable belonging to a specific class [50]. In this case, probability of inclusion is computed for the analysis.

2.7 Feature Separation

The illustration as shown in Fig. 2.12 refers to the separation of ECUs under the utilized features. It explains the pairwise relationship among the seven features. It visualizes relationship between each variable in a matrix format and by that, we can have an instant examination of ECU data. Also, the diagonal shows the separation of the individual features among the ECUs. For example, the V_{pp} feature shows maximum separation and the energy feature shows minimum separation.

2.8 Experimental Results And Performance Evaluation

2.8.1 k-Nearest Neighbour

The classification results of k-NN classifier as illustrated in Fig. 2.13, states that the two test classes from ECU 3 and one class from ECU 1 are wrongly classified as ECU 1. It is clearly shown that the clusters of ECU 1, ECU 2, ECU 4, ECU 5, and ECU 6 have distinct separations.

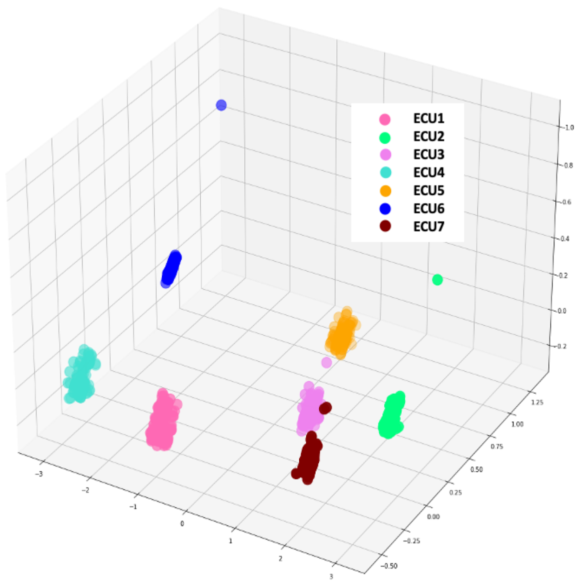


Figure 2.13: k-NN results.

Table 2.1 shows the accuracy results across the seven ECUs by using k-NN classification algorithm. It shows that an accuracy of 97.3% is obtained for predicting ECU 2. However, a total accuracy of 99.61% is achieved using the algorithm.

Table 2.1: Confusion matrix of k-Nearest Neighbour

		Predicted Class							
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	$ECU_{(7)}$	Total%
Target Class	$ECU_{(1)}$	114	1	0	0	0	0	0	99.12
	$ECU_{(2)}$	0	108	0	0	0	0	0	100
	$ECU_{(3)}$	0	2	103	0	0	0	0	98.06
	$ECU_{(4)}$	0	0	0	117	0	0	0	100
	$ECU_{(5)}$	0	0	0	0	113	0	0	100
	$ECU_{(6)}$	0	0	0	0	0	115	0	100
	$ECU_{(7)}$	0	0	0	0	0	0	116	100
	Total%	100	97.3	100	100	100	100	100	99.61

2.8.2 Support Vector Machine

The distinguished area of each ECU under SVM classification is illustrated by the surface diagram in Fig. 2.14. It is clearly observed that the five test classes from ECU 3 are mis-classified as ECU 1. Further, the accuracy is reduced because of the intersection in the boundaries of ECU 7 and ECU 5.

Further, table 2.2 shows the accuracy results across the seven ECUs under SVM classification algorithm. It is inferred that an accuracy of 95.65% is obtained for ECU 1 prediction and a total accuracy of 99.35% is achieved by the algorithm.

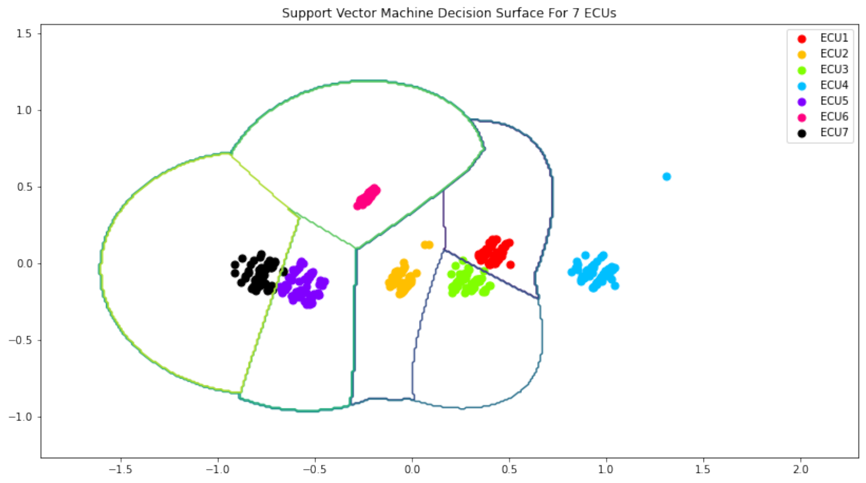


Figure 2.14: SVM results.

Table 2.2: Confusion matrix of Support Vector Machine

Predicted Class									
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	$ECU_{(7)}$	Total%
$TargetClass$	$ECU_{(1)}$	115	0	0	0	0	0	0	100
	$ECU_{(2)}$	0	108	0	0	0	0	0	100
	$ECU_{(3)}$	5	0	100	0	0	0	0	95.24
	$ECU_{(4)}$	0	0	0	117	0	0	0	100
	$ECU_{(5)}$	0	0	0	0	113	0	0	100
	$ECU_{(6)}$	0	0	0	0	0	115	0	100
	$ECU_{(7)}$	0	0	0	0	0	0	116	100
	Total%	95.65	100	100	100	100	100	100	99.35

2.8.3 Logistic Regression

Table 2.3 outlines the accuracy results across the seven ECUs under LR classification algorithm. It is clear that that the seven test classes from ECU 3 are mis-classified as ECU 1 and one test class from ECU 1 is mis-classified as ECU 3. Moreover, accuracy figures of 94.78% and 94.29% are achieved for ECU 1 and ECU 3 predictions respectively. Overall, a total accuracy is estimated to be 99.35%.

Table 2.3: Confusion matrix of Logistic Regression

Predicted Class									
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	$ECU_{(7)}$	Total%
$Target\ Class$	$ECU_{(1)}$	114	0	1	0	0	0	0	99.13
	$ECU_{(2)}$	0	108	0	0	0	0	0	100
	$ECU_{(3)}$	7	0	98	0	0	0	0	93.33
	$ECU_{(4)}$	0	0	0	117	0	0	0	100
	$ECU_{(5)}$	0	0	0	0	113	0	0	100
	$ECU_{(6)}$	0	0	0	0	0	115	0	100
	$ECU_{(7)}$	0	0	0	0	0	0	116	100
	Total%	94.78	100	94.29	100	100	100	100	98.68

CHAPTER 3

Eigenwave Based Principal Component Analysis for ECU Profiling

3.1 Background and Motivation

Automation and the application of artificial intelligence are modernizing the automotive industry and changing the way people perceive it. The ECU (Electronic Control Unit) is pivotal in facilitating vehicle automation such as elevating the user experience by monitoring and making dynamic decisions based on the sensor data for ADAS. Cars that offer semi-autonomous driving features like Tesla Autopilot hinge heavily on the signal processing of the ECU. Automotive researchers are experimenting with vehicle-to-vehicle interactions, and ECUs will play an additional role in communicating with the nearest ECUs during transit. This versatile component is subjected to a multitude of threats and attacks, such as remote hacking into a Jeep Cherokee's ECU in 2015 [51] and the 2017 Tesla Model S ECU breach using a physical custom-designed system [52]. These threats pose a perilous situation for the vehicle and its occupants. In a wave of the increasingly connected and automated automobile era, it is paramount to ensure that ECUs are resilient to hacking and spoofing attacks.

The ECUs communicate in the CAN bus in such a way as to ensure seamless operation and good performance. One such commonly used approach to ECU configuration in a significant part of auto manufacturers is the daisy chain configuration. Some of the notable brands that deploy the above configuration are Mercedes-Benz, Audi, Volkswagen, and Porsche. A substantial proportion of the other auto manufacturers use similar configurations.

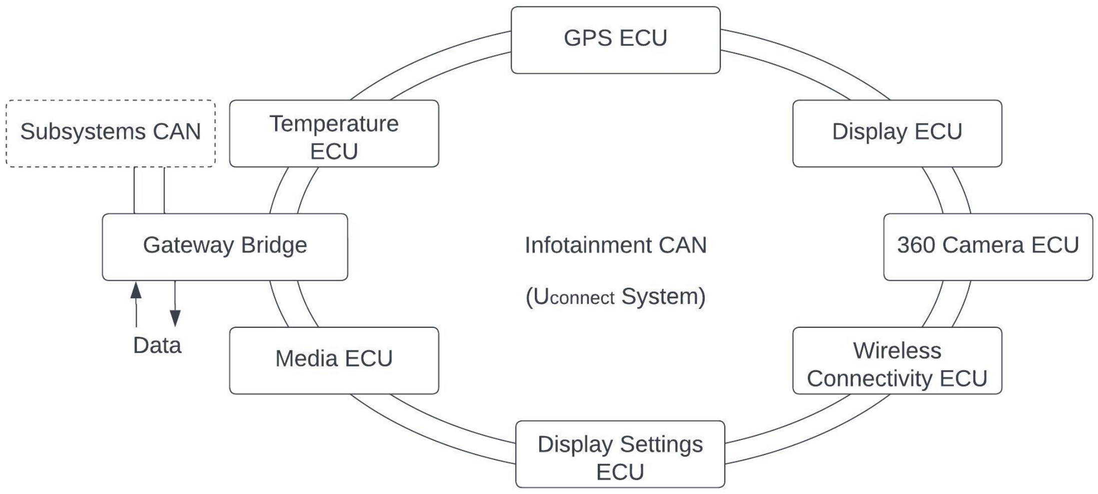


Figure 3.1: Infotainment CAN bus illustrating UConnect system

Specific ECUs in the high-speed CAN bus has a unique ID that protects them against spoofing to some extent. And the CAN bus arbitration of a signal happens by taking into account this unique ID and a priority level. But a sizeable segment of the low-speed CAN bus ECUs involves lighting, HVAC, and infotainment systems as shown in Fig. 3.1, climate control, and a few other less critical areas. These peripheral CAN bus needs to be profiled in order to prevent critical CAN bus hacking through these low-speed CAN bus. The incident where the engine and brake ECU of the 2015 Jeep Cherokee were accessed by remotely hacking [51] into the Uconnect system is a compelling scenario that highlights the need to secure the less critical CAN bus.

The daisy chain a is commonly used configuration for in-vehicle infotainment systems. In this configuration, ECUs are connected in series with the adjacent ones and the CAN bus is systematically designed to connect through the every ECU node. To reduce signal reflections, electromagnetic interference, and signal distortion, we are maintaining a minimum distance of 1 meter between two ECUs. Each ECU in the CAN bus are approximately 1 meter apart as shown in Fig. 3.2.

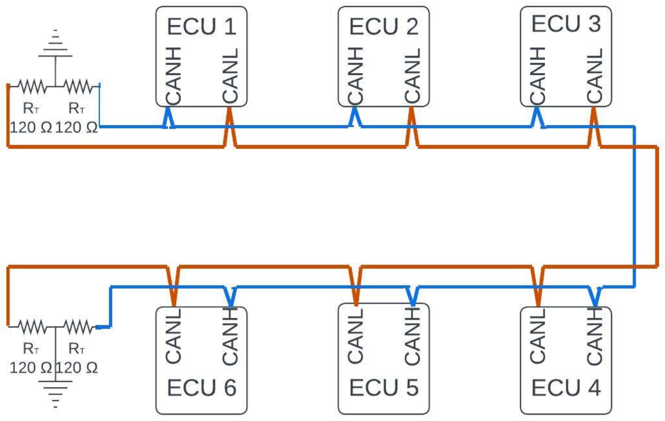


Figure 3.2: ECUs connected in Daisy chain CAN configuration.

The Fig. 3.2 expounds the Daisy chain CAN configuration, in which each ECU in the CAN bus has a CAN input and CAN output, and every ECU is connected as a chain in a linear fashion. The CAN data link connects multiple ECUs to enable engine operations. It consists of two twisted pair wires, CAN high and CAN low, terminated with a resistor on each end. The bus has two logic states: logical 0, which corresponds to the dominant bus level, and logical 1, which is the recessive level. The voltage level on the bus is recessive (2.5V) when there is no transmission of ECU information and goes into a dominant state (3.5V)[3] when a message is transmitted.

Contemporary research suggests mitigating hacking vulnerabilities through opportunities like Authentic Group Keys [53], ML-based security approaches [54, 55]. Literature has proposed a number of preventive methods like message authentication-based approaches which implement security at the data link layer. [5, 6, 7, 8, 9, 10, 11, 12, 13]. Approaches based on intrusion detection can be used to implement security at the physical layer. [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36].

Though the evolving technologies pave way for innovative security measures, hackers find new methodologies to exploit the ECUs and break into the vehicle system. With no to less standardization of the ECU fingerprinting most of vehicles are still prone to attacks. Many proposed solutions require a complex feature set and have high computation time. The main objective of this paper is to propose an efficient solution by considering the visualization of signal loss and distortion due to the position of the ECU in the CAN bus and using the weight of principal components to classify the ECUs.

Section II of this paper presents the possible arrangements of ECUs and the data collection process for various configuration studies. This section also emphasizes the extraction of usable data from the raw data. In Section III, we discuss the restructuring of data into images to fit into the principal component analysis (PCA) framework. Section IV elaborates on the PCA methodology and visualizes the projected standard variations as eigenwaves, from which we derived eigenvectors. The experimental results are presented in Section V, and finally, in Section VI, we conclude the paper.

3.2 Data Preparation

Information gathering, scrubbing, and preparation are crucial steps in making informed decisions. Our methodology involves studying the behavior of all six ECUs and analyzing the differences. In a daisy chain configuration, the position of each ECU in the chain plays a crucial role in determining the overall efficiency of the system. With six ECUs in the Hardware-in-the-loop (HIL) simulation, we have

$$nP_r = \frac{n!}{(n-r)!} = \frac{6!}{0!} = 720 \tag{3.1}$$

different ways the daisy chain CAN can be configured as shown in the Fig. 3.3

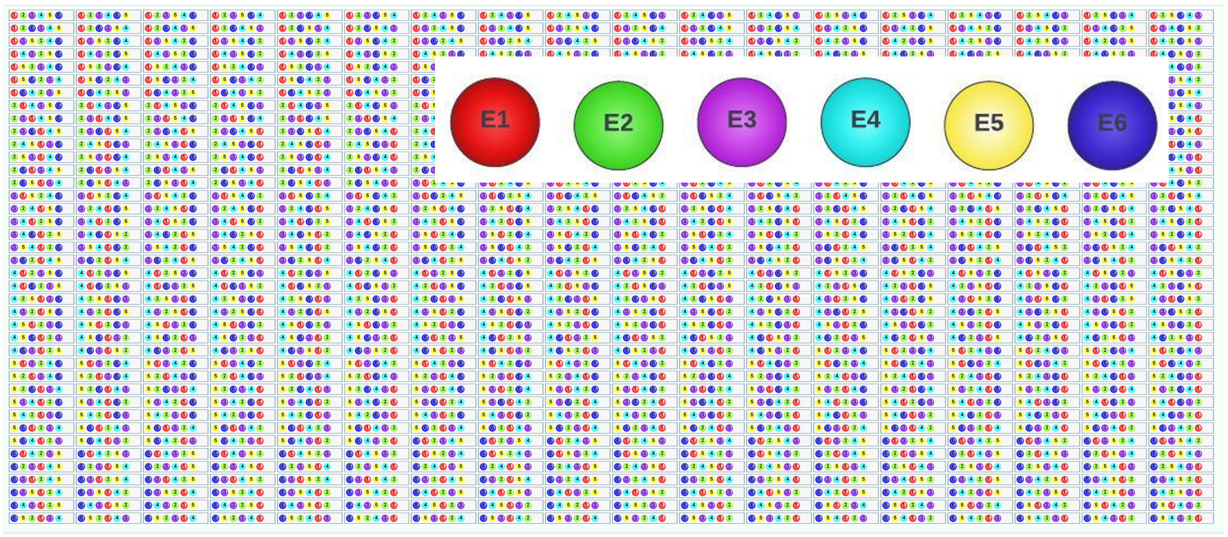


Figure 3.3: Possible combination of six ECUs in a Daisy chain configuration

To demonstrate the applicability of this method, data is collected from each ECU using six different wire lengths, including 1 meter, 2 meters, 3 meters, 4 meters, 5 meters, and 10 meters. This is based on the assumption that the ECUs can be placed in different positions in the CAN configuration. The dominant and recessive voltage data is collected for each ECU using six different wire lengths. This method has the benefit of being able to place the ECUs in any of the six positions and test for the efficiency of the system as shown in Fig 3.4.

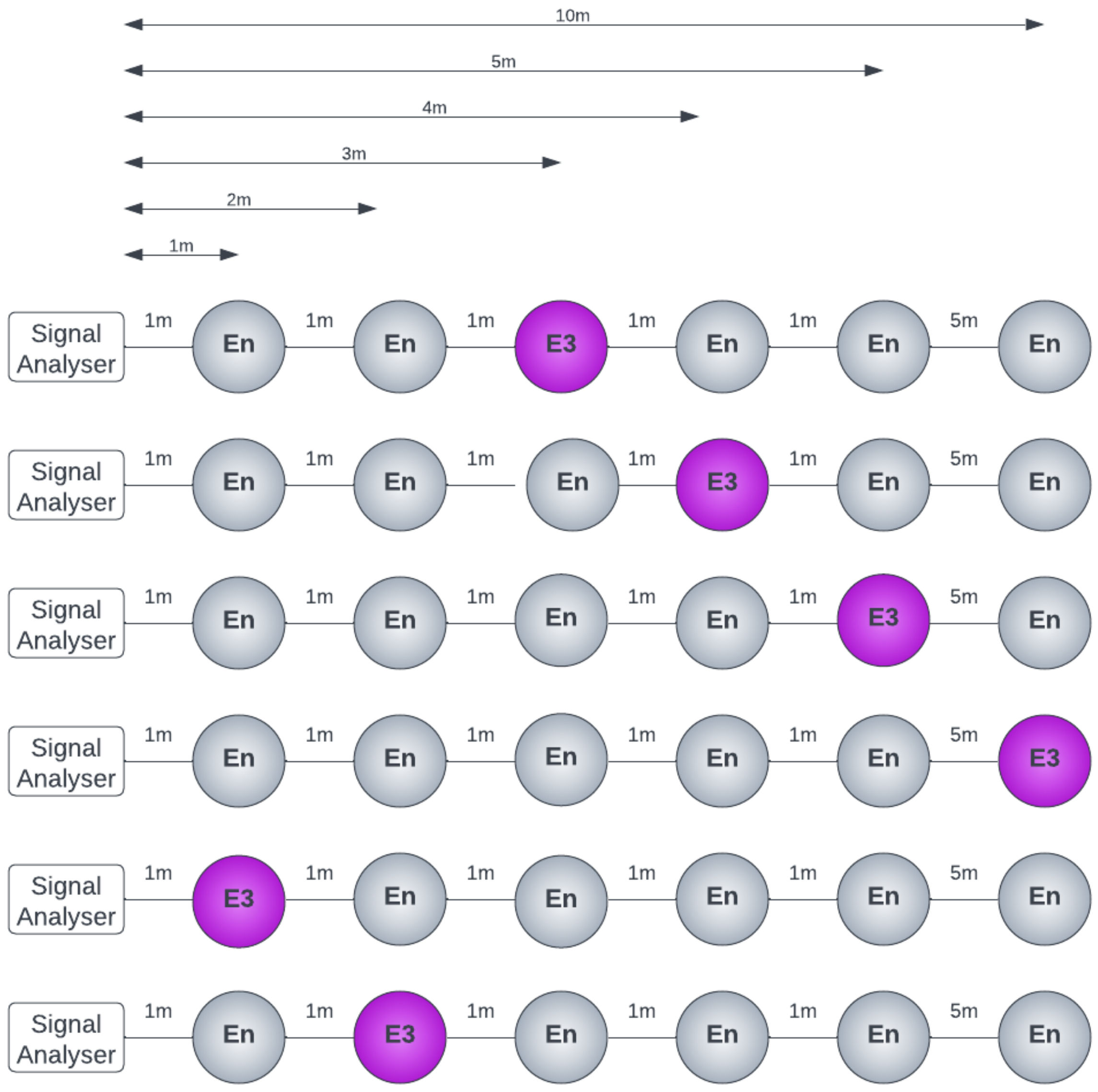


Figure 3.4: Combinations of ECU 3 placement in the CAN with the signal analyzer

The collected data is cleaned to remove short pulses and Inter-Frame Spacing (IFS) data. After scrubbing, the voltages are inspected to identify individual square waves corresponding to each wire length for each ECU. The square wave data for each wire length and ECU is then utilized for data manipulation and feature extraction.

3.3 Data Restructuring And Filtering

Computational machines well understand images as every pixel color can be coded and converted to numerical values to extract features.[56] The square waves from individual ECUs for all the lengths are converted into graphical images, which are then subjected to feature extraction. The area under each square wave is filled with a uniform color, such as red, to make the region of interest distinct as shown in the Fig.3.5.

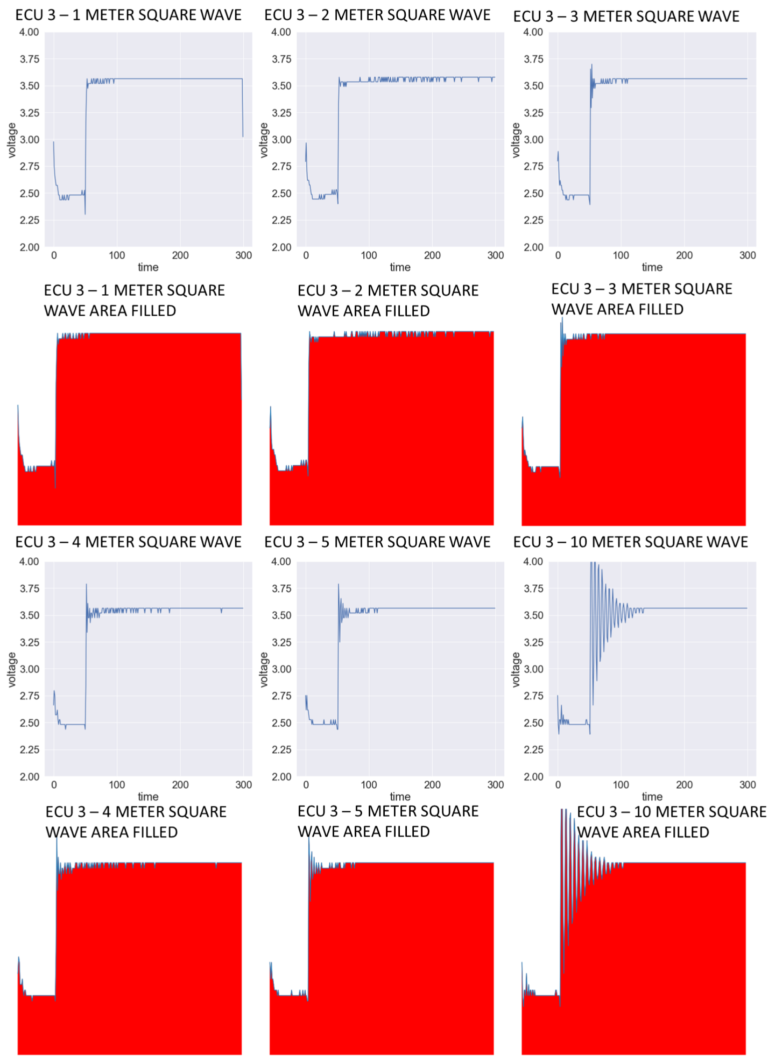


Figure 3.5: ECU 3 Square waves 1m-10m - Area under the curve

Before extracting features from the M x M images, it is necessary to perform geometric rectification by adjusting and standardizing the axis and adjusting the dimensions across all the images[57]. This ensures a consistent input for all feature extraction processes. 240 images are chosen for feature extraction, with 40 images taken for each length under each ECU.

3.4 Methodology

Any tone or intensity in an image can be represented as a numerical value assigned to that point of coordinate in the image [58]. A geometrically rectified, size-compressed, and color-coded M x M image can be represented by a M^2 x N matrix of numerical values that a computational machine can understand. This technique enables the classification of similar images into a category, even if they are not identical.[56]. The primary action to make this image matrix computationally efficient is to reduce its dimensionality, which we are achieving using PCA.

$$E_{rd} = f_{PCA}[E] \tag{3.2}$$

Where E is the initial set of attributes of the image and E_{rd} is the reduced dimensions of the initial set of attributes using PCA function f_{PCA}. The principal components or reduced dimensions are derived by centering the data around the origin, which involves subtracting the mean from each attribute.

$$E_i = E_i - \mu \tag{3.3}$$

Where E_1 and E_2 are the high-dimensional attributes and μ is mean. Once the data is centered around the origin the covariance matrix of the data is computed to understand the type of relationship between the attributes.

$$cov(\Sigma) = \frac{1}{n} \sum_{i=0}^{n} Ei.(E_i)^T \tag{3.4}$$

Where n is the number of observations in the sample. Once the covariance matrix is in place each vector from the original dataset is multiplied by the covariance matrix. The vectors that point in the same direction as the variance are termed eigenvectors v. Eigenvalues are calculated by solving the below equation, where Σ is the covariance matrix, λ is the eigenvalue, and I is the identity matrix.

$$\det(\Sigma - \lambda I) = 0 \tag{3.5}$$

The eigenvectors v with maximum eigenvalues λ are called principal components. As the final step to get the low-dimension data E_{rd}, the eigenvectors v are multiplied with centered data.

$$E_{rdi} = v.(E_i - \mu) \tag{3.6}$$

Where E_{rd} is the low-dimension data obtained by principal component analysis of the images [59].

PCA is extensively utilized in face identification to extract a small set of standardized features from vast datasets, which are subsequently employed for classification[60]. Many memory-efficient algorithms are based on this concept, and eigenfaces have inspired many image classifiers. Eigenface decomposes a large set of face samples into a collection of dominant variance sets[61].

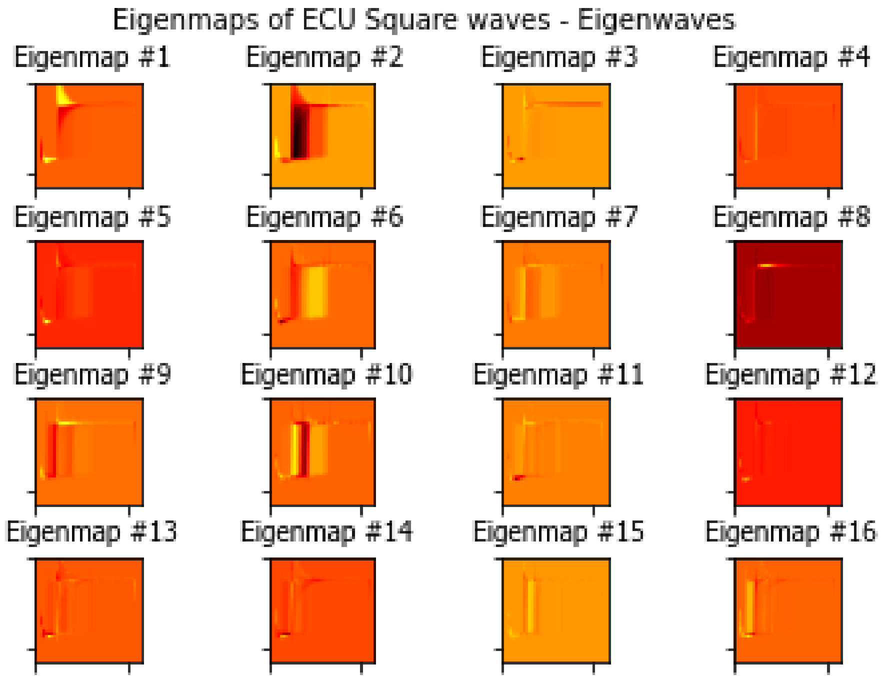


Figure 3.6: Eigenwaves showing dominant variance across the sample set

The eigenwaves presented in this paper capture the standard variations observed across the samples, and the dominant features exhibiting high variance are identified as a set of eigenwaves. These eigenwaves represent the principal components of the square wave sample set and help visualize the prototypical wave attributes. Using these eigenwaves, it is possible to reconstruct actual square waves with high accuracy.

Fig.3.6 showcases the distinguishing features of the six different ECUs through a collection of images highlighting dominant variance. The original square wave can be reconstructed by partially summing these images, along with the mean wave.

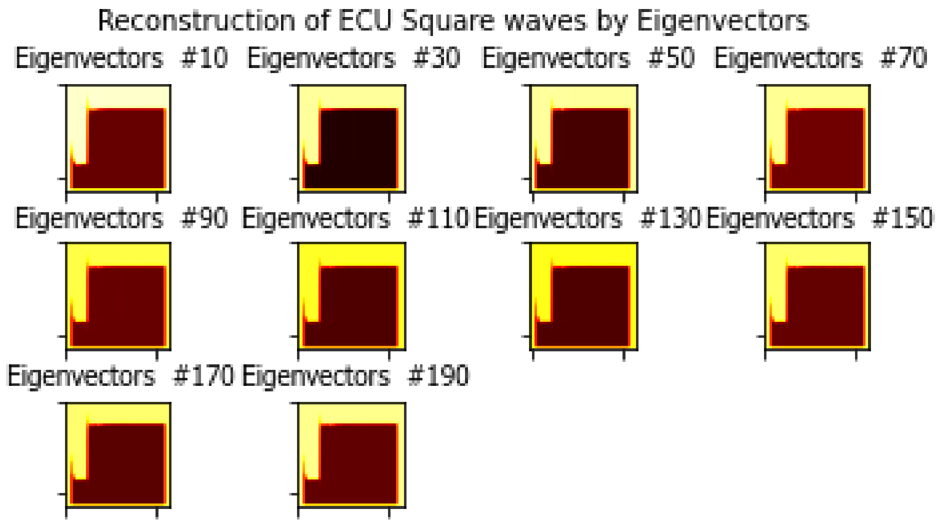


Figure 3.7: Eigenvectors showing the progression of square wave reconstruction

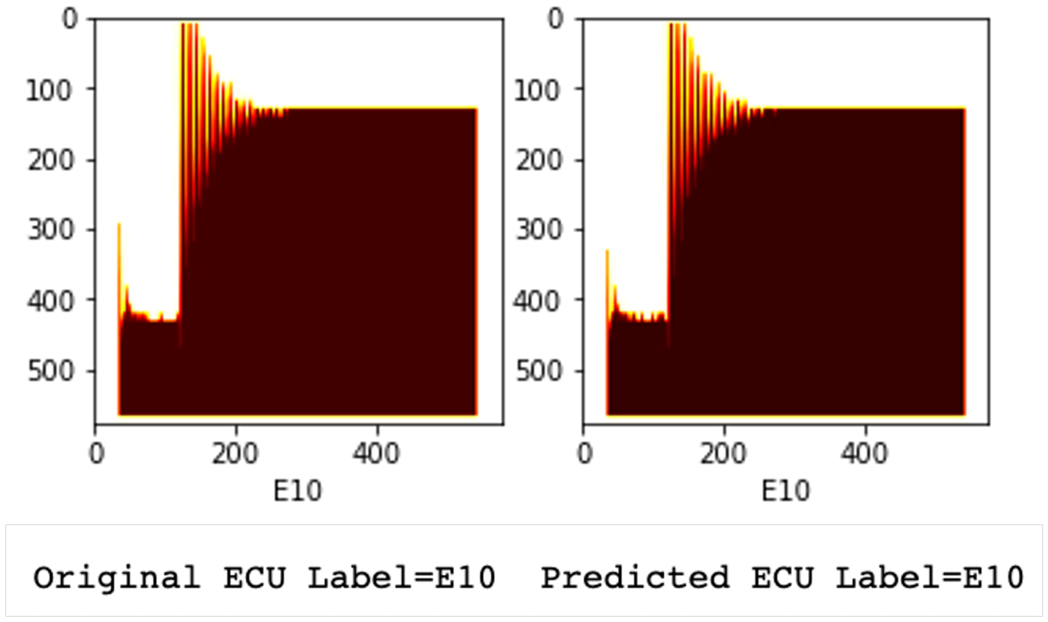
The progression of square wave reconstruction is represented by the eigenvectors shown in Fig.3.7.

The next step involves classification, and various ML-based models can be utilized to fit the PCA attributes and classify the ECU corresponding to the square wave. In this paper, we adopt a Euclidean distance-based approach for classification. To classify the ECU square wave image into its corresponding class, the algorithm begins by subtracting the mean vector obtained from PCA from the vectorized input image. The resulting mean-subtracted vector is then projected onto each eigenface, and the weights are calculated. These weights are then used for classification. Every image in the sample set has weight data to compare, and the obtained weight data is compared with the training data to classify the ECU square wave. This method is highly effective in our approach.

3.5 Results

To understand the efficiency of the daisy chain configuration with different positions of ECUs, we simulated the ECUs in various arrangements. This paper focuses on testing the system's efficiency in six different configurations. To verify whether the position of an ECU has a significant impact on profiling, we combine the data of all the ECUs tested in the hardware-in-loop simulation, assuming that any ECU can be placed in six different positions in any order, with similar wire lengths. This analysis shows that the square waves from the ECUs closer to the signal analyzer in the CAN bus chain exhibit distinct differences compared to those farther away.

The Fig.3.8 shows an accurate identification and also we observe 97.14% of classification accuracy by this approach. The efficiency of this system can be further validated by considering real-time scenarios and exploring various ECU placements in the CAN BUS to validate the working accuracy. The confusion matrix of this approach is listed in 3.1.



```
Original ECU Label=E10  Predicted ECU Label=E10
```

Figure 3.8: Classification result of ECUs

Table 3.1: Confusion matrix of Daisy Chain Simulation

Predicted Class								
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
Target Class	$ECU_{(1)}$	136	0	0	3	1	0	97.14
	$ECU_{(2)}$	0	139	1	0	0	0	99.28
	$ECU_{(3)}$	13	0	127	0	0	0	90.71
	$ECU_{(4)}$	0	0	1	136	3	0	97.14
	$ECU_{(5)}$	1	0	0	1	138	0	98.57
	$ECU_{(6)}$	0	0	0	0	0	140	100
	Total%	90.66	100	98.44	97.14	97.18	100	97.14

3.5.1 Configuration 1

In Fig.3.9, The arrangement of the ECUs involves placing $ECU1$, $ECU2$, $ECU3$, $ECU4$, $ECU5$, and $ECU6$ at a distance of $1meter$, $2meters$, $3meters$, $4meters$, $5meters$, and $10meters$ from the signal analyzer.

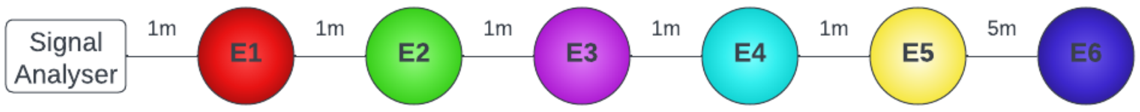
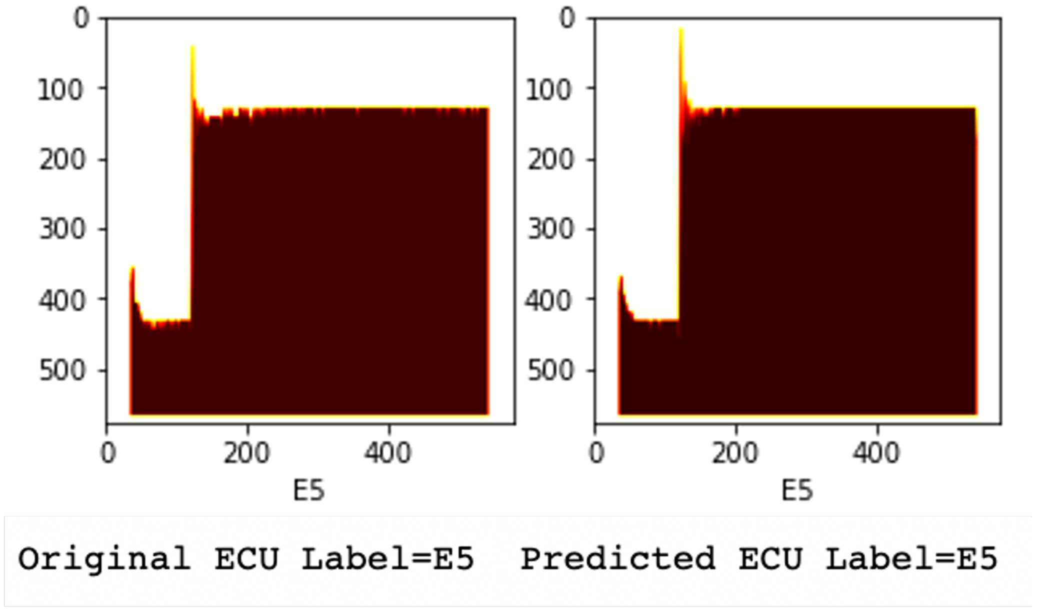


Figure 3.9: Arrangement of ECUs in the CAN - Configuration 1

The results of this arrangement show a classification accuracy of 98.35%. The confusion matrix of this approach is listed in 3.2.



```
Original ECU Label=E5  Predicted ECU Label=E5
```

Figure 3.10: Classification result of configuration 1

Table 3.2: Confusion matrix of Daisy Chain Configuration 1

Predicted Class								
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
$Target\ Class$	$ECU_{(1)}$	20	0	0	0	0	0	100
	$ECU_{(2)}$	0	20	0	0	0	0	100
	$ECU_{(3)}$	0	1	20	0	0	0	95.23
	$ECU_{(4)}$	1	0	0	19	0	0	95
	$ECU_{(5)}$	0	0	0	0	20	0	100
	$ECU_{(6)}$	0	0	0	0	0	20	100
	Total%	95.23	95.23	100	100	100	100	98.35

3.5.2 Configuration 2

In Fig.3.11, The arrangement of the ECUs involves placing $ECU3$, $ECU5$, $ECU1$, $ECU6$, $ECU4$, and $ECU2$ at a distance of $1meter$, $2meters$, $3meters$, $4meters$, $5meters$, and $10meters$ from the signal analyzer.

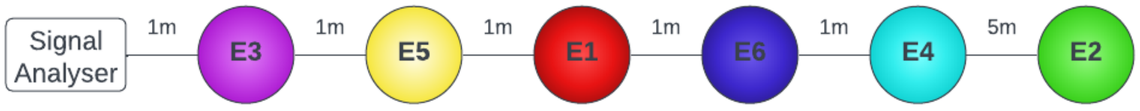
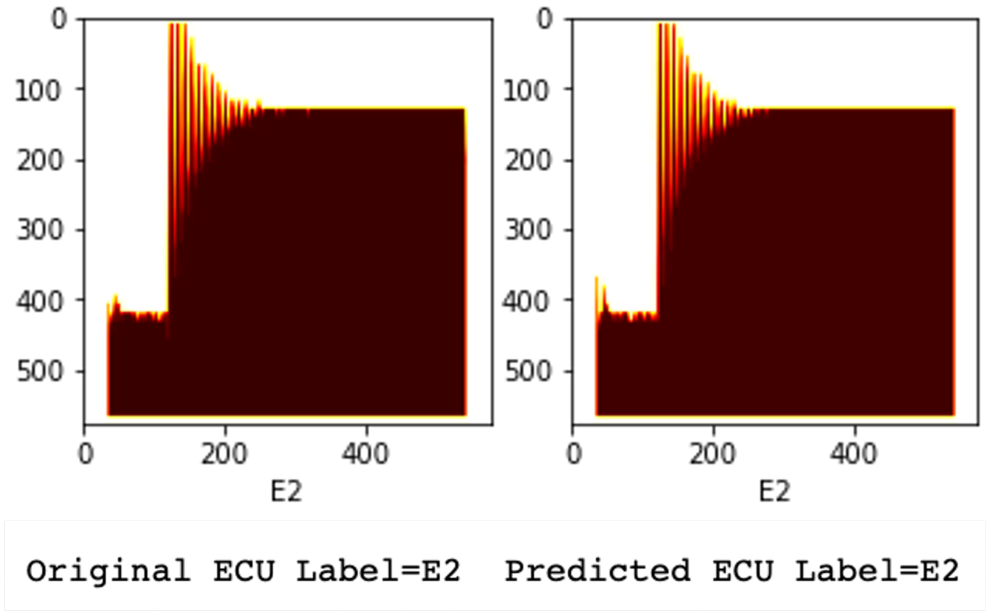


Figure 3.11: Arrangement of ECUs in the CAN - Configuration 2

The results of this arrangement show a classification accuracy of 100%. The confusion matrix of this approach is listed in 3.3.



```
Original ECU Label=E2  Predicted ECU Label=E2
```

Figure 3.12: Classification result of configuration 2

Table 3.3: Confusion matrix of Daisy Chain Configuration 2

Predicted Class								
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
$Target\ Class$	$ECU_{(1)}$	20	0	0	0	0	0	100
	$ECU_{(2)}$	0	20	0	0	0	0	100
	$ECU_{(3)}$	0	0	20	0	0	0	100
	$ECU_{(4)}$	0	0	0	20	0	0	100
	$ECU_{(5)}$	0	0	0	0	20	0	100
	$ECU_{(6)}$	0	0	0	0	0	20	100
	Total%	100	100	100	100	100	100	100

3.5.3 Configuration 3

In Fig.3.13, The arrangement of the ECUs involves placing $ECU5$, $ECU4$, $ECU2$, $ECU1$, $ECU6$, and $ECU3$ at a distance of $1meter$, $2meters$, $3meters$, $4meters$, $5meters$, and $10meters$ from the signal analyzer.

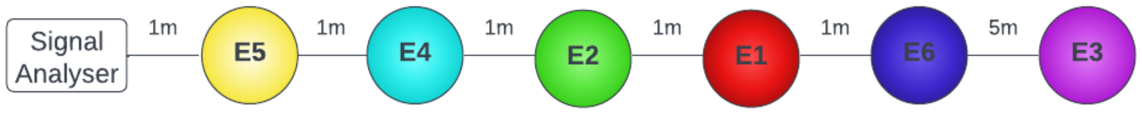
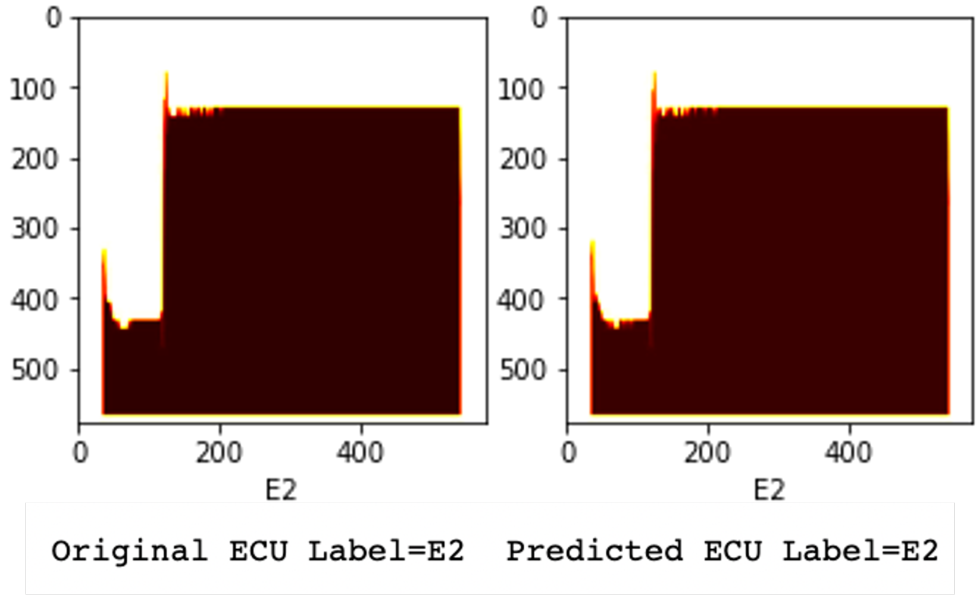


Figure 3.13: Arrangement of ECUs in the CAN - Configuration 3

The results of this arrangement show a classification accuracy of 95.83%. The confusion matrix of this approach is listed in 3.4.



```
Original ECU Label=E2   Predicted ECU Label=E2
```

Figure 3.14: Classification result of configuration 3

Table 3.4: Confusion matrix of Daisy Chain Configuration 3

Predicted Class								
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
$Target\ Class$	$ECU_{(1)}$	16	3	0	1	0	0	80
	$ECU_{(2)}$	0	20	0	0	0	0	100
	$ECU_{(3)}$	0	0	20	0	0	0	100
	$ECU_{(4)}$	0	0	0	20	0	0	100
	$ECU_{(5)}$	0	1	0	0	19	0	95
	$ECU_{(6)}$	0	0	0	0	0	20	100
	Total%	100	83.33	100	95.23	100	100	95.83

3.5.4 Configuration 4

In Fig.3.15, The arrangement of the ECUs involves placing $ECU2$, $ECU1$, $ECU6$, $ECU5$, $ECU3$, and $ECU4$ at a distance of $1meter$, $2meters$, $3meters$, $4meters$, $5meters$, and $10meters$ from the signal analyzer.

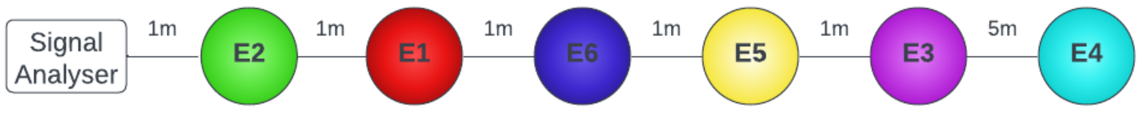
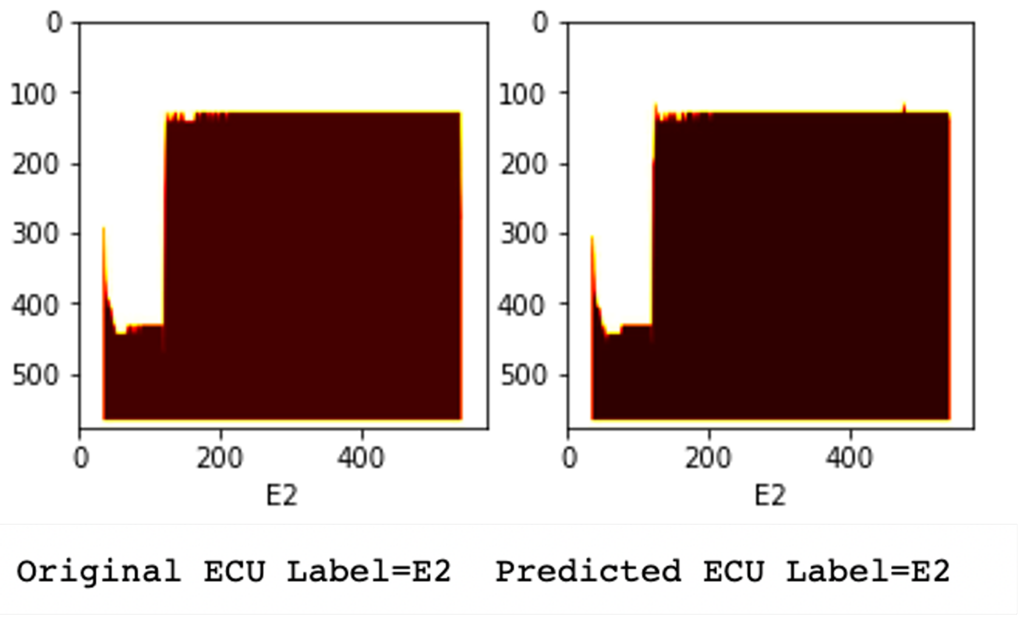


Figure 3.15: Arrangement of ECUs in the CAN - Configuration 4

The results of this arrangement show a classification accuracy of 100%. The confusion matrix of this approach is listed in 3.5.



```
Original ECU Label=E2  Predicted ECU Label=E2
```

Figure 3.16: Classification result of configuration 4

Table 3.5: Confusion matrix of Daisy Chain Configuration 4

Predicted Class								
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
$Target\ Class$	$ECU_{(1)}$	20	0	0	0	0	0	100
	$ECU_{(2)}$	0	20	0	0	0	0	100
	$ECU_{(3)}$	0	0	20	0	0	0	100
	$ECU_{(4)}$	0	0	0	20	0	0	100
	$ECU_{(5)}$	0	0	0	0	20	0	100
	$ECU_{(6)}$	0	0	0	0	0	20	100
	Total%	100	100	100	100	100	100	100

3.5.5 Configuration 5

In Fig.3.17, The arrangement of the ECUs involves placing $ECU6$, $ECU3$, $ECU4$, $ECU2$, $ECU1$, and $ECU5$ at a distance of $1meter, 2meters, 3meters, 4meters, 5meters$, and $10meters$ from the signal analyzer.

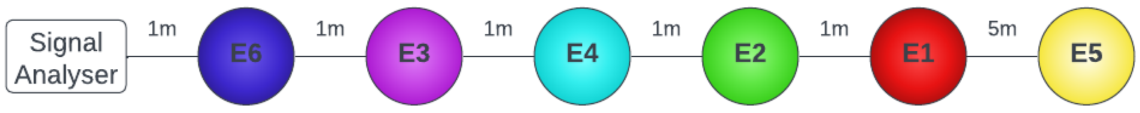
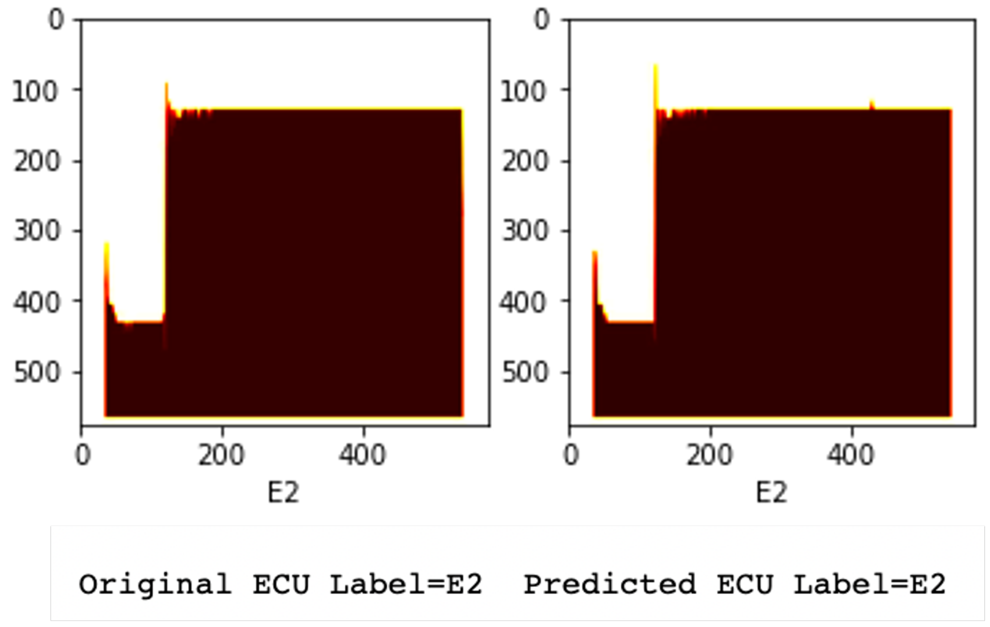


Figure 3.17: Arrangement of ECUs in the CAN - Configuration 5

The results of this arrangement show a classification accuracy of 99.17%. The confusion matrix of this approach is listed in 3.6.



```
Original ECU Label=E2  Predicted ECU Label=E2
```

Figure 3.18: Classification result of configuration 5

Table 3.6: Confusion matrix of Daisy Chain Configuration 5

Predicted Class								
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
$Target\ Class$	$ECU_{(1)}$	19	1	0	0	0	0	95
	$ECU_{(2)}$	0	20	0	0	0	0	100
	$ECU_{(3)}$	0	0	20	0	0	0	100
	$ECU_{(4)}$	0	0	0	20	0	0	100
	$ECU_{(5)}$	0	0	0	0	20	0	100
	$ECU_{(6)}$	0	0	0	0	0	20	100
	Total%	100	95.23	100	100	100	100	99.17

3.5.6 Configuration 6

In Fig.3.17, The arrangement of the ECUs involves placing $ECU4$, $ECU6$, $ECU5$, $ECU3$, $ECU2$, and $ECU1$ at a distance of $1meter$, $2meters$, $3meters$, $4meters$, $5meters$, and $10meters$ from the signal analyzer.

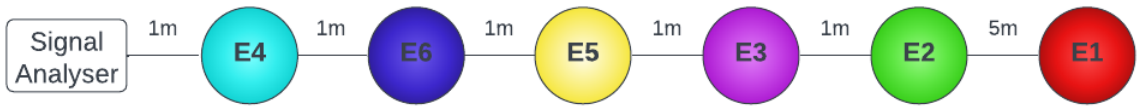


Figure 3.19: Arrangement of ECUs in the CAN - Configuration 6

The results of this arrangement show a classification accuracy of 95%.The confusion matrix of this approach is listed in 3.7.

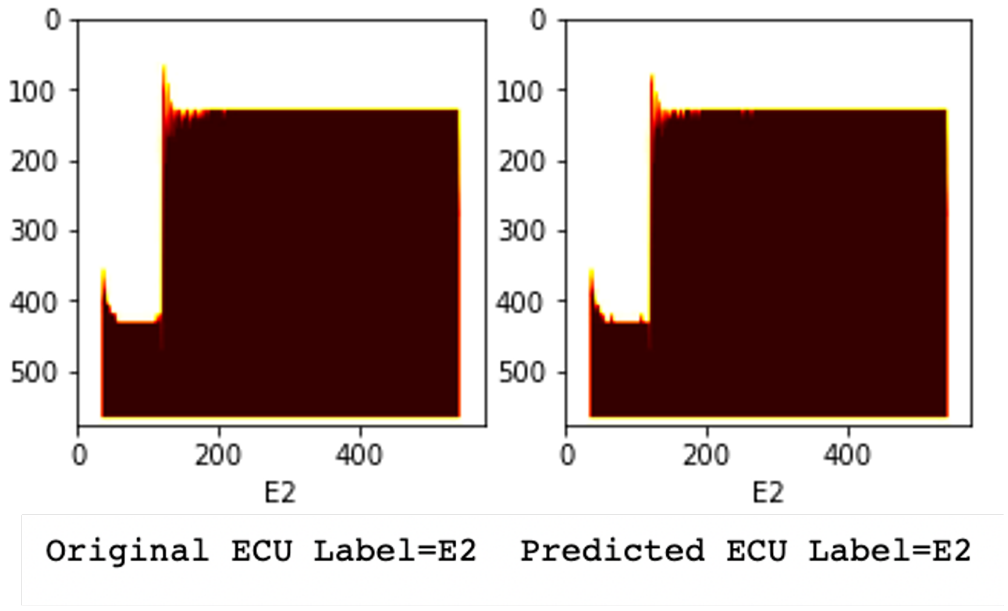


Figure 3.20: Classification result of configuration 6

Table 3.7: Confusion matrix of Daisy Chain Configuration 6

		Predicted Class						
-	-	$ECU_{(1)}$	$ECU_{(2)}$	$ECU_{(3)}$	$ECU_{(4)}$	$ECU_{(5)}$	$ECU_{(6)}$	Total%
$Target\ Class$	$ECU_{(1)}$	19	0	0	0	0	1	95.23
	$ECU_{(2)}$	0	20	0	0	0	0	100
	$ECU_{(3)}$	0	0	19	0	1	0	95.23
	$ECU_{(4)}$	0	0	0	20	0	0	95.23
	$ECU_{(5)}$	0	0	0	0	20	0	100
	$ECU_{(6)}$	0	0	0	4	0	16	80
	Total%	100	100	100	83.33	95.23	94.11	95

Based on the experimental results, it can be concluded that configurations 2 and 4 are the most efficient for ECU profiling using the signal analyzer. Additionally, other configurations were also tested and demonstrated good performance.

CHAPTER 4

Conclusion

In this study, we have proposed two methods for enhancing the security of in-vehicle networks against potential cyber-attacks. Method 1 is based on ECU fingerprinting, which successfully identifies the source ECU responsible for sending malicious packets in the communication channel by utilizing various ML classification algorithms (e.g., k-NN, SVM and LR). To evaluate the proposed approach, we have used classification accuracy as a performance metric, and obtained promising figures of 99.61%, 99.35% and 98.68% for the respective algorithms. These results demonstrate that the proposed method is potentially feasible and effective in identifying abnormal behaviors in an in-vehicle CAN network, and can significantly improve the security of in-vehicle networks. For future work, we suggest exploring the performance of other channels such as FlexRay, MOST, and Ethernet, and testing the proposed method under different environmental conditions such as electromagnetic interference and high temperatures.

Method 2 is based on PCA-based image classification, which effectively identifies and profiles ECUs based on their position in the CAN bus. Our proposed system provides a robust and scalable solution for securing automotive systems against potential cyber-attacks. The direct data acquisition from the CAN bus minimizes the possibility of corrupted or non-square wave data, and the scalability of the system enables economic and commercial viability. Our results demonstrate the effectiveness of the proposed system in identifying anomalies in the signal distortion, which can be used to detect errors in the ECU termination resistor. Future research can be done on investigating the applicability of the proposed method in other automotive networks and exploring its performance under different environmental conditions.

Overall, the proposed methods have shown significant promise in improving the security of in-vehicle networks against cyber-attacks. Method 1's ECU fingerprinting approach can potentially identify abnormal behaviors in the in-vehicle CAN network, while Method 2's PCA-based image classification can provide a robust and scalable solution for securing automotive systems. Further research can be done to explore the performance of these methods under various conditions and in different automotive networks.

REFERENCES

[1] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 International Conference on Information Networking (ICOIN)*, 2016, pp. 63–68.

[2] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *2017 AEIT International Annual Conference*, 2017, pp. 1–6.

[3] A. Hafeez, "A robust, reliable and deployable framework for in-vehicle security," Ph.D. dissertation, The University of Michigan Dearborn.

[4] A. A. Elkhail, R. U. D. Refat, R. Habre, A. Hafeez, A. Bacha, and H. Malik, "Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses," *IEEE Access*, pp. 1–1, 2021.

[5] A. Hafeez, K. Topolovec, C. Zolo, and W. Sarwar, "State of the art survey on comparison of can, flexray, lin protocol and simulation of lin protocol," in *State of the Art Survey on Comparison of CAN, FlexRay, LIN Protocol and Simulation of LIN Protocol*, 04 2020.

[6] A. Hafeez, K. Rehman, and H. Malik, "State of the art survey on comparison of physical fingerprinting-based intrusion detection techniques for in-vehicle security," in *State of the Art Survey on Comparison of Physical Fingerprinting-Based Intrusion Detection Techniques for In-Vehicle Security*, 04 2020.

[7] A. Radu and F. Garcia, "Leia: A lightweight authentication protocol for can," in *LeiA: A Lightweight Authentication Protocol for CAN*, vol. 9879, 09 2016, pp. 283–300.

[8] T. P. Doan and S. Ganesan, "Can crypto fpga chip to secure data transmitted through can fd bus using aes-128 and sha-1 algorithms with a symmetric key," SAE Technical Paper, Tech. Rep., 2017.

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

[10] T. Sugashima, D. K. Oka, and C. Vuillaume, "Approaches for secure and efficient in-vehicle key management," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 9, no. 2016-01-0070, pp. 100–106, 2016.

[11] A. Hafeez, H. Malik, O. Avatefipour, P. R. Rongali, and S. Zehra, "Comparative study of can-bus and flexray protocols for in-vehicle communication," SAE Technical Paper, Tech. Rep., 2017.

[12] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.

[13] R. U. D. Refat, A. A. Elkhail, A. Hafeez, and H. Malik, "Detecting can bus intrusion by applying machine learning method to graph based features," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2021, pp. 730–748.

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

[15] B. Groza and P.-S. Murvay, "Efficient intrusion detection with bloom filtering in controller area networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1037–1051, 2019.

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

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

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

[19] H. Song, H. Kim, and H. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *Int. Conf. on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[20] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *IEEE 2nd Int. Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. IEEE.

[21] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks," *IEEE Access*, vol. 6, pp. 45 233–45 245, 2018.

[22] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *Int. Annual Conf. AEIT*. IEEE, 2017, pp. 1–6.

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

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

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

[26] N. Jain and S. Sharma, "The role of decision tree technique for automating intrusion detection system," *Int. Jour. of Computational Engineering Research*, vol. 2, no. 4, 2012.

[27] R. Rieke, M. Seidemann, E. Talla, D. Zelle, and B. Seeger, "Behavior analysis for safety and security in automotive systems," in *25th Euromicro Int. Conf. on Parallel, Distributed and Network-based Processing (PDP)*. IEEE, 2017, pp. 381–385.

[28] S. Narayanan, S. Mittal, and A. Joshi, "Using data analytics to detect anomalous states in vehicles," *arXiv preprint arXiv:1512.08048*, 2015.

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

[30] A. Hafeez, M. Tayyab, C. Zolo, and S. Awad, "Fingerprinting of engine control units by using frequency response for secure in-vehicle communication," in *2018 14th International Computer Engineering Conference (ICENCO)*. IEEE, 2018, pp. 79–83.

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

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

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

[34] O. Avatefipour, A. Hafeez, M. Tayyab, and H. Malik, "Linking received packet to the transmitter through physical-fingerprinting of controller area network," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*. IEEE, 2017, pp. 1–6.

[35] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000*. IEEE, 2000, pp. 56–73.

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

[37] J. R. Fernandes, F. A. de Sá, J. L. Santos, and E. Joanni, "Optical fiber interferometer for measuring the d33 coefficient of piezoelectric thin films with compensation of substrate bending," *Review of Scientific Instruments*, vol. 73, no. 5, pp. 2073–2078, 2002. [Online]. Available: https://doi.org/10.1063/1.1463713

[38] S.-H. Cho, C.-H. Park, J. Han, and G. Jang, "A waveform distortion evaluation method based on a simple half-cycle rms calculation," *IEEE Transactions on Power Delivery*, vol. 27, no. 3, pp. 1461–1467, 2012.

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

[40] F. E. Satterthwaite, "Synthesis of variance," *Psychometrika*, vol. 6, no. 5, pp. 309–316, 1941.

[41] K. V. Cartwright, "Determining the effective or rms voltage of various waveforms without calculus," *The Technology Interface*, vol. 8, no. 1, pp. 1–20, 2007.

[42] P. S. Horn, "A measure for peakedness," *The American Statistician*, vol. 37, no. 1, pp. 55–56, 1983. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1983.10483090

[43] K. A. Sharp and B. Honig, "Calculating total electrostatic energies with the nonlinear poisson-boltzmann equation," *Journal of Physical Chemistry*, vol. 94, no. 19, pp. 7684–7692, 1990.

[44] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[45] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227–248, 1980.

[46] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Advances in neural information processing systems*, 2012, pp. 1061–1069.

[47] J. M. Merigo and M. Casanovas, "A new minkowski distance based on induced aggregation operators," *International Journal of Computational Intelligence Systems*, vol. 4, no. 2, pp. 123–133, 2011.

[48] H. Bhavsar and M. H. Panchal, "A review on support vector machine for data classification," *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, vol. 1, no. 10, pp. 185–189, 2012.

[49] S. Zhang, C. Tjortjis, X. Zeng, H. Qiao, I. Buchan, and J. Keane, "Comparing data mining methods with logistic regression in childhood obesity prediction," *Information Systems Frontiers*, vol. 11, no. 4, pp. 449–460, 2009.

[50] T. Rymarczyk, E. Kozłowski, G. Kłosowski, and K. Niderla, "Logistic regression for machine learning in process tomography," *Sensors*, vol. 19, no. 15, p. 3400, Aug 2019. [Online]. Available: http://dx.doi.org/10.3390/s19153400

[51] A. A. Elkhail, R. U. D. Refat, R. Habre, A. Hafeez, A. Bacha, and H. Malik, "Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses," *IEEE Access*, vol. 9, pp. 162 401–162 437, 2021.

[52] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *Def Con*, vol. 24, no. 8, p. 109, 2016.

[53] D. Püllen, N. A. Anagnostopoulos, T. Arul, and S. Katzenbeisser, "Using implicit certification to efficiently establish authenticated group keys for in-vehicle networks," in *2019 IEEE Vehicular Networking Conference (VNC)*, 2019, pp. 1–8.

[54] K. Verma, M. Girdhar, A. Hafeez, and S. S. Awad, "Ecu identification using neural network classification and hyperparameter tuning," in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, 2022, pp. 1–6.

[55] A. Hafeez, J. Mohan, M. Girdhar, and S. S. Awad, "Machine learning based ecu detection for automotive security," in *2021 17th International Computer Engineering Conference (ICENCO)*, 2021, pp. 73–81.

[56] J. Brownlee, "Machine learning mastery with weka," *Ebook. Edition*, vol. 1, no. 4, 2019.

[57] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International journal of Remote sensing*, vol. 28, no. 5, pp. 823–870, 2007.

[58] R. M. Haralick and L. G. Shapiro, "Glossary of computer vision terms." *Pattern Recognit.*, vol. 24, no. 1, pp. 69–93, 1991.

[59] V. Lavrenko and C. Sutton, "Iaml: Dimensionality reduction," *School of Informatics*, 2011.

[60] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Josa a*, vol. 4, no. 3, pp. 519–524, 1987.

[61] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[62] A. A. Elkhail, R. U. D. Refat, R. Habre, A. Hafeez, A. Bacha, and H. Malik, "Vehicle security: A survey of security issues and vulnerabilities, malware attacks and defenses," *IEEE Access*, 2021.