CAN-LOC: Spoofing Detection and Physical Intrusion Localization on an In-Vehicle CAN Bus Based on Deep Features of Voltage Signals

Efrat Levy, Asaf Shabtai, Bogdan Groza, Pal-Stefan Murvay and Yuval Elovici

***Abstract*—The Controller Area Network (CAN), which is used for communication between in-vehicle devices, has been shown to be vulnerable to spoofing attacks. Voltage-based spoofing detection (VBS-D) mechanisms are considered state-of-the-art solutions, complementing cryptography-based authentication whose security is limited due to the CAN protocol's limited message size. Unfortunately, VBS-D mechanisms are vulnerable to poisoning performed by a malicious device connected to the CAN bus, specifically designed to poison the deployed VBS-D mechanism as it adapts to environmental changes that take place when the vehicle is moving. In this paper, we harden VBS-D mechanisms using a deep learning-based mechanism which runs immediately, when the vehicle starts; this mechanism utilizes physical side-channels to detect and locate physical intrusions, even when the malicious devices connected to the CAN bus are silent. We demonstrate the mechanism's effectiveness (100% intrusion detection accuracy and error rates of close to 0%) in various physical intrusion scenarios and varying temperatures on a CAN bus prototype. In addition, we present a deep learning-based VBS-D mechanism that securely adapts to environmental changes. This mechanism's robustness (99.8% device identification accuracy) is demonstrated on a real moving vehicle.**

***Index Terms*—Intrusion detection, CAN Bus, side-channel analysis, deep learning**

I. Introduction

The Controller Area Network (CAN) protocol has been widely adopted for real-time communication between electronic control units (ECUs) in modern vehicles [1], [2]. The CAN protocol was designed to provide a high level of fault tolerance, however less attention was paid to security issues (e.g., authentication), which were not a major source of concern when it was developed. These unaddressed security issues make the CAN protocol vulnerable to today's threats [3], [4], [5], such as spoofing attacks [6], [7], [8].

A common approach for mitigating spoofing attacks on the CAN bus is to add a cryptography-based authentication mechanism [9], [10]. However, limitations in the 64-bit payload of CAN messages make it difficult to embed cryptographic elements for a sufficiently high security level, resulting in the need to develop complement techniques. In other studies, Liu et al. [11], [12] propose a privacy-preserving trust evaluation scheme; that can realize trust management and conditional privacy preservation simultaneously with low communication overhead for facilitating distributed data fusion in cooperative vehicular safety applications.

Another means of coping with spoofing attacks is to authenticate connected ECUs by analyzing and modeling their communication on the CAN bus. This can be done by performing timing analysis, using various statistical and machine learning-based mechanisms [13], [14], [15], [16], [17] or by conducting payload-based analysis [18], [19], [20]. However, research has demonstrated that an attacker can evade detection by such mechanisms [21], [13]; the attacker can replicate the propagation delay behavior of a legitimate frame transmitter [13].

Taking the evasion constraint into consideration, previously proposed methods have statistically analysed the unique characteristics of voltage signals generated during transmission by each individual ECU in order to detect spoofing attacks [22], [23]. Compared to the timing- and payload-based methods, voltage-based spoofing detection (VBS-D) mechanisms are more difficult to evade. Although the software of an in-vehicle ECU can be remotely compromised, it is difficult to alter the voltage signal's characteristics, and hence the corresponding fingerprints in a controlled manner.

In a recent study [24], the researchers presented a novel technique to evade VBS-D mechanisms. In their work, they exploit the VBS-Ds' retraining process by connecting a malicious ECU, specifically designed for poisoning the VBS-Ds' models, to the CAN bus. Poisoning attacks against statistical models have also been extensively researched [25], [26], [27].

In this study we propose a novel mechanism which utilizes *physical side-channels* to detect and locate malicious intrusions, even when the connected devices are silent. Therefore, detecting and locating physical intrusions can be done by our proposed mechanism immediately when the vehicle starts.

The methods presented by Murvay et al. [13] represent a mechanism for physical intrusion detection and localization. In addition to the low accuracy reported in their work, there are two other drawbacks to their approach: (1) to detect and localize malicious intruders, the mechanism must rely on their CAN bus transmissions, and (2) the proposed mechanism is based on timing analysis and is thus susceptible to location evasion by malicious intruders [13].

In order to secure the CAN bus, we propose CAN-LOC, a security hardening system for in-vehicle networks, which monitors the voltage signals transferred on the CAN bus (illustrated in Figure 1). Our system consists of two mechanisms. The first

Efrat Levy, Asaf Shabtai and Yuval Elovici are with the Faculty of Information Systems Engineering, Ben-Gurion University of the Negev, Israel, Bogdan Groza and Pal-Stefan Murvay are with the Faculty of Automatics and Computers, Politehnica University of Timisoara, Romania. Email: elevy@post.bgu.ac.il, {shabtaia, elovici}@bgu.ac.il, {bogdan.groza, pal-stefan.murvay}@aut.upt.ro Corresponding author: Efrat Levy.

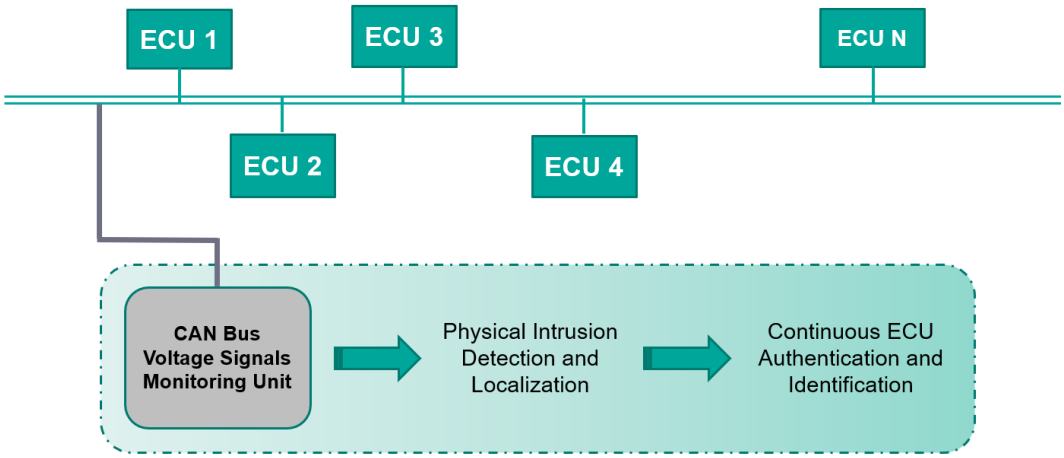


Fig. 1: Example of CAN bus line topology with a CAN bus voltage signals monitoring unit connected. This unit is responsible for sampling and analyzing voltage signals transferred on the CAN bus.

is a physical intrusion detection and localization mechanism; this mechanism uses a deep autoencoder that detects whether an additional ECU is introduced on the CAN bus, and a convolution neural network (CNN) multiclass classifier that reports the exact location in insertion scenarios. The second is a continuous ECU authentication and identification mechanism, which uses CNN binary classifiers and is capable of detecting spoofing attacks performed by legitimate ECUs that impersonate their peers.

From a practical standpoint, the proposed system is comprehensive in that the physical intrusion detection and localization mechanism runs immediately, when the vehicle is started, attempting to detect and locate changes that have been made to the original network's topology, and the continuous ECU authentication and identification mechanism runs continuously after the vehicle has been started. In this study, we show that the proposed physical intrusion detection mechanism is robust to environmental changes and present a complement VBS-D mechanism (i.e., the continuous ECU authentication and identification mechanism) which does not require the ECUs to communicate using MACs during the proposed VBS-D retraining.

Our system design is inspired by recent power analysis research in which classification using deep learning has been shown to be more powerful and robust to environmental conditions than other statistical methods [28], [29], [30]. In the course of our research, we derived the novel insight that information related to physical intruders and their location is encoded within the **legitimate** ECUs' voltage signals transferred on the CAN bus. Thus, our physical intrusion detection and localization abilities do *not* depend on malicious data transfers. In other words, our system is effective against sniffing devices connected to the CAN bus.

We validate the physical intrusion detection and localization mechanism on a CAN bus prototype using a large dataset of ECU replacement and insertion attacks, and show that our mechanism can locate the intrusion points with 100% accuracy. We demonstrate that the proposed mechanism can detect physical intrusions with +0.99 precision and 1 recall in various environmental conditions (e.g., at temperatures of 0°C, 24°C, 50°C, and 60°C).

We validate the continuous ECU authentication and identification mechanism on a CAN bus prototype and traffic recorded from a real vehicle, i.e., a 2015 Honda Civic. We report 99.8%-99.9% ECU identification accuracy.

The main contributions of this study are summarized as follows:

- We harden VBS-D mechanisms and present a mechanism that allows the detection and localization of physical intruders, even when they are silent.
- We perform a comprehensive evaluation of physical intrusion detection and localization on a CAN bus prototype, examining a wide variety of intrusion attacks and temperature variations.
- We extend the above mechanism and show a VBS-D mechanism (i.e., ECU authentication and identification mechanism) that securely adapts to possible environmental changes that occur when the vehicle is moving. We show that the mechanism's retraining procedure does not require the ECUs to communicate using MACs, serving as another layer of defense against spoofing attacks.
- We perform an evaluation of the authentication and identification mechanism on a moving vehicle.
- This research complements a recently published prevention solution that requires accurate localization capability [31].

II. Background

A. CAN Communication

The CAN bus is a two-wire broadcast bus which uses the differential voltage between the two bus lines, CAN-H and CAN-L, to encode the bits. During the dominant state, the CAN-H line is driven toward a nominal voltage of 3.5V, and the CAN-L line is driven toward a nominal voltage of 1.5V. The resulting differential voltage V_{diff} during the dominant

state must be within 0.9-2.0V, a case in which a ”0” is interpreted by the ECU transceiver. During the recessive state, both the CAN-H and CAN-L lines are driven toward a nominal voltage of 2.5V, and a ”1” is interpreted for a differential voltage less than 0.5V. An illustration of the differential voltage is presented in Figure 2.

Figure 3 presents a standard CAN frame structure. The CAN frame begins with the start-of-frame (SOF) bit, which is a ”0” bit that drives the CAN bus from the recessive state to a dominant state. The identifier field ID, which is used for arbitration, is next. Since multiple ECUs can write on the CAN bus at the same time, an arbitration mechanism is needed to avoid collisions. The arbitration mechanism is based on the message identifier (ID), which is the first field after the start of frame (SOF). Identifiers with lower values have the highest priority; note that dominant bits, i.e., zeros, will always overwrite recessive bits, i.e., ones. Several control fields follow the identifier field ID: the RTR bit, which signals remote frames; the IDE bit, which signals the extended identifier; a reserved field, which signals future extensions; and the DLC field, which represents the length of the data field. The latter, which represents the actual data, can occupy up to eight bytes. This field is followed by a 15-bit CRC field and a delimiter. The acknowledgement field, ACK, is written by all ECUs that successfully receive the frame. It is followed by a delimiter and the end-of-frame (EOF).

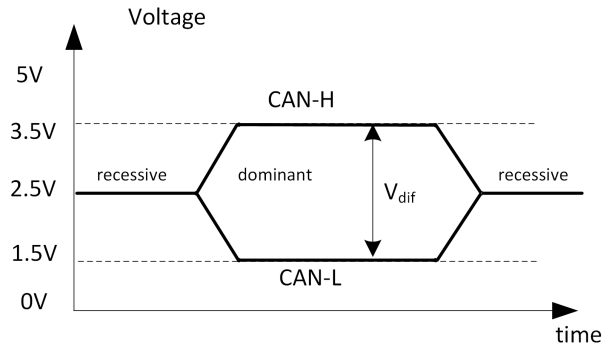


Fig. 2: Nominal voltage of the CAN-H and CAN-L lines during the recessive and dominant states.

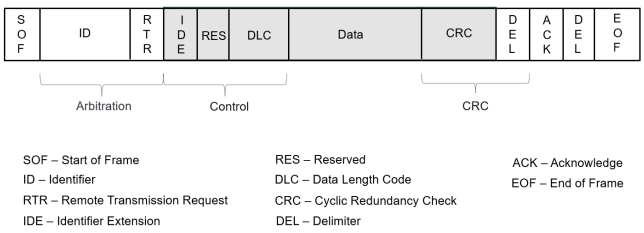


Fig. 3: Structure of a standard CAN frame.

B. ECU Voltage Signals

Modern vehicles contain a variety of ECUs, each of which generates unique analog signals. Even if the same CAN frames are transmitted by two identical ECUs manufactured in the same batch, their signals' characteristics are different. Recent studies showed that these characteristics are useful for highly accurate ECU fingerprinting [40], [39]. When analyzing the digital representation of a sampled signal, those differences are expressed in relatively minor changes. Figure 4 visually illustrates the difference between the signals of two ECUs, as sampled from the rising and falling edges of a CAN frame. Figure 5 visually illustrates how existing ECU signals are influenced when an ECU is added at different locations on the CAN bus.

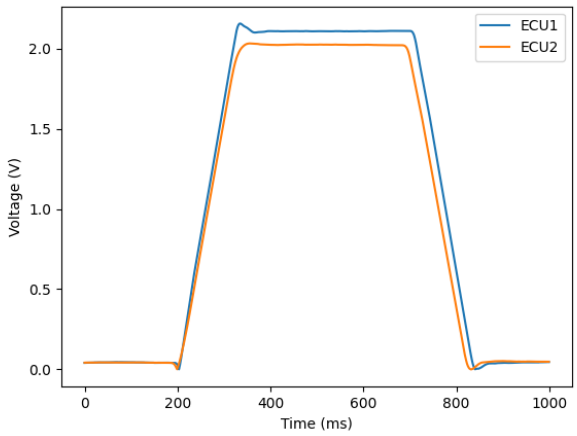


Fig. 4: A demonstration of a recessive ”1” to a dominant ”0” transition and return by two distinct ECUs (differential voltage V_{diff} recorded from two distinct ECUs).

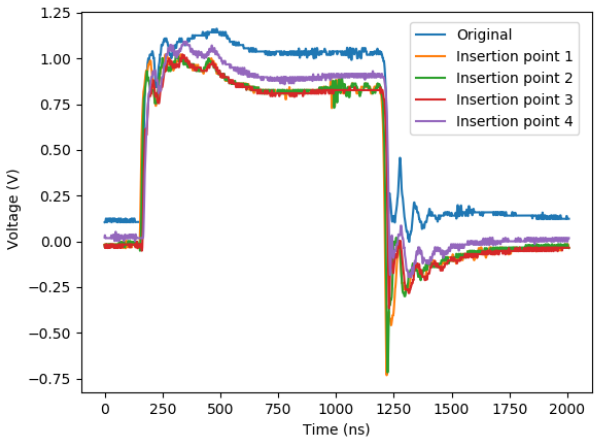


Fig. 5: A demonstration of how existing ECU signals are influenced when an ECU is added at different locations on the CAN bus.

Each CAN bus ECU outputs a signal that has unique physical characteristics which are due to both manufacturer-specific designs and tiny imperfections in the components, e.g., the internal resistance and capacitance of the ECU's transceiver. Furthermore, each ECU added to the CAN bus contributes its own resistance and capacitance, modifying the overall electronic characteristics of the CAN bus and thus

TABLE I: Summary of related work.

Ref.	Attack vector			Physical intrusion detection	Method used	Features	Experimental testbed	Sampling frequency
	Compromise ECU	Add new ECU	Replace ECU					
[32]	✓	-	-	-	Signal processing	Raw signal	CAN bus prototype	2 GS/s
[33]	✓	-	-	-	Signal processing	Statistical features extracted from the raw signal	CAN bus prototype & two real cars	50 kS/s
[34]	✓	✓	-	-	ML (SVM, NN, BDT)	9 frequency domain & 8 time domain features	CAN bus prototype	2.5 GS/s
[35]	✓	-	-	-	ML (LiSVM, BDT)	9 frequency domain & 8 time domain features	CAN bus prototype	2.5 GS/s
[36]	✓	✓	-	-	ML (logistic regression)	Features extracted from rising and falling edges	CAN bus prototype & two real cars	20 MS/s
[37]	✓	✓	-	-	Signal processing	Features extracted from rising and falling edges	One real car	2 MS/s
[38]	✓	-	-	-	Statistical analysis	Temperature and voltage	CAN bus prototype	50 MS/s
[23]	✓	-	-	-	Reinforcement learning	Raw signal (sampled from a dominant (0) bit)	CAN bus prototype	N/A
[22]	✓	-	-	-	ML (neural network)	Raw signal	CAN bus simulation	250 MS/s
[39]	✓	-	-	-	ML (deep learning)	Statistical features extracted from the raw signal	CAN bus prototype	2 GS/s
CAN-LOC	✓	✓	✓	✓	ML (deep learning)	Raw signal sampled from rising and falling edges	CAN bus prototype & one real car	500 MS/s

affecting the signals of all existing ECUs. The influence of this varies, depending on the connection location and the ECUs' transceiver characteristics.

III. Related Work

In contrast to timing-based or payload-based analysis, our system relies on CAN bus voltage signals, which are difficult to fake. Therefore, as related work, we only consider studies that are based on features extracted from voltage signals.

Several methods to detect spoofing attacks based on ECUs' voltage signals have been proposed. Table I summarizes and compares prior studies based on the following criteria: attack vector, physical intrusion detection, detection methods used, features extracted, experimental testbed, and signal sampling frequencies.

The first study that presented the idea of using voltage signals for ECU fingerprinting applied simple signal processing techniques on the raw signal sampled from the CAN frame's arbitration field [32]. Another study [33] applied adaptive signal processing on statistical features extracted from the raw signal; the proposed mechanism enables modification of the fingerprints and hence allows the mechanism to adapt to possible environmental changes.

In other research presented by Choi et al. [34], the authors presented improvements related to signal processing. In this study, 17 features were extracted from the extended identifiers, and a variety of machine learning algorithms were employed in order to improve the identification accuracy obtained in prior work.

Further improvements were presented in subsequent studies [36], [41], [35] in which significantly higher accuracy was achieved. The core idea behind the study is the observation that the identification accuracy can be significantly improved by processing the samples of the rising and falling edges of the voltage signals.

Additional improvements in terms of computational and data collection resources were achieved in another study [38] in which statistical analysis was applied on either temperature or voltage variations.

In another line of research, optimization of the authentication techniques mentioned above was suggested [23]. This approach is based on reinforcement learning, which allows authentication optimization via a trial and error mechanism.

More recently, deep learning techniques have been suggested [22]. The researchers used a recurrent neural network (RNN) multiclass classifier for the authentication of ECUs on the CAN bus given a raw voltage signal. In other research, a combination of feature extraction and a deep learning-based mechanism was suggested [39]. Both methods achieved good identification accuracy, however the studies did not demonstrate the proposed method's robustness to environmental changes.

In a recent study [24], the researchers presented a novel technique to evade VBS-D mechanisms. In their work, they exploit the mechanisms' retraining process by connecting a malicious ECU, specifically designed for poisoning them while adapted to environmental changes. To defend against poisoning, the researchers in [24] propose that all ECUs communicate using MACs to collect the voltage signals of ECUs during VBS-D retraining. However, as explained earlier, it is difficult to embed cryptographic elements in CAN messages for a sufficiently high security level, resulting in the need for complement techniques.

In order to address the limitations of the prior work mentioned above, in this study we first propose a mechanism which utilizes physical side-channels to identify and locate both passive and active physical intruders on the CAN bus network when the vehicle starts. Based on the **legitimate** ECUs' voltage signals transferred on the CAN bus, our proposed physical intrusion detection and localization mechanism determines whether the CAN bus has been physically compromised. To ensure safe operation of the vehicle, this process is executed when the vehicle is started to immediately issue an alert before the car is moving. We take advantage of the fact that each CAN bus topology change influences all of the voltage signals transferred on the CAN bus. Thus, our proposed mechanism is effective against silent intruders.

For evaluation, we used a CAN bus prototype identical

to that of [13], which used timing analysis to detect and locate malicious devices connected to the CAN bus. Despite this similarity, our study differs in the following ways. First, in [13], only the difference in the arrival time was used (extracted by setting a threshold for the voltage level); the shape of the signal on the CAN bus was not considered. Second, their method requires a connection to each end of the CAN bus, whereas our method only requires one connection to the CAN bus, which simplifies the wiring harness. Third, their method was unable to localize the physical intruder in cases in which a new ECU was inserted into the CAN bus, i.e., a change in the voltage characteristics of the CAN bus. By using deep learning, we can localize malicious ECUs, even when they are unknown to the mechanism and/or silent.

Then, after the vehicle has started and no intrusion device has been detected, the ECUs' voltage signals transferred on the CAN bus are utilized to detect spoofing attempts; we propose a robust continuous ECU authentication and identification mechanism that adapts to possible environmental changes that occur when the vehicle is moving. For evaluation, we use voltage signals collected from both a CAN bus prototype and a real vehicle.

IV. Network and Threat Models

A. Network Model

While in-vehicle networks may have more than a hundred ECUs, they are always grouped together in sub-networks of less than a dozen ECUs. Typically, the sub-network topology is bus oriented. In this topology, a two-wire cable connects multiple ECUs that implement various car functionalities, as illustrated in Figure 1. To protect the entire vehicle, our proposed security hardening system must be connected to each sub-network in order to sample signals from each of the existing CAN buses.

B. Threat Model

In this study, we focus on spoofing scenarios, where attackers can typically utilize the CAN bus to attack a vehicle or take full control of the ECUs by maliciously injecting forged or modified frames into the CAN bus; we consider various ways to evade detection done by the state of the art spoofing detection techniques (i.e., voltage-based techniques).

We consider two types of attackers: (1) an attacker with remote access to the CAN bus, and (2) an attacker with physical access to the CAN bus. The attacker with remote access aims to compromise the software level of an existing device and conduct spoofing attacks. The attacker with physical access aims to replace an existing ECU with a malicious ECU or insert an additional ECU in an available location. This physical intrusion is done for the purpose of poisoning VBS-D mechanisms while they are adapted to environmental changes.

An illustration of the attack surfaces is presented in Figure 6. These include open entry points to the CAN bus (e.g., the OBD port), existing critical ECUs that can be physically replaced (e.g., steering systems) or existing ECUs that can be remotely compromised (e.g., infotainment systems).

We assume that an attacker has the knowledge required to connect both sniffing and active tools to the CAN bus. We also assume that an attacker is aware of the presence of the security hardening system and how it works. However, physical signals that originate from existing ECUs cannot be cloned due to intrinsic physical properties of the transceivers on each ECU and of the transmission line that connects them. This assumption is well known and stays at the foundation of many recent works that use voltage levels in order to determine the source of messages on the CAN bus [24], [24], [35], [36], [37]. Needless to say, the physical signals that originate from each ECU are unique and cannot be reproduced by basic laboratory instrumentation, e.g., signal generators. Moreover, connecting a new device to the bus will immediately lead to changes in the impedance of the line which can be immediately detected by inspecting minute changes in the voltage levels from the bus. Indeed, the protection mechanism that we design allows for detecting topology changes and thus it is resilient to such adversarial interventions.

Similar to the way in which software systems are secured, we consider a CAN bus physically divided into trusted areas and untrusted areas. The untrusted areas are cheap, easily accessible, and spread over the entire network cable. The trusted area is small and located in a secure place. Since our defined attacker has physical access to the CAN bus, our proposed security mechanism is assumed to reside on a trusted area of the CAN bus, where malicious access is physically hardened.

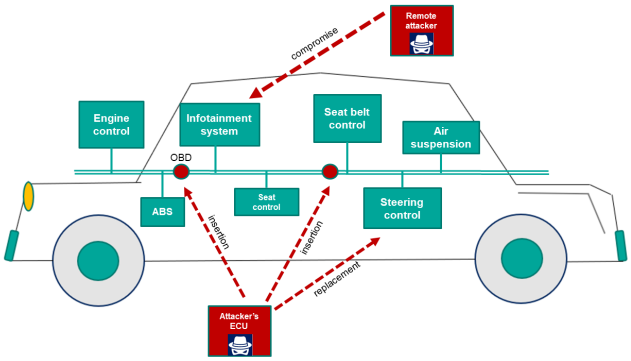


Fig. 6: Surfaces that can be used to conduct attacks on the CAN bus.

V. High-Level Description of the System

In order to secure the CAN bus from physical intruders and spoofing attacks, we propose a security hardening system which is based on continuous monitoring of the voltage signals transferred on the CAN bus.
The proposed system (illustrated in Figure 7) consists of two mechanisms:

1) Physical Intrusion Detection and Localization - this mechanism is activated when the vehicle is started. It detects whether the CAN bus has been physically compromised by a malicious intruder, and computes the location of the intrusion. A new ECU can be introduced

by inserting a new ECU into an available location on the CAN bus or by replacing an existing ECU.

2) Continuous ECU Authentication and Identification - this mechanism runs continuously after the vehicle has been started. It detects spoofing attempts and identifies the real origin of the spoofed frame.

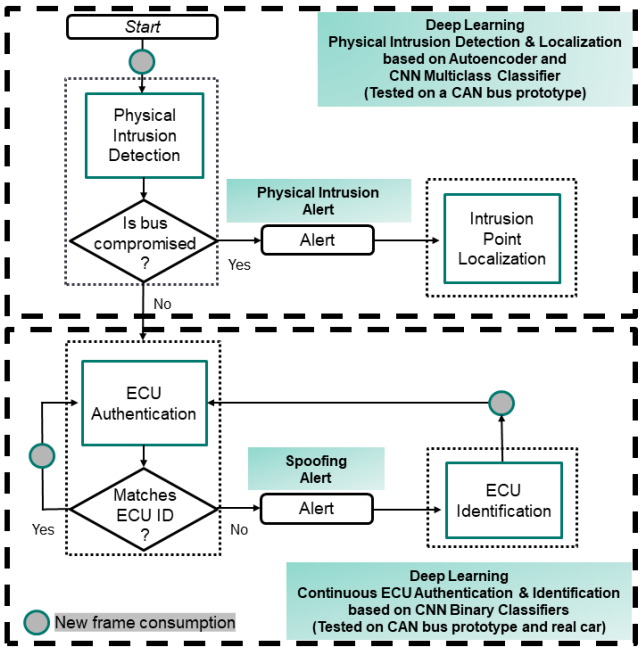


Fig. 7: High-level architecture of the proposed system.

In this work we require all known ECUs to communicate using MACs for a few seconds when the vehicle is started (i.e., during the physical intrusion detection phase). During this phase, training set is collected for generating the fingerprints of the known ECUs. If no intrusion device is detected, then our system moves to the second phase in which all the frames transferred on the CAN bus, are authenticated.

The advantages of the proposed system are twofold. First, for physical intrusion detection and localization, our system does not depend on voltage signals transferred by the intruder device. Our system only relies on known ECUs' signals, and is capable of detecting physical intrusions immediately when the vehicle starts. We show that a single voltage signal per each known ECU is sufficient to detect and locate the malicious intruder with high accuracy. Second, to detect spoofing attacks, the system is based on the analysis of voltage signals transferred on the CAN bus, which is considered as a state-of-the-art technique that well complements cryptographic-based techniques. We show that our proposed system is robust both to environmental changes and poisoning conducted by known ECUs, without requiring cryptographic operations during re-training.

A. Data Acquisition

When data is acquired from the CAN bus, CAN-LOC samples the voltage signal of CAN frames. Each CAN frame can be associated with a particular sender based on its ID field. While the CAN bus is a broadcast bus, in existing practical implementations each ECU is associated with a set of IDs that it uses to send data on the CAN bus. Remote frames (which request specific data) with the same ID as data frames can be sent by distinct ECUs, but since this type of frame does not carry any data, it cannot be the source of an impersonation attack and is not relevant to our analysis. Remote frames are easily distinguished by the RTR bit, which is set at one.

One of our goals is to authenticate each of the legitimate ECUs based on the sampled signals. Therefore, when generating the fingerprints, we need to associate each sampled signal with its ECU. Since other ECUs are allowed to transmit information in the arbitration and acknowledgement fields, the only fields that can be sampled for ECU identification are the control, data, and CRC fields (gray fields in Figure 3). As shown in previous studies [36], rising or falling edges should be sampled in order to increase the detection accuracy.

B. Proposed System Description

Physical Intrusion Detection and Localization. This mechanism determines whether the CAN bus is *clean* (no ECU was replaced or added to the CAN bus) or *dirty* (a new ECU was added or replaced an existing ECU), i.e., the CAN bus is compromised. In this study we show that a single voltage signal is sufficient to detect that the CAN bus is physically compromised. If the CAN bus is compromised, an alert is generated, and the intrusion point location is returned. As illustrated in Figure 7, two modules are proposed: (i) the physical intrusion detection module, and (ii) the intrusion point localization module.

Algorithm 1 describes the physical intrusion detection and localization mechanism. The input to the algorithm is the inspected signal (denoted by sig), which is a list of voltage samples collected from the CAN bus during a frame transmission. First, the physical intrusion detection module is used to detect whether the CAN bus is compromised (line 2). If the CAN bus is compromised, an alert is generated (line 3), and then the physical intrusion localization module is used to locate the physical intrusion point (line 4).

In this study, we assume that in-vehicle ECUs transmit frames periodically, although the presence of a silent ECU is technically possible. This would be uncommon, since each ECU handles several functionalities and must periodically report data from various sensors/actuators. Thus, the physical intrusion localization module is based on a process of monitoring legitimate ECUs' signals (line 6) which is performed to distinguish between insertion and replacement scenarios:

- All known ECUs are identified within a given time period (line 7). In this case, we conclude that it is an insertion attack, and an insertion localization procedure is executed (line 8) to return the insertion location (line 9).
- If the time period has ended, and there is a known ECU that has not been identified (line 10), we conclude that it is a replacement attack. In this case, the location of the missing ECU is returned (line 11).

The replacement of a faulty ECU is not a very frequent operation, but it may indeed happen in practice. Such an

Algorithm 1 Physical Intrusion Detection & Localization

```
1: procedure DetectPhysicalIntrusion(sig)
2:     if IsBusCompromised(sig) then
3:         GenerateAlert()
4:         return LocatePhysicalIntrusion()
5: procedure LocatePhysicalIntrusion
6:     M ← Monitor.getMissingECUs()
7:     if M = ∅ then
8:         S ← Monitor.getMonitoredSignals()
9:         location ← LocateInsertionPoint(S)
10:    else
11:        location ← LocateReplacementPoint(M)
12:    return location
```

intervention in the proposed system can be handled by updating the models used by the Physical Intrusion Detection and Localization mechanism which can be done in authorized locations since it requires data collection and retraining the models as described later in this paper. We believe that it is reasonable to assume that this operation is done in a secure environment, since in practice ECU replacements can be done only in authorized shops.

Continuous ECU Authentication and Identification. This mechanism is responsible for the continuous detection of ECU spoofing attempts. We call this mechanism Continuous ECU Authentication and Identification since the source of each frame is continuously checked for each new frame that arrives on the CAN bus. This stands in contrast to the Physical Intrusion Detection and Localization which runs only when the car starts to check that there are no topology changes. When a spoofing attempt is detected, an alert is generated, and the real origin of the spoofed frame is returned. As illustrated in Figure 7, two modules are proposed: (i) the ECU authentication module, and (ii) the ECU identification module.

The input to the ECU authentication module is the inspected signal, which is a list of voltage samples collected from the CAN bus during a frame transmission, and the identifier of the ECU transmitting it. If there is no match, an alert is generated, and then the identification module is used to return the real origin of the spoofed frame. Since the voltage fingerprint of an ECU fluctuates over time due to environmental factors [24], there is a need to frequently update the fingerprints.

Physical fingerprinting can be also done for any other sensors or actuators that are present in the car. However, in general, these components are not directly linked to the CAN bus, they are connected via a dedicated port, e.g., GPIO (General Purpose Input/Output), ADC (Analog-to-Digital Converter) or PWM (Pulse-width modulation) circuitry, to a single ECU, which makes these devices harder to access from outside. In contrast, the CAN bus is a broadcast line where adversaries can gain access and impersonate connected ECUs simply by tapping the two CAN wires, which may stretch for several meters inside the car, and are also easy to access from convenient ports, e.g., the mandatory OBD (On-Board Diagnostics) interface. Therefore, we are strictly concerned with ECU fingerprinting in this work.

In the next section, we describe our proposed fingerprints update method; because we assume that the CAN bus is *clean* (verified earlier by the physical intrusion detection module), we do not require the ECUs to communicate using MACs during retraining. Further details are provided in the next section.

VI. Low-Level Description of the System

A. Physical Intrusion Detection and Localization

The physical intrusion detection and localization mechanism consists of two modules: detection and localization. This mechanism is activated when the vehicle is started and determines whether the CAN bus is *clean* (no ECU was replaced or added to the CAN bus) or *dirty* (a new ECU was added or replaced an existing ECU), i.e., the CAN bus is compromised. If a compromised CAN bus is detected, the intrusion point is returned. The input to the mechanism should only consist of legitimate (MAC-based authenticated) signals.

The Physical Intrusion Detection module. This module is based on an autoencoder, which is an unsupervised learning algorithm that comprises an encoder and a decoder. The encoder first recreates the input data in a lower dimensionality, and then the decoder reconstructs the data back to its original dimensionality. In this manner, the normal instances are reconstructed properly, and the outliers are not. This allows the identification of anomalous input data.

We define the encoder so that it has two hidden layers set at decreased sizes of 50 percent and 25 percent of the input layer's dimension. For fast and robust training, we use batch normalization and leaky ReLU activation. The decoder has a similar structure, although in reverse.

As described in Section II-B, the basis for this module is the electric property of CAN bus topologies, in which each network topology change influences all of the signals transferred on the CAN bus. Since any new ECU connected to the CAN bus affects the voltage signals of all of the ECUs, a single CAN frame (regardless of the sender) is sufficient for detecting whether the CAN bus topology has changed.

During the training phase of the autoencoder, we use two separate chronological datasets that only contain benign data (i.e., signals transferred on the CAN bus when the network is *clean*), from which the autoencoder learns the patterns of the original CAN bus topology.

The first dataset is the training set (TR^{clean}), and the second dataset is the validation set (VAL^{clean}). Given TR^{clean}, we train the autoencoder until the *mean absolute error* (MAE) reaches its minimum on VAL^{clean}. We use the root mean square propagation (RMSProp) optimizer and a learning rate of 10^{-4} with a rate decay of 0.2.

During the inference, given an authenticated signal transferred on the CAN bus, we execute the autoencoder and measure the reconstruction error of the signal. If the reconstruction error exceeds a predefined threshold (thr), an alert is generated, and the intrusion point localization module is used to locate the intrusion point on the CAN bus. The method used to calculate thr is described later in this section.

The Intrusion Point Localization module. This module is responsible for physically locating the intrusion point on the CAN bus when the latter is detected to be compromised (i.e., *dirty*). First, we need to eliminate a case in which the CAN bus is *dirty* due to the replacement of a legitimate ECU. We

identify the replacement of a legitimate ECU by monitoring the CAN bus for a certain period of time TP, in order to determine whether all of the ECUs are present.

Given the cyclical nature of in-vehicle traffic in which there are predefined cycles (usually in the range of 10-100 ms) and each ECU is in charge of multiple such frames, a few dozen milliseconds, on average, should be sufficient to verify whether all the legitimate ECUs are present.

As illustrated in Figure 8, when the CAN bus found to be *dirty* by the physical intrusion detection module, the monitoring process collects signals authenticated during time period TP (one per ECU). If all of the ECUs are successfully authenticated during time period TP, the insertion point localization procedure is used to locate the intruder. Otherwise, the location of the ECU that wasn't authenticated is returned.

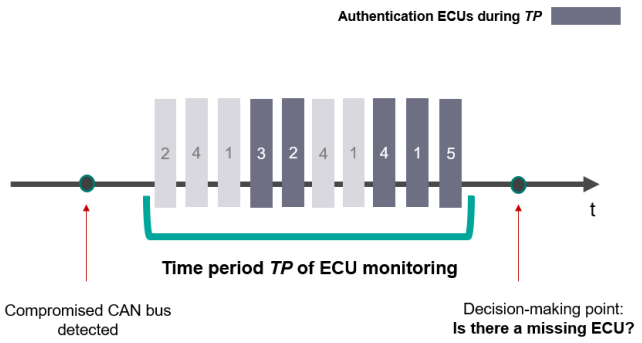


Fig. 8: Authenticated signal monitoring process on a CAN bus containing five legitimate ECUs.

The insertion point localization procedure is based on a CNN multiclass classifier. The proposed architecture is a one-dimensional variant of VGG16 [42] in which a softmax output layer is attached, providing a probability distribution over the predicted output classes. VGG16 is a neural network-based classification model whose architecture was originally designed for image classification; the default input size is 224x224. In our case, we changed the input layer to a one-dimensional layer whose input size is $1x|s|$ for s representing a voltage signal containing $|s|$ samples.

Let $P = \{p_1, p_2, ..., p_n\}$ be a set of insertion points on the CAN bus. These points are represented by the classes of the classifier's output layer. During the training set collection, the transmitted signals are collected for a predefined time period at each point $p \in \{p_1, p_2, ..., p_n\}$ where a new silent ECU should be inserted. The transmitted signals collected in each time period are labeled with insertion point p. Only signals that are associated with legitimate ECUs are considered.

To increase classification accuracy, we employ a data augmentation technique. Data augmentation is the creation of data from original data, typically by applying a transformation to the original data. Data augmentation is commonly used to improve the versatility of machine learning models, as well as to provide more training examples for datasets of a limited size. In signal data, for example, it is common to use data augmentation techniques like Gaussian noise addition, cyclic rolling-off (shifting), clipping distortion, and frequency masking [43], [44]. In this study, to ensure robustness of the proposed classifier to environmental changes[45], we extend the collected training set by using the following data augmentation techniques: (1) Gaussian noise addition, and (2) random cyclic rolling-off (shifting). When the data set is too small, data augmentation techniques can be used to improve the classification accuracy [43], [44].

Algorithm 2 Generate Augmented Signals

```
1:  procedure GenerateSignals(S^i, P, K, R)
2:      AS^i ← ∅
3:      μ ← 0
4:      σ ← 1
5:      for p ∈ P do
6:          AS^i_p ← ∅
7:          for each s ∈ S^i_p do
8:              for c ∈ C(K, s) do
9:                  n ← RandomizeGaussian(μ, σ)
10:                 c ← c + n
11:                 r ← RandomizeUniform([0, R])
12:                 AS^i_p ← RollOff(c, r)
13:         AS^i ← AS^i_p
14:     return AS^i
```

The proposed data augmentation process is described in Algorithm 2. The input to the algorithm consists of the collected signals associated with ECU i (denoted by S^i). Other input to the algorithm consists of a set of discrete insertion points (denoted by P) and two integers K and R. For each insertion point $p \in P$ (line 5) for each signal $s \in S^i_p$ (line 7), we generate K copies of the signal s (line 8). To each copy (line 9), we first add Gaussian noise that is distributed with mean $\mu = 0$ and standard deviation $\sigma = 1$ (lines 10-11), and then we roll off (shift) a random amount of steps (line 12). Finally, we assign class p to each signal generated in this loop (line 13).

During the training phase of the classifier, we use the RMSProp optimizer and a learning rate of 10^{-5} with a rate decay of 0.9, and *categorical cross-entropy* is used as the loss function. First, we chronologically extract 30% of the training set to serve as the validation set. Then, we train the classifier until the loss function reaches its minimum on the validation set.

During the inference (i.e., insertion point localization), the signals collected by the monitoring process described earlier are used by the insertion point localization procedure for locating the insertion point. The insertion location estimation technique is presented in Algorithm 3.

Algorithm 3 Locate Insertion Point

```
1:  procedure LocateInsertionPoint(S)
2:      P ← Classifier(S)
3:      C ← ∅
4:      for P_i ∈ P do
5:          C.add(argmax(P_i))
6:      L ← majority(C)
7:      if size(L) = 1 then
8:          location ← L
9:      else
10:         location ← RandomizeElement(L)
11:     return location
```

Given m which represents the number of legitimate ECUs that are connected to the CAN bus, let $s = \{s_1, s_2, ...s_m\}$ be a set of signals represented as one-dimensional vectors (one per ECU). Let P be a matrix such that the column P_i is the classifier prediction given the input s_i. P_i represents the probability distribution over the classes (insertion locations on the CAN bus). The input to the algorithm 3 is a group of m signals where signal i is associated with legitimate ECU i (the group is denoted by S). First, we call the classifier and obtain $|P| = m$ predictions (line 2). Then, we take the most probable class from each column P_i as a class candidate (lines 4-5) and apply a majority over the candidates (line 6). If one candidate remains (line 7), it is returned (line 8). Otherwise, a randomized candidate is returned (line 10).

Calculating thr. The threshold (thr) is determined to discriminate between benign (i.e., voltage signals transferred on the CAN bus when the CAN bus is *clean*) and anomalous signals (i.e., voltage signals transferred on the CAN bus when the CAN bus is *dirty*). To determine a robust threshold thr, we need to consider that environmental conditions (e.g., temperature) influence the voltage signals. However, as we show later in this paper, insertion/replacement of another ECU to the CAN bus influences the voltage signals significantly more than the temperature. This is due to the fact that insertion/replacement is an intrusive operation. Using our trained autoencoder, we show that the signal reconstruction on a *dirty* CAN bus is significantly worse than the signal reconstruction on a *clean* CAN bus, even when the latter is computed under a wide range of temperature conditions. Let Val^{dirty} be the validation set representing the voltage signals collected from a *dirty* network (i.e., while the network is physically compromised at different points), a robust thr is calculated as:

$$thr = min(MAE_{Val^{dirty}}) \tag{1}$$

The replacement location estimation technique is presented in Algorithm 4. Given the group of missing, known ECUs detected by Algorithm 1, the set of their original physical locations on the CAN bus is returned. These locations are assumed to be pre-known by our localization mechanism.

Algorithm 4 Locate Replacement Point

```
1: procedure LocateReplacementPoint(M)
2:     L ← ∅
3:     for M_i ∈ M do
4:         L.add(location(M_i))
5:     return L
```

B. Continuous ECU Authentication and Identification

The ECU Authentication module. For each legitimate ECU *i*, a binary classifier is built based on a CNN responsible for authenticating the ECU. We define each binary classifier so it includes two convolutional layers followed by a max pooling layer to reduce the size. Each convolution layer has 32 filters. In addition, one fully connected layer is attached, which contains 100 neurons. All layers use the rectified linear unit (ReLU) as an activation function except for the output layer. The output layer consists of a single neuron which uses a Sigmoid as an activation function; this layer is aimed at producing the probability that a given example is associated with ECU *i*.

The training set used to train the binary classifiers consists of the voltage signals transferred on the CAN bus and associated with the legitimate ECUs. To train the binary classifier B_i for ECU *i*, each signal is classified according to the associated frame's origin ('1' if the origin of the signal is ECU *i* and '0' otherwise). To address a possible data unbalance, we use the cost-sensitive learning method described in [46]. The idea behind this method is that the training procedure is modified so that some examples have more or less errors than others. In addition, to avoid overfitting, we define two dropouts set at 0.5; one is for the max pooling layer, and the other is for the fully connected layer.

The initial training set is collected when the vehicle is started. First, we chronologically extract 30% of the initial training set to serve as the validation set. Then, to generate B_i, we use the RMSProp optimizer and a learning rate of 10^{-4} with a rate decay of 0.9, and *binary cross-entropy* is used as the loss function. Finally, when the loss function reaches its minimum on the validation set, the hyperparameters for each binary classifier are locally stored.

During the inference (i.e., ECU authentication), given a signal associated with a CAN frame, we extract its ID and apply the appropriate binary classifier to the signal. The output returned from the classifier is the probability that the given signal matches the CAN frame ID. If the network output is less than 0.5, an alert is generated, and the ECU identification module is used to return the real origin of the spoofed frame.

In this work, we assume that environmental changes occur progressively, and accordingly, we use each authenticated signal to retrain the binary classifiers. Each classifier is retrained given the most recently stored hyperparameters (i.e., CNN's weights, learning rate, and rate decay). A single epoch is performed per each authenticated signal.

The ECU Identification module. Given a signal to identify, we call each of the binary classifiers and return the appropriate identifier according to highest value returned.

VII. Experiments and Results

In this section we describe the experiments performed to evaluate the proposed system and present the results.

A. Evaluation Setup

CAN Bus Prototype. As shown in Figure 9, our experimental setup is identical to the setup used in prior research [13]. In this section we show that significantly better results are achieved when using the proposed physical intrusion detection and localization mechanism while only utilizing physical side-channels. As illustrated in Figure 10, there are 10 connection points on the CAN bus. Some of them (green) are for legitimate ECUs, and the others (white) are left open for malicious ECUs to be connected to the CAN bus.

We consider a number of networks with different configurations to evaluate the CAN-LOC system:

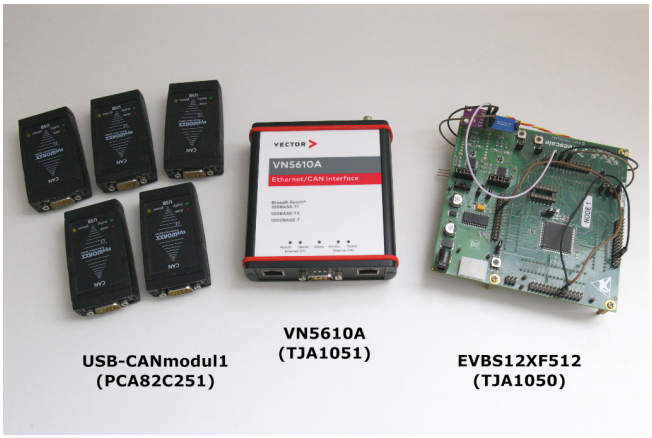


Fig. 9: CAN bus prototype.

- *Network 0* - a *clean* network in which all of the legitimate ECUs (and only those ECUs) are connected and transfer CAN frames, as depicted in Figure 10.
- *Networks 1-3* - *dirty* networks in which a malicious ECU replaces a legitimate ECU at one of the locations depicted by the red circles in Figure 11 (i).
- *Networks 4-8* - *dirty* networks in which a malicious ECU is inserted into the CAN bus at one of the locations depicted by the red circles in Figure 11 (ii).

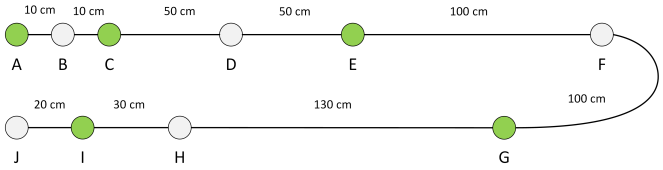


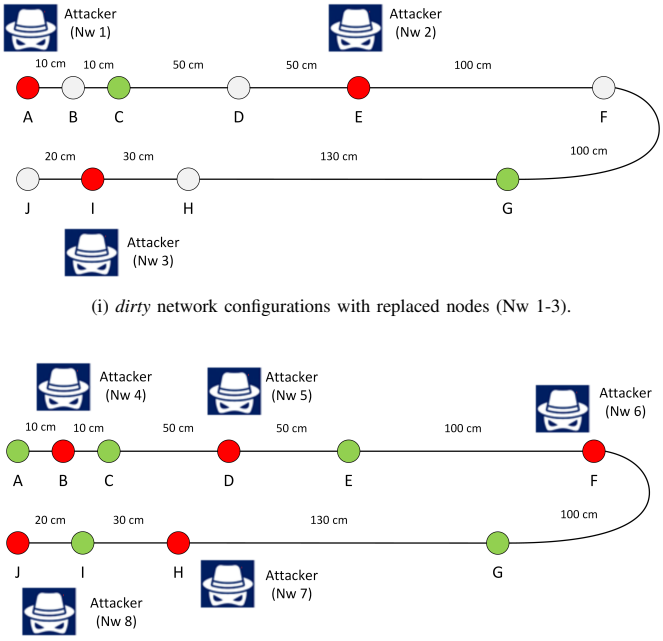
Fig. 10: Location of legitimate ECUs (green) and available points (white) for intruders on our CAN bus prototype.

As depicted in Figure 9, we use PC-to-CAN adapters (USB-CANmodul1 and VN5610A) and the EVBS12XF512 automotive grade development board, equipped with an external transceiver (TJA1050), in our setup. Table II lists the abbreviated notation used, device type, transceiver type, and amount of each ECU, as well as its role in our experiment. L_i is the legitimate ECU i $(1 \leq i \leq 5)$, A_1 is the malicious ECU used for training, and A_2 (a completely different ECU related to A_1) is the malicious ECU used for testing. The network configurations, along with their designations, are listed in Table III.

TABLE II: ECU devices and transceiver types and their role in the experiments.

Abbrev.	Device	Transceiver	Amount	Role
L_i	USB-CANmodul1	PCA82C251	5	legitimate
A_1	VN5610A	TJA1051	1	adversary
A_2	EVBS12XF512	TJA1050	1	adversary

Real Vehicle. A 2015 Honda Civic (Figure 12) was used to evaluate the proposed continuous ECU authentication and



(i) *dirty* network configurations with replaced nodes (Nw 1-3).

(ii) *dirty* network configurations with inserted nodes (Nw 4-8).

Fig. 11: The adversarial network configurations examined.

TABLE III: Experimental network configurations.

Nw. Conf.	Connection point									
	A	B	C	D	E	F	G	H	I	J
Nw0	L_1		L_2		L_3		L_4		L_5	
Nw1	$A_{1,2}$		L_2		L_3		L_4		L_5	
Nw2	L_1		L_2		$A_{1,2}$		L_4		L_5	
Nw3	L_1		L_2		L_3		L_4		$A_{1,2}$	
Nw4	L_1	$A_{1,2}$	L_2		L_3		L_4		L_5	
Nw5	L_1		L_2	$A_{1,2}$	L_3		L_4		L_5	
Nw6	L_1		L_2		L_3	$A_{1,2}$	L_4		L_5	
Nw7	L_1		L_2		L_3		L_4	$A_{1,2}$	L_5	
Nw8	L_1		L_2		L_3		L_4		L_5	$A_{1,2}$

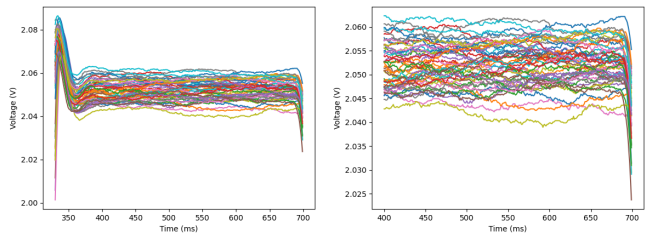
Fig. 12: 2015 Honda Civic.

identification mechanism. Through the OBD-II port, the voltage signals were sampled from the in-vehicle CAN bus containing six ECUs, running at a transmission bitrate of 500Kbps which is the usual bitrate for passenger cars.

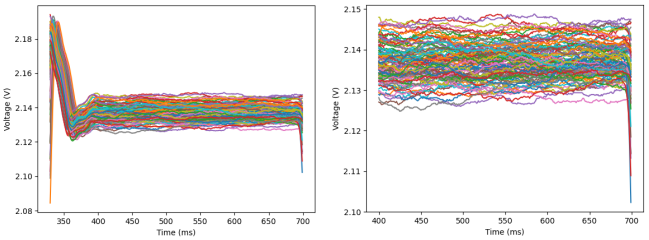
In all the experiments described in this study, we sampled

individual bits that are synchronized based on recessive to dominant transitions (one to zero) which, in the worst case, occur after 5 recessive bits. Note that the CAN protocol requires one bit of opposite polarity to occur after 5 bits of the same value (stuffing rule) in order to avoid desynchronization. The digital representation of all voltage signals are normalized to the range of 0 to 1, because this significantly impacts the classification accuracy.

In Figure 13 we present two examples of the data used for ECU identification, for two different ECUs. As can be seen, the data exhibits realistic noise and voltage fluctuations, which are expected in a real car but will not affect the accuracy of the identification as we later show.



(i) ECU 1.



(ii) ECU 2.

Fig. 13: A demonstration of a recessive "1" to a dominant "0" transition and return for 100 voltage signals (differential voltage V_{diff} recorded from two different ECUs).

B. Results

1) Evaluating the physical intrusion detection: As described in the previous section, this module focuses on detecting whether the CAN bus is physically compromised or not. The CAN bus prototype (Figure 9) is used to evaluate this module. We only sample CAN-H values, since we observed that sampling a single wire is sufficient to detect and locate physical intrusions. We observe in preliminary experiments that sampling CAN-L values provides similar results.

Training set collection. 300 signals for each ECU are collected to train the autoencoder, all of which are collected from network 0, where only legitimate ECUs are attached. Additional 300 signals for each ECU are collected for each network from the set of networks 1-8 to determine the threshold thr, associated only with the legitimate ECUs. The malicious ECU used for insertion and replacement is A_1. All the signals collected to train the autoencoder (i.e., the signals collected from network 0) are collected at a temperature of 24°C.

Test set collection. 700 signals for each ECU are collected to test the autoencoder, all of which are collected from network 0; the expected prediction for each signal is *clean*. Additional 700 signals for each ECU are collected for each network from the set of networks 1-8, all of which are associated with legitimate ECUs, and the expected prediction for each of those signals is *dirty*. The malicious ECU used for insertion and replacement is A_2. The signals collected from network 0 to test the autoencoder are collected at temperatures of 0°C, 24°C, 50°C, and 60°C.

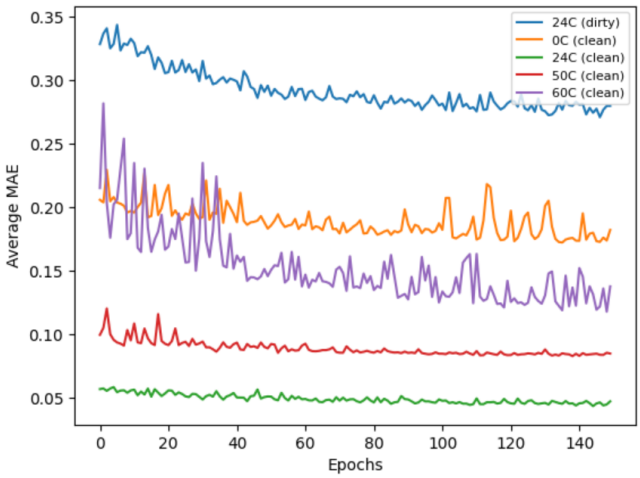


Fig. 14: The average MAE of *clean* and *dirty* signals as a function of the number of epochs used to train the autoencoder.

Detection evaluation. Figure 14 presents the average MAE of *clean* and *dirty* signals a function of the number of epochs used to train the autoencoder. As can be seen, there is a large margin between the average reconstruction errors of *clean* and *dirty* signals. Our evaluation results show high accuracy (+0.99 precision and 1 recall) in identifying *dirty* scenarios given the entire test set. These results reflect the ability of the proposed module to detect intruders using solely the signals of legitimate ECUs, in a wide range of temperatures. The sampling frequency used in this experiment was 250 MS/s.

2) Evaluating the intrusion point localization: As described in the previous section, this module is responsible for physically locating the intruder on the CAN bus when it has been compromised. The CAN bus prototype (Figure 9) was used to evaluate this module.

Since replacement point localization relies on the ability to identify all of the legitimate ECUs, its performance is derived directly from the accuracy of the MACs.

Training set collection. 300 signals for each ECU are collected for each network from the set of networks 4-8 and assigned respectively with points B, D, F, H, and J (see Figure 11). Those signals are provided to the data augmentation algorithm (Algorithm 2, denoted by S^i) which generates additional signal examples for training. To generate the entire dataset for training, Algorithm 2 is executed five times, against five legitimate ECUs that the CAN bus prototype contains.

TABLE IV: Authentication experiment results evaluated on the CAN bus prototype.

ECU1		ECU2		ECU3		ECU4		ECU5	
FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR
0.001	0	0	0	0	0	0	0	0.001	0

On a call i to Algorithm 2, we provide the collected signals associated with ECU i as input (denoted by S^i), the set of insertion points P=B, D, F, H, J, a parameter K set at 100, and a parameter R set at 10. For all the five ECUs, the resulting signals are used to train the classifier. The malicious ECU used for insertion and replacement is A_1.

Test set collection. 700 signals for each ECU are collected for each network from the set of networks 4-8. The malicious ECUs used for insertion and replacement are both A_1 and A_2.

Localization evaluation. The localization evaluation is performed by providing the classifier with five signals as input (one per legitimate ECU). During our evaluation, perfect results are achieved: we report 100% success rate in the ability of the proposed module to localize inserted intruders using solely the signals of legitimate ECUs. The sampling frequency used in this experiment was 500 MS/s. In Figure 15 we present the localization accuracy as a function of the sampling frequency; as can be seen, sampling frequency of 250 MS/s reduced the localization accuracy by more than 10%.

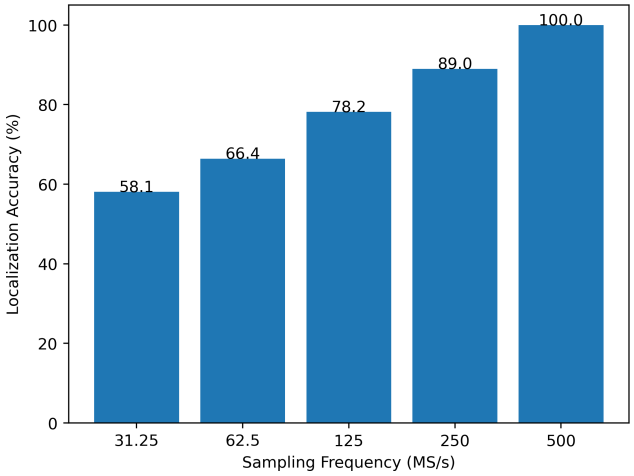


Fig. 15: Localization accuracy as a function of the sampling frequency.

3) Evaluating the continuous ECU authentication: Both the CAN bus prototype and a real vehicle are used to evaluate this module. We observed that using the differential between the CAN-H and CAN-L values contributes to the robustness of the proposed ECU authentication module. Each binary classifier's performance is evaluated in terms of the false rejection rate (FRR) and false acceptance rate (FAR).

Evaluation on the CAN bus prototype. We train the binary classifiers using 300 signals for each ECU that are collected from each network from the set networks 0-8. Then, we evaluate them using another 700 signals for each ECU that are collected from each network at the set of networks 0-8. The malicious ECU used for insertion and replacement is A_1. As presented in Table IV, good results were achieved. Identical results were achieved also when repeating the experiment using A_2 for insertion and replacement.

TABLE V: Authentication experiment results evaluated on a real vehicle.

ECU1		ECU2		ECU3		ECU4		ECU5		ECU6	
FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR
0 minutes											
0	0	0	0.001	0.002	0	0	0	0	0	0.002	0
15 minutes											
0.008	0	0	0.004	0.006	0.003	0	0	0	0	0.006	0
30 minutes											
0	0	0	0	0	0	0	0	0	0	0.01	0
60 minutes											
0	0	0	0.004	0	0.003	0	0	0	0	0.003	0

Evaluation on a real vehicle. We evaluate our proposed authentication method on a real vehicle when moving. To generate the initial binary classifiers, 700 signals for each ECU are collected when the vehicle is started. All of the signals are collected when the vehicle is stationary, and their true labels are used for training.

The signals used in our evaluation are collected when the vehicle is moving. They are grouped into four separate datasets according to the length of time the car was running: (1) 0 minutes, (2) 15 minutes, (3) 30 minutes, and (4) 60 minutes. Each dataset contains 4,000 signals for each ECU.

Table V presents the performance of the proposed authentication module. Evaluation of dataset 1 was performed using the binary classifiers generated when the vehicle was started. Then, Dataset 1 is used to update the fingerprints. For each j greater than 1, we evaluate dataset j using the binary classifiers that were updated given dataset j-1. The labels used during each update are determined according to the output value obtained by the binary classifiers themselves.

In the table, we can see that low FRR and FAR values were achieved on each dataset. The overall identification accuracy was 99.8%. The sampling frequency used in this experiment was 250 MS/s. Assuming that environmental changes occur progressively, we conclude that the results achieved demonstrate the robustness of our method to environmental changes.

VIII. System Deployment

Similar to the mechanism proposed in [31], our system can be implemented on an external node attached to the CAN bus, and the ML models should be stored in a physically secure area. Such deployment addresses the fact that there is already a large number of vehicles on the road. The dataset required to induce the intrusion detection and localization models of the system for vehicles on the road be collected at the garage. The ML models should be re-trained each time an ECU is added or replaced, since the ECUs' electrical characteristics differ. When an ECU's software is updated, re-training the model is not required, since no physical mechanism changes, but only the logic. For new vehicles, the dataset required to induce these models can be collected after the vehicle has been produced, i.e., during the vehicle testing phase on the production line. To

achieve the accurate detection demonstrated in Section VII, a DSP with a sampling rate of 500 MS/s should be used in the deployed system.

Regarding the computational power, on a 2.11 GHz Intel Core i7-8665U processor, it took about three seconds to parse 15K frames during authentication, which corresponds to a processing time in the order of $200\mu s$ per frame. This corresponds to the time required to process frames in real time, since the time spent by a frame on a 500 Kbps CAN bus is around $200\mu s$. This amount of computational power is available on a modern high-end DSP. During the authentication models' initial training, we observed that 2K signals are sufficient for training when the vehicle is started. This number of samples can usually be collected from an ECU in a matter of seconds. Given 2K signals per each ECU, each binary classifier is generated within a few seconds.

The neural networks' hyperparameters were tuned using data collected from the CAN bus prototype. All the results presented in Section 7 reflect the experiments performed on a new dataset using the pre-tuned hyperparameters.

As stated in Section I, our proposed system complements a prevention mechanism proposed in [31] that requires accurate localization of the intrusion point, which our proposed system provides by using deep learning. From a data collection perspective, the mechanism described by the authors in [31] can also be used for the automatic examination and diagnosis of specific segments of the CAN bus.

IX. Conclusion

In this study we demonstrated how CAN bus voltage signals can be used to identify and locate physical intruders on the CAN bus network. Since we do not depend on an adversary's transmission, our physical intrusion detection and localization mechanism is effective against silent intruders. Our evaluation results show high detection accuracy when simulating a large variety of physical intrusion scenarios.

By using data collected in a wide range of temperatures, we showed that the physical intrusion detection module is robust to environmental changes; we observed that connecting a new ECU to the CAN bus influences the original voltage signals significantly more than temperature changes. Using an autoencoder trained to reconstruct voltage signals transferred on a *clean* CAN bus, we showed that the signals' reconstruction for a *dirty* CAN bus is significantly worse than the reconstruction for a *clean* CAN bus (when the latter is measured under a wide range of temperature conditions). We rely on this observation as an indication of the robustness of the localization module, which is trained given augmented data, to adapt to possible temperature changes.

Regarding adversarial machine learning, although the software of an in-vehicle ECU can be remotely compromised, it is difficult to alter the voltage signals' characteristics, and hence the corresponding voltage fingerprint of the compromised ECU, in a controlled manner [24]. Therefore, the only feasible way to evade voltage-based mechanisms is by exploiting the retraining process using an additional ECU connected to the CAN bus. In our work, we address this by detecting the intruder prior to retraining as a preventative action, which complements the limited cryptography-based authentication methods.

Regarding the manipulation of bits inside frames, we point out that given the electrical properties of the CAN bus, an adversary can only force a recessive to dominant transition (dominant bits cannot be overwritten by recessive bits). Since this will immediately trigger a rising edge, and the deep neural network is specifically trained on rising edges, such manipulation will be detected; the only requirement is that the intrusion detection system monitors every rising edge in a frame. Such attacks are very difficult to perform due to synchronization issues and the fact that the legitimate node needs to be eliminated from the CAN bus, otherwise there is a strong chance of at least a CRC error, since the data field is manipulated.

In this study, we focused on the degree of sensitivity of the proposed mechanism to different physical conditions like (1) steady state, (2) short and long-time movement, and (3) temperature. This sensitivity has been shown to detect physical intrusions with high accuracy; in Section 2, we visually illustrate the difference between the signals of two ECUs, as sampled from the rising and falling edges of a CAN frame (Figure 4). In Figure 5 we visually illustrate how existing ECU signals are influenced when an ECU is added at different locations on the CAN bus.

For real-world vehicles, it is much more likely that an adversary performs a single insertion on exposed ports, like the OBD, due to obvious difficulties in exposing the in-vehicle wires that are generally hard to access. Still, even if such a scenario occurs, the attack will be detected as it will lead to impedance changes on the bus and, if the localization becomes inaccurate, the car can be sent to a specialized garage for inspection by qualified personnel (note that the presence of adversarial devices, single or multiple, will be signaled immediately after the car is started, so that the owner will be informed in time).

As stated, our experiments include realistic noise from an in-vehicle CAN bus. But it is indeed true that one cannot collect all possible kinds of noise that occur on a CAN bus. Still, the differential signaling of the bus is specifically meant to eliminate electronic noise (by computing the difference between CAN-H and CAN-L and thus removing common mode errors, rather than using the ground voltage GND as a reference). We leave further experiments with extreme noise variations as potential future work.

In future research, we also plan to evaluate the authentication and identification mechanism in additional scenarios, e.g., when an ECU goes into bus-off or low-power mode, and when the supply voltage from the ECUs' fluctuates.

References

[1] C. Specification, "Version 2.0," *Robert Bosch GmbH*, vol. 27, 1991.

[2] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 534–545, 2014.

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

[4] K. Koscher, S. Savage, F. Roesner, S. Patel, T. Kohno, A. Czeskis, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2010, pp. 447–462.

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

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

[7] A. Palanca, E. Evenchick, F. Maggi, and S. Zanero, "A stealth, selective, link-layer denial-of-service attack against automotive networks," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*. Springer, 2017, pp. 185–206.

[8] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

[9] A. Harel and A. Hezberg, "Optimizing can bus security with in-place cryptography," SAE Technical Paper, Tech. Rep., 2019.

[10] H. Schweppe, Y. Roudier, B. Weyl, L. Apvrille, and D. Scheuermann, "Car2x communication: Securing the last meter - a cost-effective approach for ensuring trust in car2x applications using in-vehicle symmetric cryptography," in *2011 IEEE Vehicular Technology Conference (VTC Fall)*, 2011, pp. 1–5.

[11] Z. Liu, J. Ma, J. Weng, F. Huang, Y. Wu, L. Wei, and Y. Li, "Lppte: A lightweight privacy-preserving trust evaluation scheme for facilitating distributed data fusion in cooperative vehicular safety applications," *Information Fusion*, vol. 73, pp. 144–156, 2021.

[12] Z. Liu, J. Weng, J. Guo, J. Ma, F. Huang, H. Sun, and Y. Cheng, "Pptm: A privacy-preserving trust management scheme for emergency message dissemination in space–air–ground-integrated vehicular networks," *IEEE Internet of Things Journal*, vol. 9, no. 8, pp. 5943–5956, 2021.

[13] P.-S. Murvay and B. Groza, "Tidal-can: Differential timing based intrusion detection and localization for controller area network," *IEEE Access*, vol. 8, pp. 68 895–68 912, 2020.

[14] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of can bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, 2017, pp. 1–4.

[15] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 911–927.

[16] S. Kulandaivel, T. Goyal, A. K. Agrawal, and V. Sekar, "Canvas: Fast and inexpensive automotive network mapping," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 389–405.

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

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

[19] G. M. Zago and E. P. de Freitas, "A quantitative performance study on can and can fd vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, pp. 4413–4422, 2017.

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

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

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

[23] T. Xu, X. Lu, L. Xiao, Y. Tang, and H. Dai, "Voltage based authentication for controller area networks with reinforcement learning," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–5.

[24] R. Bhatia, V. Kumar, K. Serag, Z. B. Celik, M. Payer, and D. Xu, "Evading voltage-based intrusion detection on automotive can," in *Network and Distributed System Security Symposium (NDSS)*, 2021.

[25] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 19–35.

[26] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," *arXiv preprint arXiv:1206.6389*, 2012.

[27] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[28] B. Timon, "Non-profiled deep learning-based side-channel attacks with sensitivity analysis," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 107–131, 2019.

[29] D. Das, A. Golder, J. Danial, S. Ghosh, A. Raychowdhury, and S. Sen, "X-deepsca: Cross-device deep learning side channel attack," in *Proceedings of the 56th Annual Design Automation Conference 2019*, 2019, pp. 1–6.

[30] F. Wegener, T. Moos, and A. Moradi, "Dl-la: Deep learning leakage assessment," *IACR Cryptology ePrint Archive*, 2019.

[31] L. P. L. Bogdan Groza and S. Murvay, "Canary - attack prevention on the can bus by load balancing with active relays," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[32] P.-S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, 2014.

[33] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1109–1123.

[34] 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.

[35] 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.

[36] 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*, 2018, pp. 787–800.

[37] M. Kneib, O. Schell, and C. Huth, "On the robustness of signal characteristic-based sender identification," *arXiv preprint arXiv:1911.09881*, 2019.

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

[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)*. IEEE, 2019, pp. 29–38.

[40] R. M. Gerdes, M. Mina, S. F. Russell, and T. E. Daniels, "Physical-layer identification of wired ethernet devices," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1339–1353, 2012.

[41] M. Kneib, O. Schell, and C. Huth, "Easi: Edge-based sender identification on resource-constrained platforms for automotive networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2020, pp. 1–16.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[43] Z. Zhang, F. Duan, J. Sole-Casals, J. Dinares-Ferran, A. Cichocki, Z. Yang, and Z. Sun, "A novel deep learning approach with data augmentation to classify motor imagery signals," *IEEE Access*, vol. 7, pp. 15 945–15 954, 2019.

[44] L. Huang, W. Pan, Y. Zhang, L. Qian, N. Gao, and Y. Wu, "Data augmentation for deep learning-based radio modulation classification," *IEEE Access*, vol. 8, pp. 1498–1506, 2019.

[45] E. Kim, J. Kim, H. Lee, and S. Kim, "Adaptive data augmentation to achieve noise robustness and overcome data deficiency for deep learning," *Applied Sciences*, vol. 11, no. 12, p. 5586, 2021.

[46] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 4368–4374.

Efrat Levy is a senior security researcher at Intel Corporation and a Ph.D. student at Ben-Gurion University of the Negev (BGU). She has been actively leading significant security innovative projects in the industry and academia for more than a decade. She holds an M.Sc. degree in the field of Quantum Computing from the Hebrew University of Jerusalem, Israel. Her primary research interests are computer and network security, machine learning, cryptography and side-channel attacks.

Asaf Shabtai is a Professor in the Department of Software and Information Systems Engineering at Ben-Gurion University of the Negev. His main areas of interest are computer and network security, machine learning, security of the IoT and smart mobile devices, and security of avionic and operational technology systems.

Bogdan Groza is Professor at Politehnica University of Timisoara (UPT). He received his Dipl.Ing. and Ph.D. degree from UPT in 2004 and 2008 respectively. In 2016 he successfully defended his habilitation thesis having as core subject the design of cryptographic security for automotive embedded devices and networks. He has been actively involved inside UPT with the development of laboratories by Continental Automotive and Vector Informatik. Besides regular participation in national and international research projects in information security, he lead the CSEAMAN (2015-2017) and PRESENCE (2018-2020) projects, two research programs dedicated to the security of vehicular ecosystems funded by the Romanian National Authority for Scientific Research and Innovation.

Pal-Stefan Murvay is a Lecturer at Politehnica University of Timisoara (UPT). He graduated his B.Sc and M.Sc studies in 2008 and 2010 respectively and received his Ph.D. degree in 2014, all from UPT. He has a 10-year background as a software developer in the automotive industry. He worked as a postdoctoral researcher in the CSEAMAN project and is currently a senior researcher in the PRESENCE project. He also leads the SEVEN project related to automotive and industrial systems security. His current research interests are in the area of automotive security.

Yuval Elovici is the director of the Telekom Innovation Laboratories at Ben-Gurion University of the Negev (BGU), head of BGU Cyber Security Research Center, Professor in the Department of Software and Information Systems Engineering at BGU. He holds B.Sc. and M.Sc. degrees in Computer and Electrical Engineering from BGU and a Ph.D. in Information Systems from Tel-Aviv University. His primary research interests are computer and network security, cyber security, web intelligence, information warfare, social network analysis, and machine learning. Prof. Elovici also consults professionally in the area of cyber security and is the co-founder of Morphisec, startup company that develop innovative cyber-security mechanisms that relate to moving target defense.