An efficient and applicable physical fingerprinting framework for the controller area network utilizing deep learning algorithm trained on recurrence plots

Rafi Ud Daula Refat[0000−0003−3812−3244], Alireza Mohammadi[0000−0002−1089−3872], and Hafiz Malik[0000−0001−6006−3888]

University of Michigan - Dearborn, Dearborn MI 48128, USA
rerafi,amohmmad,hafiz@umich.edu

Abstract. The Controller Area Network (CAN) is widely used in the automotive industry for its ability to create inexpensive and fast networks. However, it lacks an authentication scheme, making vehicles vulnerable to spoofing attacks. Evidence shows that attackers can remotely control vehicles, posing serious risks to passengers and pedestrians. Several strategies have been proposed to ensure CAN data integrity by identifying senders based on physical layer characteristics, but high computational costs limit their practical use. This paper presents a framework to efficiently identify CAN bus system senders by fingerprinting them. By modeling the CAN sender identification problem as an image classification task, the need for expensive handcrafted feature engineering is eliminated, improving accuracy using deep neural networks. Experimental results show the proposed methodology achieves a maximum identification accuracy of 98.34%, surpassing the state-of-the-art method's 97.13%. The approach also significantly reduces computational costs, cutting data processing time by a factor of 27, making it feasible for real-time application in vehicles. When tested on an actual vehicle, the proposed methodology achieved a no-attack detection rate of 97.78% and an attack detection rate of 100%, resulting in a combined accuracy of 98.89%. These results highlight the framework's potential to enhance vehicle cybersecurity by reliably and efficiently identifying CAN bus senders.

Keywords: CAN, Deep leaning, Transfer learning, MobileNetV2, EfficientNet.

1 Introduction

One of the most popular in-vehicle networking protocol is called the controller area network (CAN) through which vehicle computing devices communicate with each other. The famous protocol was first introduced by Robert GmBH in 1983 and became a defacto for in-vehicle communication due to two specific reasons. 1) by design the protocol is applicable for hard real-time environments that guarantees communication with minimal time latency. 2) it reduced the wiring problem of a vehicle and was able to reduce the cost of vehicle manufacturing [1]. That is why the CAN bus protocol is used in all modern vehicles as the backbone of in-vehicle network communication.

By default, the CAN protocol is broadcasting in nature which means messages that are sent to the bus are accessible by all the entities connected to the network. It brings simplicity in terms of design but on the other hand the simplistic design can be leveraged by hackers [2, 3], as it lacks a basic security feature i.e. implementation of a message authentication mechanism which makes it vulnerable to a variety of spoofing attacks [4, 5]. In a single CAN message packet, a field that contains information of the source is absent. Because of the absence of the sender information, any electronic control unit (ECU) on the network can impersonate other ECUs in the network. An adversary can leverage that vulnerability of this protocol to launch various attacks leading to malfunctioning of the vehicle.

For example, in 2015 Charlie Miller and Chris Valasek remotely took control of a vehicle by injecting CAN data in the network. Surprisingly, the vehicle could not differentiate the impersonating CAN message and moved into a ditch [2]. Another demonstration was shown by the Keen Security Lab of Tencent team in 2016 where researchers remotely controlled a Tesla Model S. The researchers have gained entrance remotely by using Wi-Fi/Cellular as backdoor and was able to compromise many in-vehicle systems like IC, CID, and Gateway. Moreover, the team injected malicious CAN message into the network [3]. In December 2019, a gray-hat hacker created an android application that used an arduino microcontroller in order to inject CAN message to a Mercedes vehicle. The basic functionality of the application was to add features such as locking and unlocking doors, display custom text in instrument cluster, control hazard light etc. [4]. These evidences clearly indicate that the researchers took advantage of a known weakness of CAN protocol to spoof the network, i.e. the absence of source identification field.

To solve the above-mentioned security vulnerability, different approaches have been implemented by the security researchers [6, 7]. These solutions can be broadly categorized into two categories. (1) cryptography based solutions [8, 9], [10], (2) intrusion detection system based solution [11–13]. The traditional cryptography based solutions can provide some degree of security but they are computationally expensive and uses the network bandwidth which is critical for CAN based vehicle networks [14]. Moreover, these cryptography based solutions are vulnerable to replay attack [11]. Recently, researchers have proposed intrusion detection system based solutions for detecting CAN cyberattacks by implementing the famous physical layer identification [15] techniques [5],[14],[16].The fundamental idea of this approach is, the analog signal behaviors of data transmitters has slight variations which are introduced in the design, fabrication and manufacturing process. Researchers show that even manufactured in the same production lot, two same digital devices has unique artifacts in their signaling behavior which is difficult to control and duplicate [17]. Avatefipour et al.

was able to extract those unique artifacts and proposed a framework based on neural network for CAN sender identification by utilizing the extracted distortions [5]. Likewise, in the last 5 years researchers [14, 18, 46] have proposed a lot of frameworks that are effective in CAN transmitter identification. While, the frameworks offer high percentage of accuracy, but the core architecture of these methods depend on handcrafted feature engineering. As the approaches rely on neural network based methods, the feature engineering remains an essential step in testing phase of the framework. In some cases, the feature engineering becomes computationally so expensive, that the real time sender identification remains a challenge. So, here is the research question in this paper, ”Is it possible to identify the source of CAN message sender using an in-expensive approach that leverages deep neural network”?

In order to integrate a transmitter identification strategy to the existing CAN protocol, this paper proposes a framework that is based on the intersection between physical layer identification [15] and computer vision technique. To fingerprint, the proposed framework first extracts distortions from the analog signals sent by the ECUs. Then the distortions are converted into visual representation (images) by using recurrence plot technique [36] which are are distinctive in human eyes. To automate the process of ECU identification, the images are fed into deep neural networks (Mobilenetv2 [23] and EfficientNet [39]) to learn patterns of the signals from those generated images. Finally, the trained model is tested to evaluate the performance of the proposed framework. According to the evaluation, it achieves better accuracy in identifying the CAN senders and is lightweight in terms of computational cost.

The main contribution of the paper is as follows:

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

The rest of the paper is organized as follows. section 2 provides the background of CAN, then the state-of-the-art of CAN cybersecurity is presented in section 3. The methodology of our framework is presented in section 4. Section 5 describes the experimental result of the framework and finally, the paper is concluded with the conclusion / future work section that is followed by acknowledgement section.

2 Background

2.1 Controller Area Network (CAN)

In this subsection, the background of controller area network (CAN) is presented. To highlight the overview of CAN protocol, the protocol characteristics and its representation in terms of OSI model [22] is described here. Moreover, the security issues originated from the basic architectural design of the protocol is described at the end of the subsection.

By design the controller area network (CAN) is a broadcasting protocol where ECUs communicate with each other using a single wire. This enables the system manufacturer to reduce complex wiring design of many point to point connections between ECUs and make the system easily maintainable [20]. While connected to a standard CAN network, an ECU can send 0-8 bytes of data with an eleven bit identifier. The identifier maintains the priority scheme of CAN protocol which is, message with lower arbitration ID has high priority while going through the bus [17]. On the other hand, any entity connected to the bus can listen to all the traffic in the network for its broadcasting nature.

The CAN protocol is specified in ISO 11898 and and is defined in the physical layer and data link layer of Open Systems Interconnection model (OSI model) [22]. In the CAN physical layer, the data is handled as binary bits and the core functionality of this layer is to ensure bit encoding/decoding, bit synchronization and indicate physical wire orientation and on the other hand, CAN data link layer handles CAN data as frames and performs complex tasks like data encapsulation, frame encoding, frame error detection [22]. Physically, the CAN bus is actually a twisted pair wire, terminated with 120 ohm. The twisted pair is called the CAN high (CAN-H) and the CAN low (CAN-L) and provides protection against electromagnetic interference. In terms of physical layer orientation of OSI model, CAN protocol follows differential signaling (shown in figure 1) where the final voltage of a single bit data is extracted by subtraction between CAN-H and CAN-L. When there is a 0 bit in the bus (dominant bit), CAN-H pulls 3.5 volt where CAN-L contains 1.5 volt. In terms of a bit with value 1 (recessive), CAN-H and CAN-L both set the voltage to 2.5.

In data link layer, a CAN protocol handles data as frames. By default a standard CAN packet has 108 bits in total as shown in Table 1. It starts with a single bit of data called start of frame (SOF) field. Then it is followed by 11 bit arbitration ID (AID), 1 bit remote transmission request (RTR), 6 bit control field, 0-64 bits of data field, 16 bits cyclic redundancy check (CRC), 2 bits acknowledgment (ACK) field, 7 bits of end of frame (EOF) field [1]. While connected in a network, an ECU can send CAN packet to the traffic by sending a CAN data frame by putting dominant bit in the RTR field and an ECU can request data from another ECU by sending a CAN remote frame with a recessive bit in the RTR field. Although there is an AID field presented in a CAN packet, but there is not a single field available that indicates the source address. There is CRC field in a CAN packet which only protects the data field. So, the absence of source field and the broadcasting nature of the protocol clearly

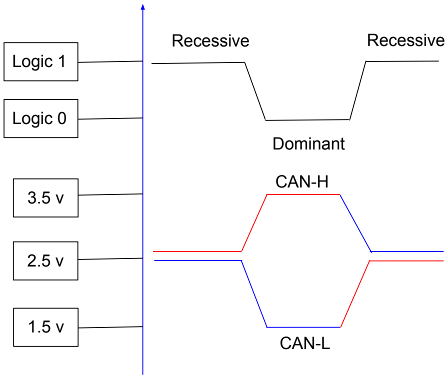


Fig. 1. CAN differential signaling

indicates that the CAN protocol lacks one of the concepts of the famous CIA triad (confidentiality, integrity and availability) i.e. integrity. The work proposes a framework to identify senders thus ensuring integrity to make CAN network security proven.

Table 1. A standard CAN data packet

Field name	**Number of bits**
Start of frame	1
Arbitration ID	11
Remote transmission request	1
Control fields	6
Data field	0 - 64
Cyclic redundancy check	16
Acknowledgement	2
End of frame	7
Total	108

3 Sender identification: state-of-the-art

To ensure integrity in the CAN bus one approach is to implement message authentication scheme [42] by including a message authentication code (MAC) inside CAN frame. While it makes the CAN bus secure but according to the standards, the least size of the MAC is 64 bit to prevent collisions [14]. So, the challenge of implementing the MAC based approaches is to add 64 bit MAC along with the data that needs to be transported to the network where the data

Table 2. Computational complexity of common state-of-the-art statistical features

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

field can only hold up to 64 bits of data 1. To overcome the approach, researchers proposed two kind of MAC implementations. one is instead of using 64 bit MAC, they were using a truncated MAC to include integrity to CAN protocol [26, 27, 42] and the other approach is to use CAN+ protocol, an improvement of the existing CAN [29, 30] where additional data can be sent in time intervals to authenticate CAN messages. For example, researchers in crafted a 4 byte MAC and put it into the data field of the CAN packet to authenticate CAN message. The disadvantage of truncating CAN data field to include MAC [26, 27] is, it limits the size of data payload to be transmitted in a CAN packet and restrict the CAN protocol to transmit 8 bytes data payload. The proposed works in [29] sends two CAN messages where one contains the data payload the other one contains the MAC address. The approach resolves the issues originated by the truncated MAC approaches but it uses the limited traffic bandwidth of CAN network (1 Mbit/s) [5] as it needs to send two packets of data to securely send a single CAN data payload.

Apart from the CAN message authentication techniques, researchers have considered to fingerprint CAN senders by using physical unclonable characteristics such as clock skews [31] and voltage [5, 14, 18]. The main idea of this approach is to identify the source of CAN transmitters. The concept is adopted from the famous physical layer identification (PLI) [15] technique where the unique characteristics of transmitters are extracted to link the physical signals to the senders. The techniques for CAN PLI can be classified into two categories.

Clock skew based fingerprinting The quartz crystal clock determines the different clock frequencies on an ECU, resulting in random clock drifts which can be used to uniquely identify an ECU. Cho and Shin proposed a Clock-based IDS (CIDS) [31] which exploits the intervals of periodic message to estimate the clock skews as the fingerprint of the transmitter ECU. The idea was used to estimate clock behaviors of ECUs to detect the intrusion and identify the source of the message. However, this method is effective in a temperature-stable environment[32].

Voltage based fingerprinting Authenticating the CAN message transmitter based on the unique and immutable physical characteristics such as the voltage, is termed as physical fingerprinting. This area of research has gained popularity now a days where utilizing the voltage characteristics is the core idea. For example, Kneib et al. [32] used voltages for fingerprinting ECUs, utilizing rising edge, falling edge of the dominant bits. The framework achieved an accuracy of 99.85% in identifying ECUs by using statistical features like mean, standard deviation, variance, skewness, kurtosis, root mean square, maximum and energy etc. Researchers in [5] extracted time domain and frequency domain statistical features using voltages captured from the ECUs and proposed a neural network based ECU classifier. They achieved an accuracy of 98.3% on an experimental setup using microcontrollers. Authors in [14] proposed an edge based identification method using voltage collected using picoscope (software defined oscilloscope) and a naive bayes classifier. As a feature they used statistical time domain features such as mean, variance, skewness, kurtosis, radio max plateau, plateau, overshoot height, irregularity, centroid, flatness, power and maximum. Similar work has been proposed in [33] that uses 10 time domain features and 10 frequency domain features and achieved an accuracy of 98.94 % accuracy at maximum while voltage data is collected using an oscilloscope at a sampling rate of 2 GS/s. Bellaire et al. [18] proposed a machine learning based ECU fingerprinting framework by handcrafting signal processing features on voltage data such as transient response length, maximum transient voltage, energy of the transient period, average dominant bit steady-state value, peak noise frequency and average noise. Similar kind of approaches are also proposed in [34, 35].

The research works described above achieved high accuracy in identifying CAN signal senders, but the feature extraction is highly expensive in terms of computational complexity. Table 2 represents the common statistical features and their corresponding computational cost. To overcome this, this paper proposed a novel framework that eliminates the necessity of extracting highly computational statistical features described above by utilizing images generated from the uniqueness presented in the voltage data to identify CAN signal transmitter. The image is generated using recurrence plot method whose computational complexity is $\Theta(n^2)$ whereas the computational complexity of any framework that uses feature shown in table 2, is $3 * (\Theta(n^2) + \Theta(n))$. Experimental result shows that the proposed framework processes features to identify ECUs with a lower computational time than the state-of-the-art work.

4 Methodology

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

4.1 Linking CAN signal to the transmitter

Physical layer identification is a popular concept for identification of senders in connected networks for so many years [15, 37]. The fundamental idea of this approach is, the behavior of senders in terms of analog signal has slight variations. The differences are introduced in the design, fabrication and manufacturing process, even two identical digital devices that are manufactured in the same production lot, have unique artifacts which is difficult to control and duplicate [5, 14, 32]. In a practical world, although it can be reproduced by reverse-engineering, but the process is difficult if not impossible for a determined attacker. Fig. 2 illustrates the amount of inherent variation between two different CAN transmitters.

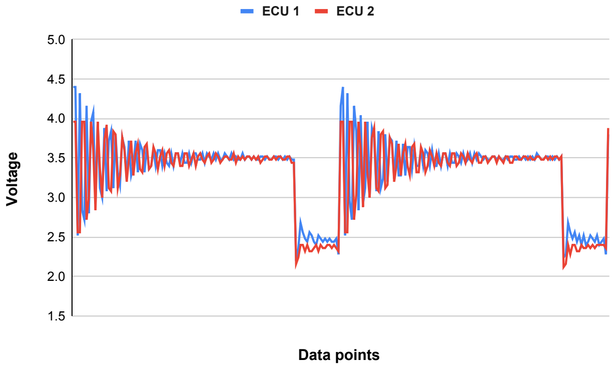


Fig. 2. Analog signal difference of two ECUs

The paper uses the above mentioned inherent variation of the CAN transmitter and uses them to fingerprint the transmitter as it is unique. The Figure 3 shows how a CAN signal stays in an idea condition and how it distorts in practical world. The spikes from the idea line is considered as the impurity of each CAN transmitter which is identical to it. The proposed work uses it to create a unique signal characteristics profiling for transmitters.

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

$$V = (V_1, V_2, V_3....V_n) \tag{1}$$

Ideally, V_i should be 3.5 when it is a dominant bit and 2.5 when it is a recessive bit. In real world, the unique artifacts add noise to the ideal value and creates spikes (see 3). In order to extract the unique variations, the spiking points needs to be subtracted from 3.5 or 2.5 depending on it is a dominant or recessive bit. So, the unique artifacts (Distortions, D_i) of an ECU is,

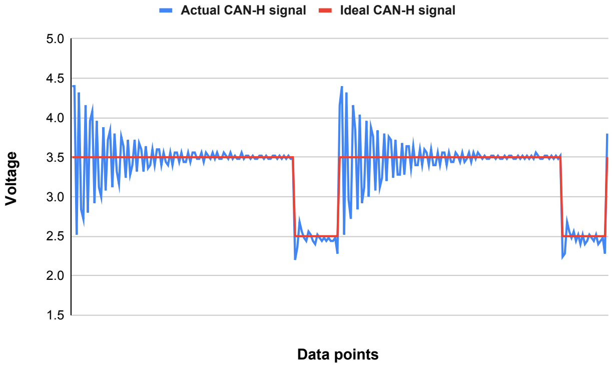


Fig. 3. CAN-H signals with unique artifacts

$$D_i = (V_i - T_j) \tag{2}$$

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

4.2 Representing signal profiling by recurrence plots

In this phase of the proposed method, the extracted unique slight variations of CAN senders are used to create images for each transmitter. The variations are turned into images via the recurrence plot (RP) [36] technique where each image represents the pattern of a sender. The initial purpose of recurrence plots was to create a visualisation of the recurrences of a system's states in a phase-space (with dimension n) within a small deviation ϵ. That means, a recurrence of a state at time i and at a different time j is marked within a two-dimensional squared matrix with ones and zeros (black and white points in a plot), where both axis represent time. The RP can be formally expressed by the following matrix in equation 3 [36].

$$R_{ij} = \Theta(\epsilon - |\overrightarrow{D_i} - \overrightarrow{D_j}|) \qquad i, j = 1....n \tag{3}$$

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

4.3 Source identification of ECUs

Finally, the paper proposes a framework that uses the above mentioned steps to identify the source of CAN ECUs. Figure 4 shows overall architecture of our source identification framework. The proposed architecture first extracts the unique artifacts of the CAN transmitters. Then the recurrence plots are created from the extracted distortions which are a representation of uniqueness of CAN transmitters. Finally, the recurrence plots are used to train and test a deep learning model. In the figure 4 the green line shows the training phase and the red line represents the testing phase of the system. The data processing needed for the framework is to extract the unique artifacts and the creation of recurrence plot, which is same for both the training and testing phase. For the selection of deep learning architecture, we have chosen two popular network MobileNetV2 [38] and EfficientNet [39] for the experiment.

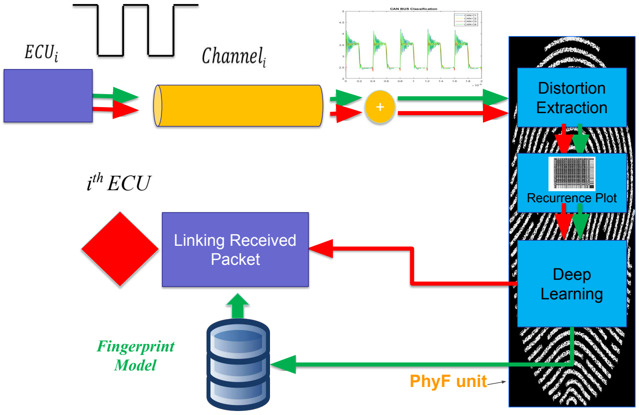


Fig. 4. Source identification of CAN message senders

5 Experimental result

In this section the effectiveness of the proposed methodology is evaluated by conducting experiments in the laboratory where subsection *5.1* presents the experimental setup. It is followed by subsection *5.2*, *5.3* i.e. the generation of recurrence plot of CAN senders and the performance of the proposed framework consecutively. Then the effect of environmental factors over the proposed framework is discussed in subsection *5.4* and effect of information aware downsampling is presented in subsection *5.5*. Finally, the performance of spoof detection in a vehicle test bench and the comparison with the state-of-the-art is presented in subsection *5.6* and *5.7* respectively to conclude the section.

5.1 Experimental setup

To verify the effectiveness of the proposed framework, an experiment is designed (Figure 5) on a CAN protocol based test bed that has total 8 ECU built by Arduino Uno microcontrollers connected to CAN transceivers where each ECU has 1 meter channel length. Physical signal for each ECU is captured using a DSO1012A oscilloscope with a sampling rate of 2GS/s, 100MHz bandwidth, and 8-bit vertical resolution. The data was collected in a laboratory environment from the CAN-H pin which ideally ranges from 3.5v to 2.5v. Multiple programming languages are used in this experiment as the microcontrollers are programmed using C programming language and Python is used for training & testing deep learning models and result analysis. The experimental testbed is set up in a plug and play mode, because some of the experiments were done using 4 ECUs and some of the experiments were conducted using 8 ECUs. To check the performance of the proposed framework on a real vehicle, an experiment is designed on a vehicle test bench that is based on the GM Sierra 2020 model for spoof detection also (elaborated extensively in subsection *5.6*).

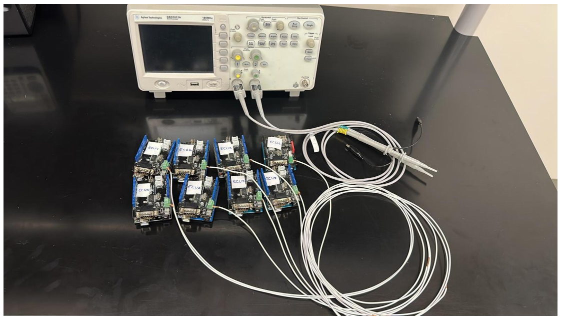


Fig. 5. Experimental settings

5.2 Generation of recurrence plot of CAN senders

The goal of this subsection was to create recurrence plot by using distortions captured from the CAN transmitters and visualize them in human eyes. To perform that experiment we collected analog signals from 4 ECUs using the testbed described in subsection *5.1* and extracted the distortions of the ECUs. The distortions are mapped to create recurrence plot and saved as images for visualization. Each image was generated from 96 voltage data points (length of CAN-H dominant bit) started from the peak of the voltage signals. Figure 6 shows the generated plots of 4 ECUs where each row has images for each ECU. It indicates that, the images has their own patterns and they are different to each other visually.

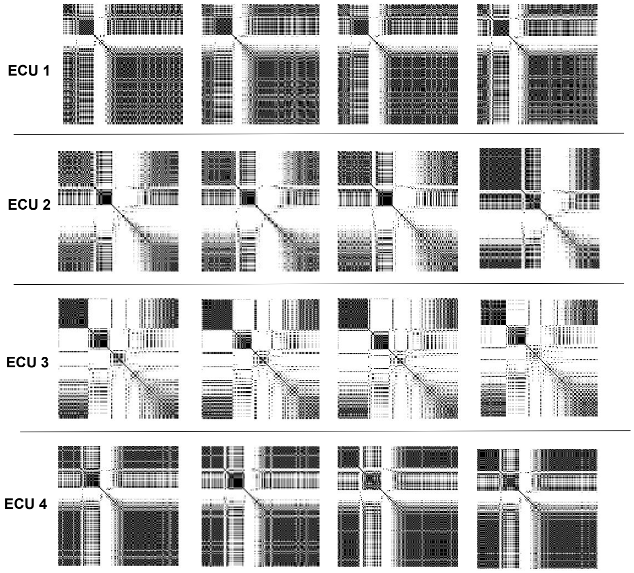


Fig. 6. Recurrence plot representing 4 ECUs

Visually RPs provide some useful insights about the sender ECUs. But the question arises how much information the RPs contain to distinguish the CAN transmitters. In order to do so, an experiment was deigned to quantify the RPs generated from the CAN ECU signals. To achieve that, a recurrence quantification analysis was performed to quantify the RPs by extracting recurrence properties from the generated RPs. We used recurrence parameters [45] such as recurrence rate, entropy diagonal lines, longest diagonal line length, laminarity, divergence, additional diagonal line length to perform data analysis. To do so, a Python program is written to generate RPs from CAN high analog voltages collected from 8 ECUs and then the images are used to perform recurrence quantification analysis (RQA) in an Apple M1 chip computer with 8 GB RAM. For RQA, the images are fed into python library PyRQA [43] and the parameters are extracted for rigorous analysis. To see the feature differences of the ECUs in terms of RQ parameters the data is plotted in a box plot shown in Figure 7. The figure shows that the 8 ECUs have notable variations when compared against the 6 recurrence parameters.

5.3 Performance of the proposed framework

This experiment evaluates the accuracy of sender identification in a CAN network using the proposed framework. The main goal of this subsection is to demonstrate the applicability of image classification via deep learning models in CAN physical fingerprinting. To achieve this, the proposed framework is validated against deep learning networks using (Mobilenetv2 [23] and EfficientNet

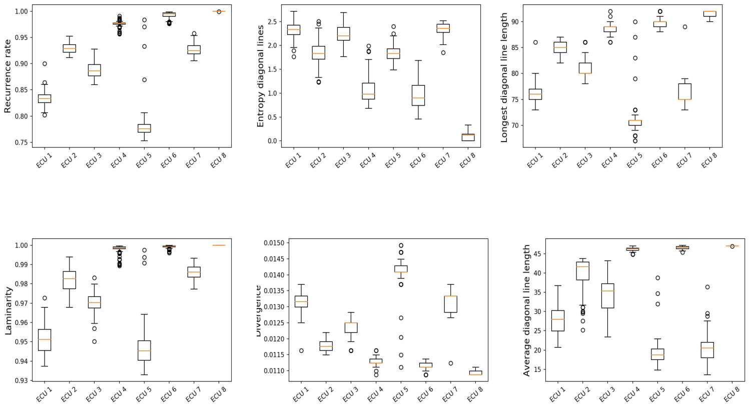


Fig. 7. Feature differences of recurrence quantification parameters

[39]) architecture. First the data from 8 ECUs are collected from the testbed described in subsection *A*, then data processing which involves extraction of distortion & image generation is done in an Apple M1 computer with 8 GB RAM and finally, the training and testing of deep learning architecture is performed in a Google-Colab environment. The code for the data processing, model training and model testing is written in Python programming language.

Table 3. CAN sender identification using Deep learning models

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

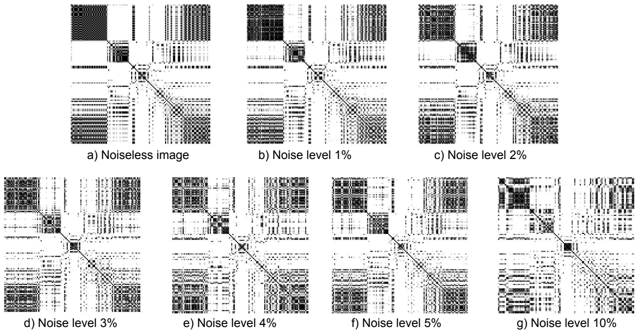
To conduct the experiment 131,760 analog voltage data points are gathered in total and 1098 images were created as described in subsection *B* where each ECU has 250 images and each image has a dimension of 192X192 pixels. To handle the smaller number of images we used transfer learning [41], where a trained model is tuned to solve a problem which is unknown to the trained model. In order to do so, a pre-trained MobileNetV2, trained with a public dataset (imagenet [40]) with 1,700,505 parameters is selected and it's weights are used to retrain the model to solve the CAN sender identification problem. For retraining the model, 70% data was used, while the retrained model is tested with 15% and validated with remaining 15% data. Table 3 shows the result of the simulation. It indicates, Mobilenetv2 architecture achieves a maximum validation accuracy of 97.52%. To check the performance of the proposed methodology while using a different deep learning algorithm, the same experiment was repeated using an

EfficientNet model pre-trained on imagenet dataset [40] with 5,338,572 parameters. It was re-trained using 70% data and tested using 15% data where the image dimension was 224X224 pixels. Finally, the performance of the methodology was measured by validating the trained model with 15% remaining data. The experimental result shows that the EfficientNet achieves a validation accuracy of 95.04%. While using both MobileNetv2 and EfficientNet it takes 0.07 ms per image on average for data processing task which involves noise extraction and image creation.

5.4 Effect of Environmental factors on the proposed IDS

This subsection represents the analysis of the effect of environmental conditions on the performance of the proposed framework. It is important because, the foundation of the proposed methodology is image classification where the images are created from the distortions present in electrical signals and the signal characteristics are sensitive to environmental factors like temperature, amount of moisture contamination, aging, etc. [21]. These factors, if not accounted for, could lead to incorrect identification of the senders in real-world scenarios. In order to verify their effect, data is collected from a setup testbed with 8 ECUs and then analysis is performed to measure performance of the proposed framework under the presence of noise that may be produced by environmental factors. To add the noise, a simulation is created by adding Additive White Gaussian Noise (AWGN) [44] of different percentage of the voltage distortions (1%, 2%, 3%, 4%, 5% & 10%) to the voltage signals. The overall experiment is divided into two different steps, first one is adding AWGN to the electrical signals and creating the images by using the noisy distortions. Figure 8 shows the distorted images that are generated from different level of noisy voltages. in the subfigure (a) represents an image generated without AWGN, while subfigure b,c,d,e,f,g represents images with 1%, 2%, 3%, 4%, 5% & 10% added AWGN. The noiseless and noisy figures clearly indicates that the noises caused by environmental factors has significant effect on the images generated by the voltage distortions while there is deviation from the original image increases with the addition of level of noises to the voltages.

In the second step of the experiment, a deep learning model is trained using the noiseless original noise, while the images with noise is tested against the trained model and the model performance is evaluated in terms of sender identification accuracy (shown in Table 4). While introducing 1% noise in the testing data the performance of the proposed framework degrades by a 30.23% so the proposed model is sensitive to environmental noise. To check the performance of the proposed approach when the model is retrained, again the trained model is retrained by adding images with 1% AWGN and tested against noisy images. Later the trained model was retrained with 2% noise and model performance against noisy images was evaluated again. The experimental result is summarized and shown in Table 4. According to it, when the generated images with 1% AWGN are introduced during the model training with noiseless images, testing accuracy improves significantly for noisy images (1%, 2%, 3%, 4%, 5% and



a) Noiseless image b) Noise level 1% c) Noise level 2%

d) Noise level 3% e) Noise level 4% f) Noise level 5% g) Noise level 10%

Fig. 8. Images generated from voltages under different environmental conditions

10% GN). Although the training data had only noisy images with low AWGN (1%), the trained model was able to classify noisy images with an improvement of maximum 34.47% and minimum 19.37%. Again, when the model was again retrained by introducing noisy images with 2% AWGN and the model can identify senders with an maximum upgrade of 9.2% in terms of accuracy. So, it can be concluded that, the proposed framework is performs better if the model is retrained with noisy images.

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

Since, the amount of data to be processed for generating each image has a larger influence on the required computing power, a major goal is to reduce the required amount of sampling points. To reduce the sampling points considered to create the image, an experiment with rigorous analysis is conducted. If we look carefully, the backbone of the methodology is the images which are created from the distortions of the ECUs. Again, the distortions are created from analog voltage signal of the CAN signals. Figure 9 shows the plot of analog signal captured form an ECU and it is clearly visible that, the signals has spikes at the beginning and gradually it settles down in terms of voltage. From that we can infer that, the distortions which is extracted from the overshoot portion of analog signals (marked as a red box in Figure 9) holds significant unique information which is vital in sender identification. and after that we have data points that are less informative. Based on that observation, an experiment was conducted where images were generated by varying the informative and uninformative data points. Then the images are fed into a MobileNetV2 model for evaluation. So, in order to verify that an simulation is designed where images generated by three approaches are tested against MobileNetV2 model and the validation accuracy are evaluated. They are,

- **Truncated sampling**: images generated using all the informative points .

Table 4. Effect of environmental factors over the proposed framework

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

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

Table 5. Performance analysis on information aware down sampling

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

To conduct the above mentioned experiments, analog voltage samples for 8 ECUs are collected using the experimental setup described in subsection A and images are generated using the truncated sampling, the custom odd sampling and the 5th sequence sampling methods. The images are then trained, tested and validated with the MobileNetV2 deep learning architecture and evaluated based on validation accuracy that is shown in Table 5. According to the analysis, 5th sequence sampling achieved a validation accuracy of 98.34% which is better than

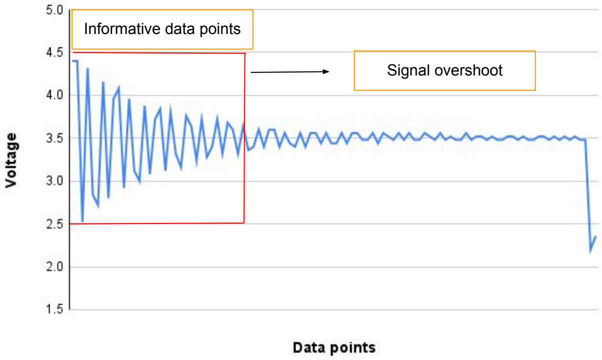


Fig. 9. Information aware down sampling technique

the accuracy achieved while using truncated sampling and custom odd sampling. Where the truncated sampling is faster than the other two approaches in data processing by at least .01 ms.

5.6 Spoof detection on actual vehicle test bench

To evaluate the performance on an actual vehicle test bench, an experiment is conducted where the goal is to detect spoofing attack using the proposed methodology. First an experiment is setup on a laboratory vehicle test bench (Model: GM Sierra, Year: 2020) shown as Figure 10 to perform spoofing attack by changing the vehicle gear from park to drive mode using a raspberry pi 4 model B, where analog voltage data is captured using a picoscope 2205 A with a sampling rate of 25 MS/s. After that, analog voltage data is captured again with the same sampling rate of 25 MS/s when the vehicle is put to drive mode from park mode using the vehicle gearshift. Although the attacker ECU was sending the same data we can see form Figure 11 the data send by the attacker has different analog voltage profile than the authorized ECU. The Figure 11 shows that the two sets of benign CAN-H signal data from authorized ECU, marked in blue & orange and captured at different times, differ from the two sets of CAN-H data from the attacker, marked in green & purple and captured at different times. However, when sent from the same CAN ECU, the data from both the authorized ECU and the attacker appear almost identical. It is clear form the figure that, the ECUs has their own fingerprint in their CAN-H dominant bit analog data which is different from 3.5v (marked as red in Figure 11).

The analog voltages then prepossessed in a computer with 8 GM RAM and 416 images are generated from the distortions for both ECUs using Python programming language, where 240 were from authorized ECU and 176 were from attacker ECU. As the dataset is comparatively small we did data augmentation technique to flip each images from left to right and prepared a data size of 832

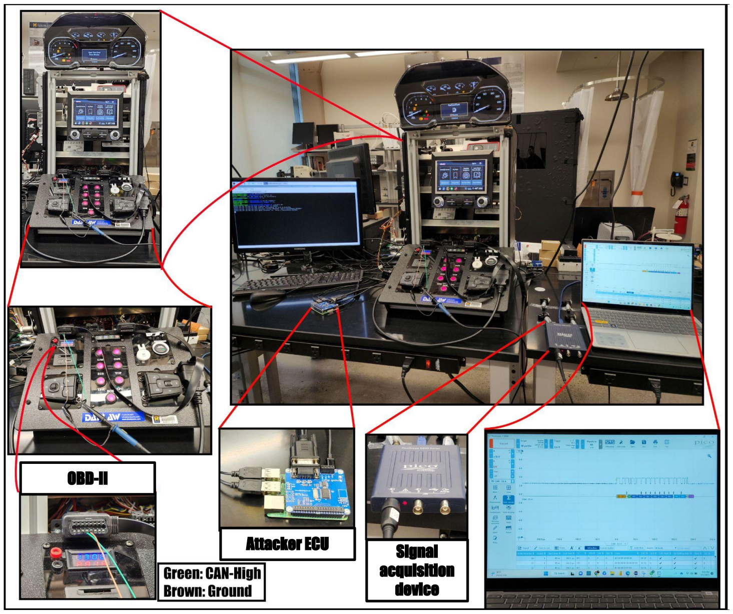


Fig. 10. Spoofing attack on vehicle test bench

images. The images are then retrained using a pre-trained MobileNetV2 deep learning architecture where 70% data are used as training, 15% for tuning the model and 15% for testing the trained model. The performance of the proposed methodology is evaluated against metrics such as (1) attack detection accuracy, precision, recall, f1-score and (2) benign data detection accuracy, precision, recall and f1-score. The simulation result is summarized in Table 6 and it shows that the proposed methodology detects spoofing attack with 100% accuracy while it achieves a precision of 96.72%, recall of 100% and f1-score of 98.33%. On the other hand, it detects benign data with an accuracy of 97.78%, precision of 100%, recall of 100% and f1-score of 98.33%. So, finally, it can be concluded that, the proposed methodology is efficient and applicable to today's vehicles as it achieved an combined accuracy of 98.89% in detecting spoofed and benign data in the vehicle test bench.

Table 6. Performance under spoofing attack on laboratory vehicle bench

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

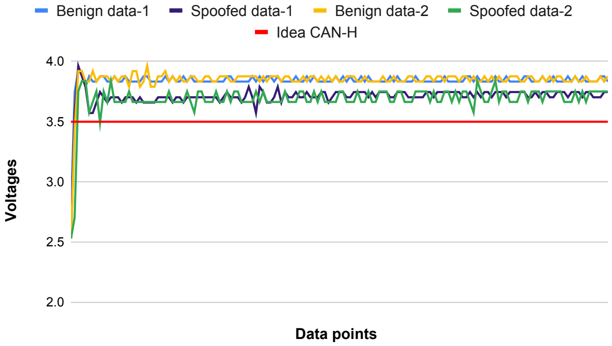


Fig. 11. Analog voltage data for an authorized ECU and an attacker ECU

Table 7. Comparison with the state-of-the-art

Approach	**Accuracy(%)**	**Precision (%))**	**Recall(%)**	**F1 (%))**	**time (ms)**
[5]	97.13	97.00	97.00	95.75	1.35
[14]	89.24	89.63	89.63	89.38	0.95
KNN	94.97	95.00	95.38	95.13	238
SVM	95.82	95.75	96.13	95.75	238
Proposed IDS	98.34	98.63	98.25	98.38	0.05

5.7 Comparison with the state-of-the-art

Finally the proposed methodology is compared against the state-of-the-art sender identification methods [5] and [14] where [5] trains a Neural Network and [14] builds a Support Vector Machine (SVM) model using statistical features. In order to do so, the state-of-the-art [5] methodology extracts the distortion of the ECUs at first and handcrafts 11 statistical analysis based features including 6 time domain features i.e. maximum, minimum, mean, variance, skewness, kurtosis and 5 frequency domain features i.e. spectral standard deviation, spectral kurtosis, spectral skewness, spectral centroid, irregularity k to feed into an artificial neural network to evaluate the performance of their approach. While simulating their approach with data gathered from 8 ECU using the experimental setup described in subsection A, an validation accuracy of 97.13% as achieved while the proposed framework achieved 98.34%. But in terms of data processing (feature engineering), the state-of-the-art takes 27 times more than the proposed framework. The proposed methodology creates the a single recurrence plot for testing the model in 0.05 ms time using the 5th sequence information aware down sampling while, the state-of-the-art [5] generates 11 statistical features in 1.35 ms for identifying a single CAN sender which is computationally more expensive (see Table 7). Again the proposed approach is compared against [14] where the authors use signal characteristics by extracting 12 features such as maximum, mean, variance, skewness, kurtosis, centroid, flatness, power, irregularity, plateau, max plateau ratio and overshoot height to train machine learning model. While training & testing a SVM model, the state-of-the-art degrades by 9.1% in terms of sender identification accuracy than the proposed methodology and needs 0.95 ms to process features which is 19 times slower than the proposed approach. In addition to that, the proposed approach is compared with traditional machine learning based approaches i.e. K-Nearest Neighbors(KNN) and SVM where recurrence quantification parameters are used as features to fit machine learning models. Based on the data collected form 8 CAN ECUs, KNN achieved an accuracy of 94.97% and SVM achieved an accuracy of 95.82% while the feature generation took around 238 ms. It clearly shows that the proposed methodology is better both in terms of accuracy of identifying senders and feature processing time.

6 Conclusion and Future Work

The paper proposed a physical fingerprinting framework for solving the CAN protocol's inability to identify the sender by modeling the problem as an image classification problem. It introduces a novel approach for creating images utilizing uniqueness of the analog signal of CAN senders and it classifies the images using deep learning model to identify CAN ECUs. As, the proposed method only requires to generate an image for the identification of ECUs, it can put an end to the trend of handcrafted feature engineering process in CAN physical fingerprinting. As per contributing to the state-of-the-art, to the best of our knowledge the proposed methodology is the first ever work that utilizes the concept of computer vision in CAN sender identification problem. The experimental result shows that it is effective as it achieved an accuracy of 98.34% and efficient as it only requires an image to identify sender. In the future, we will investigate the effect of environmental factors like aging, temperature etc. on the proposed methodology as analog signals change by time and varying temperature.

7 Acknowledgement

This research work was supported by the National Science Foundation under the award # CNS-2035770 and award # 2414729.

References

1. Elkhail, A. A., Refat, R. U. D., Habre, R., Hafeez, A., Bacha, A., & Malik, H. (2021). Vehicle Security: A Survey of Security Issues and Vulnerabilities, Malware Attacks and Defenses. IEEE Access, 9, 162401-162437.
2. Greenberg, A. (2015). Hackers remotely kill a jeep on the highway—with me in it. Wired, 7(2), 21-22.
3. Nie, S., Liu, L., & Du, Y. (2017). Free-fall: Hacking tesla from wireless to can bus. Briefing, Black Hat USA, 25, 1-16.

4. “Research.” Upstream Security, 3 Sept. 2022, https://www.upstream.auto/research/automotive-cybersecurity/?id=4710.
5. Avatefipour, O., Hafeez, A., Tayyab, M., & Malik, H. (2017, December). Linking received packet to the transmitter through physical-fingerprinting of controller area network. In 2017 IEEE Workshop on Information Forensics and Security (WIFS) (pp. 1-6). IEEE.
6. Halder, S., Conti, M., & Das, S. K. (2020, January). COIDS: A clock offset based intrusion detection system for controller area networks. In Proceedings of the 21st International Conference on Distributed Computing and Networking (pp. 1-10).
7. Jichici, C., Groza, B., Ragobete, R., Murvay, P. S., & Andreica, T. (2022). Effective Intrusion Detection and Prevention for the Commercial Vehicle SAE J1939 CAN Bus. IEEE Transactions on Intelligent Transportation Systems.
8. Ishak, M. K., & Khan, F. K. (2019). Unique message authentication security approach based controller area network (CAN) for anti-lock braking system (ABS) in vehicle network. Procedia Computer Science, 160, 93-100.
9. S. Nurnberger and C. Rossow, ”vatiCAN-vetted authenticated CAN bus”, Cryptographic Hardware and Embedded Systems: 18th International Conference Santa Barbara CA USA August 17–19 2016 Proceedings, 2016.
10. Y. Weisglass and Y. Oren, ”Authentication method for CAN messages”, ESCAR Europe, 2016
11. Islam, R., Refat, R. U. D., Yerram, S. M., & Malik, H. (2020). Graph-based intrusion detection system for controller area networks. IEEE Transactions on Intelligent Transportation Systems.
12. Lee, H., Jeong, S. H., & Kim, H. K. (2017, August). OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In 2017 15th Annual Conference on Privacy, Security and Trust (PST) (pp. 57-5709). IEEE.
13. Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., & Kadobayashi, Y. (2020). LSTM-based intrusion detection system for in-vehicle can bus communications. IEEE Access, 8, 185489-185502.
14. Kneib, M., Schell, O., & Huth, C. (2020, February). EASI: Edge-Based Sender Identification on Resource-Constrained Platforms for Automotive Networks. In NDSS.
15. Gerdes, R. M. K. (2011). Physical layer identification: methodology, security, and origin of variation. Iowa State University.
16. Hafeez, A., Topolovec, K., & Awad, S. (2019, December). 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) (pp. 29-38). IEEE.
17. Xu, T., Lu, X., Xiao, L., Tang, Y., & Dai, H. (2019, May). Voltage based authentication for controller area networks with reinforcement learning. In ICC 2019-2019 IEEE International Conference on Communications (ICC) (pp. 1-5). IEEE.
18. Bellaire, S., Bayer, M., Hafeez, A., Refat, R. U. D., & Malik, H. (2023). Fingerprinting ECUs to Implement Vehicular Security for Passenger Safety Using Machine Learning Techniques. In Proceedings of SAI Intelligent Systems Conference (pp. 16-32). Springer, Cham.
19. Hafeez, A., Ponnapali, S. C., & Malik, H. (2020). Exploiting channel distortion for transmitter identification for in-vehicle network security. SAE International Journal of Transportation Cybersecurity and Privacy, 3(11-02-02-0005), 5-17.
20. Refat, R. U. D., Elkhail, A. A., & Malik, H. (2023). Machine Learning for Automotive Cybersecurity: Challenges, Opportunities and Future Directions. AI-enabled Technologies for Autonomous and Connected Vehicles, 547-567.

21. Sierota, A., & Rungis, J. (1995). Electrical insulating oils. I. Characterization and pre-treatment of new transformer oils. IEEE Electrical Insulation Magazine, 11(1), 8-20.
22. Alani, Mohammed M. "OSI model." Guide to OSI and TCP/IP Models. Springer, Cham, 2014. 5-17.
23. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018, April). Mobilenetv2: The next generation of on-device computer vision networks. In CVPR.
24. Popa, L., Groza, B., Jichici, C., & Murvay, P. S. (2022). ECUPrint—Physical Fingerprinting Electronic Control Units on CAN Buses Inside Cars and SAE J1939 Compliant Vehicles. IEEE Transactions on Information Forensics and Security, 17, 1185-1200.
25. Lu, Z., Wang, Q., Chen, X., Qu, G., Lyu, Y., & Liu, Z. (2019, October). LEAP: A lightweight encryption and authentication protocol for in-vehicle communications. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 1158-1164). IEEE.
26. Hartkopp, C. R. O., & Schilling, R. nd." MaCAN-message authenticated CAN. In Proc. 10th Int. Conf. Embedded Security in Cars (ESCAR)(Ed.).
27. Hazem, A., & Fahmy, H. A. (2012, November). Lcap-a lightweight can authentication protocol for securing in-vehicle networks. In 10th escar Embedded Security in Cars Conference, Berlin, Germany (Vol. 6, p. 172).
28. Schmandt, J., Sherman, A. T., & Banerjee, N. (2017). Mini-MAC: Raising the bar for vehicular security with a lightweight message authentication protocol. Vehicular Communications, 9, 188-196.
29. Van Herrewege, A., Singelee, D., & Verbauwhede, I. (2011, November). CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus. In ECRYPT workshop on Lightweight Cryptography (Vol. 2011, p. 20). ECRYPT.
30. Groza, B., Murvay, S., Herrewege, A. V., & Verbauwhede, I. (2012, December). LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks. In International Conference on Cryptology and Network Security (pp. 185-200). Springer, Berlin, Heidelberg.
31. Shin, K. G., & Cho, K. T. (2021). U.S. Patent No. 11,044,260. Washington, DC: U.S. Patent and Trademark Office.
32. Kneib, M., & Huth, C. (2018, October). 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 (pp. 787-800).
33. Choi, W., Joo, K., Jo, H. J., Park, M. C., & Lee, D. H. (2018). Voltageids: Low-level communication characteristics for automotive intrusion detection system. IEEE Transactions on Information Forensics and Security, 13(8), 2114-2129.
34. Verma, K., Girdhar, M., Hafeez, A., & Awad, S. S. (2022, December). ECU Identification using Neural Network Classification and Hyperparameter Tuning. In 2022 IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 1-6). IEEE.
35. Ahmed, S., Juliato, M., Gutierrez, C., & Sastry, M. (2021). Two-Point Voltage Fingerprinting: Increasing Detectability of ECU Masquerading Attacks. arXiv preprint arXiv:2102.10128.
36. Marwan, N., Kurths, J., & Saparin, P. (2007). Generalised recurrence plot analysis for spatial data. Physics Letters A, 360(4-5), 545-551.
37. Mathur, S., Reznik, A., Ye, C., Mukherjee, R., Rahman, A., Shah, Y., ... & Mandayam, N. (2010). Exploiting the physical layer for enhanced security [security and

privacy in emerging wireless networks]. IEEE Wireless Communications, 17(5), 63-70.

38. Dong, K., Zhou, C., Ruan, Y., & Li, Y. (2020, December). Mobilenetv2 model for image classification. In 2020 2nd International Conference on Information Technology and Computer Application (ITCA) (pp. 476-480). IEEE.
39. Koonce, B. (2021). EfficientNet. In Convolutional neural networks with swift for tensorflow (pp. 109-123). Apress, Berkeley, CA.
40. You, Y., Zhang, Z., Hsieh, C. J., Demmel, J., & Keutzer, K. (2018, August). Imagenet training in minutes. In Proceedings of the 47th International Conference on Parallel Processing (pp. 1-10).
41. Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018, October). A survey on deep transfer learning. In International conference on artificial neural networks (pp. 270-279). Springer, Cham.
42. Schmandt, J., Sherman, A. T., & Banerjee, N. (2017). Mini-MAC: Raising the bar for vehicular security with a lightweight message authentication protocol. Vehicular Communications, 9, 188-196.
43. 8.0.0, P. Feb 21. (2021), https://pypi.org/project/PyRQA/
44. Aja-Fernández, S. & Tristán-Vega, A. A review on statistical noise models for magnetic resonance imaging. *LPI, ETSI Telecomunicacion, Universidad De Valladolid, Spain, Tech. Rep.* (2013)
45. Rawald, T., Sips, M. & Marwan, N. PyRQA—Conducting recurrence quantification analysis on very long time series efficiently. *Computers & Geosciences.* **104** pp. 101-108 (2017)
46. Hafeez, A., Ponnapali, S. & Malik, H. Exploiting channel distortion for transmitter identification for in-vehicle network security. *SAE International Journal Of Transportation Cybersecurity And Privacy.* **3**, 5-17 (2020)