VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System

Wonsuk Choi, Kyungho Joo, Hyo Jin Jo, Moon Chan Park, and Dong Hoon Lee, *Member, IEEE*

***Abstract*— The proliferation of computerized functions aimed at enhancing drivers' safety and convenience has increased the number of vehicular attack surfaces accordingly. The fundamental vulnerability is caused by the fact that the controller area network protocol, a de facto standard for in-vehicle networks, does not support message origin authentication. Several methods to resolve this problem have been suggested. However, most of them require modification of the CAN protocol and have their own vulnerabilities. In this paper, we focus on securing in-vehicle CAN networks, proposing a novel automotive intrusion detection system (so-called VoltageIDS). The system leverages the inimitable characteristics of an electrical CAN signal as a fingerprint of the electronic control units. The noteworthy contributions are that VoltageIDS does not require any modification of the current system and has been validated on actual vehicles while driving on the road. VoltageIDS is also the first automotive intrusion detection system capable of distinguishing between errors and the bus-off attack. Our experimental results on a CAN bus prototype and on real vehicles show that VoltageIDS detects intrusions in the in-vehicle CAN network. Moreover, we evaluate VoltageIDS while a vehicle is moving.**

***Index Terms*— Controller area network, electronic control unit, automotive IDS, fingerprinting.**

I. Introduction

VEHICLE functions are becoming increasingly computerized to improve drivers' safety and increase their convenience. Sensors enable the vehicle to detect driving conditions. This enables the vehicle to automatically control certain functions, such as motor-driven power steering (MDPS). Recently, the advanced driver assistance system (ADAS) has emerged as a prime example of an early form of automated vehicle technology. Several electronic control units (ECUs) are used to enable computer-based control of a vehicle. Approximately 50-100 ECUs can be found in some luxury vehicles [1], [2]. ECUs are each assigned a specific function (e.g., engine control) and are grouped into several subnetworks according to their functions. For example, the ECUs in charge of steering and braking are grouped into one in-vehicle subnetwork, where information regarding their functions is shared. The networked ECUs communicate through the controller area network (CAN) protocol, which provides one of the most reliable communication systems. In the US, all recently sold vehicles are required to implement the CAN protocol as one of the on-board diagnostics (OBD-II) signal protocols. Despite the high reliability and widespread implementation of the CAN protocol, it is not surprising that vehicles can be hacked owing to vulnerabilities in the CAN protocol [3]–[9]. In terms of security, the fundamental problem of the CAN protocol is the lack of message authentication. It is difficult to include a cryptographically secure message authentication code (MAC) in a CAN message because the length of the data field in the CAN message is insufficient to include a message authentication code (MAC), such as HMAC with SHA256. Because of the limited data space, the use of MAC would require that the current system be modified. CAN FD is a newer CAN protocol and has a flexible data rate. Accordingly, methods that are designed to use MAC would be available when automotive manufacturers implement the newer version of the CAN in their vehicles for ECU communication. Thus, vehicles with the current version of the CAN protocol remain exposed to this vulnerability, which implies that an adversary could cause intentional malfunction of a vehicle. Moreover, the number of access points to in-vehicle CAN networks is increasing in response to the increasing need for driver convenience technology (e.g., telematics devices).

Automotive intrusion detection systems (IDSs) are drawing attention as a promising method for securing in-vehicle CAN networks because an IDS can be applied without generating computational overhead in the CAN protocol. We propose an automotive IDS in this paper, named VoltageIDS, for securing the current version of the in-vehicle CAN network that does not require modification of the current system. We start from the observation that even if two ECUs, one of which is legitimate and the other is malicious, were to send identical messages, the electrical characteristics of their messages would be distinguishable. VoltageIDS identifies ECUs based on these electrical characteristics, which are inherently difficult for adversaries to fake. The identification result is used to detect

Manuscript received September 27, 2017; revised January 10, 2018; accepted February 12, 2018. Date of publication March 5, 2018; date of current version April 16, 2018. This work was supported by the Samsung Research Funding and the Incubation Center for Future Technology under Project SRFC-TB1403-51. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Issa Traore. *(Corresponding author: Dong Hoon Lee.)*

W. Choi, K. Joo, M. C. Park, and D. H. Lee are with the Graduate School of Information Security, Korea University, Seoul 02841, South Korea (e-mail: wonsuk85.choi@gmail.com; wnrudgh16@korea.ac.kr; rudrnrwlska@naver.com; donghlee@korea.ac.kr).

H. J. Jo is with the Department of Computer and Information System, University of Pennsylvania, Philadelphia, PA 19096 USA (e-mail: hyojinjo@seas.upenn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2812149

the presence of in-vehicle intrusion, in which malicious CAN messages are transmitted.

A. Our Contribution

Our main contributions are as follows:

1) We analyze the shortcomings of the existing automotive IDS in Subsection II-B. In particular, we describe the vulnerability of the latest automotive IDS in the Appendix [10].
2) VoltageIDS does not require any modification of the current system, which implies that it can be immediately implemented in vehicles that are currently on the road.
3) We implement and validate VoltageIDS on a CAN bus prototype, as well as in actual vehicles. Moreover, we evaluate VoltageIDS while driving on the road, which is the first attempt amongst the automotive IDS studies that leverages the unique behavioral characteristics of ECUs. The results show that VoltageIDS detects the masquerade attack or the bus-off attack with a low error rate.
4) Since device-fingerprinting methods including VoltageIDS are known to be affected by environmental factors, such as a variation in time and temperature, VoltageIDS is designed for incremental (adding new training samples) and decremental (removing training samples) learning to address this problem. We show that VoltageIDS with incremental learning becomes robust against environmental factors. The performance is considerably improved (from an average F-score of 36.75% to an average F-score of 84.89%).
5) VoltageIDS is able to detect the bus-off attack, which has recently been demonstrated [11]. To the best of our knowledge, VoltageIDS is the first automotive IDS that distinguishes between errors and the bus-off attack.

II. Related Work

In this section, we introduce existing attacks on in-vehicle CAN networks and describe the methods used for automotive IDS and their limitations in practice.

A. Attack on In-Vehicle CAN Network

The lack of message authentication makes in-vehicle CAN bus networks vulnerable to attacks, thus compromising the reliable functioning of vehicles. In 2010, researchers provided the first evidence [7] showing that it is possible to control the functions of a vehicle by simply delivering compromising CAN messages. They analyzed the CAN traffic and found CAN messages that contain commands to control a vehicle. By injecting these malicious CAN messages, they could cause intentional malfunction of a wide array of components including the engine, instrument panel, radio, heating and cooling, lights, brakes, and locks.

Following the work of Koscher *et al.* [7], which showed that direct physical access with a laptop posed the only potential threat, researchers introduced other ways for accessing the in-vehicle CAN bus network [3]–[6], [8], [9]. In 2011, Checkoway *et al.* [3] introduced many kinds of surfaces with which to indirectly access the in-vehicle CAN networks. Compared to [7], their further work [3] showed that adversaries do not have to physically access in-vehicle CAN networks to transmit malicious CAN messages. For example, they focused on entertainment systems such as the CD player or external digital multimedia port (typically either a USB port or an iPod/iPhone docking port). Because the entertainment systems and in-vehicle CAN networks are interconnected, an adversary can deliver malicious CAN messages to the in-vehicle CAN networks by infecting the multimedia systems. Foster *et al.* [4] analyzed the vulnerability of the aftermarket telematics control unit (TCU), which is connected to in-vehicle CAN networks via the OBD-II standard port. In their demonstration, the aftermarket TCU could be remotely discovered, targeted, and compromised by an adversary. In practice, however, the attack methods they demonstrated have limitations in terms of their ability to intentionally cause vehicle malfunction. For example, the aftermarket TCU sould be be plugged into the OBD-II port [4] or a media file with malicious code would need to be played on the vehicle's audio player [3]. In other words, these attacks could only take place provided certain physical requirements are met in advance. Methods for accessing in-vehicle CAN networks on unaltered vehicles were first demonstrated in 2015 [5]. They exploited the vulnerabilities of telematics devices connected to the cellular network to remotely deliver malicious CAN messages. We note that the telematics devices they used are before-market TCUs that were installed by automotive manufacturers before releasing their vehicles.

B. Automotive IDS

Because the self-adapting nature of an IDS allows its easy adaption in the automotive domain and is a convenient extension to new vehicles, the automotive IDS for the in-vehicle CAN network has been extensively studied [10], [12]–[17]. The concept of in-vehicle intrusion detection was introduced by [13], where the characteristics of intrusion detection patterns were first presented. We note that the design of VoltageIDS was motivated by the concept of [13] using the electrical characteristics of ECUs in the physical layer. As most CAN messages are periodic, this periodicity is a universal pattern that an automotive IDS leverages for intrusion detection [16], [17]. The injection of malicious CAN messages into a CAN bus network would clearly increase the message frequency. However, a message-frequency-based automotive IDS would not be able to detect an attack by an enhanced adversary who successfully mimics the original message frequency. In fact, Greenberg [5] succeeded in disabling the preventive measure based on message frequency and could transmit malicious CAN messages. These messages caused vehicle to intentionally malfunction [10].

To overcome the limitation of the message-frequency-based automotive IDS, the behavioral characteristics were examined to discover intrusion detection patterns [10], [12], [15]. ECUs can be identified correctly based on their unique behavior characteristics. If the pair consisting of a CAN message and an identified ECU is invalid, the ECU is considered compromised. Murvay and Groza [15] proposed a method to examine the

electrical characteristics of CAN messages generated from ECUs (i.e., the CAN controllers). However, the performance of their method was such that it does not always identify ECUs correctly. Furthermore, their contribution simply comprises a conceptual suggestion to examine the electrical characteristics for ECU identification. Their method was designed without appropriately considering the arbitration decision process in the CAN protocol, as often happens in actual vehicles. When the arbitration decision process occurs, an electrical CAN signal is generated by multiple ECUs, and hence their characteristics would be blended. Other authors [12] subsequently proposed a method to examine the electrical characteristics of CAN messages for ECU identification. Compared with [15], they took the arbitration decision process into consideration, and their approach identified ECUs correctly with high performances. The limitation of their method, however, is that it uses the extended frame format that enables the use of an additional 19 bit extended identifier field, whereas the standard format is commonly applied in actual vehicles. This implies that their method requires a software modification in existing ECUs (i.e., a firmware update from CAN 2.0A to CAN 2.0B), which is infeasible in practice. Cho and Shin [10] suggested a method that does not require any modification of existing ECUs. For ECU identification, they leveraged the fact that each ECU has a unique clock skew, even if these ECUs transmit messages within the same period. Although it was previously assumed that a clock skew is difficult to imitate [10], we found that it is possible to imitate the clock skew of an ECU. Because an adversary does not have to send a malicious message at the same frequency as the target ECU, the clock skew can be imitated by delaying the difference between the clock skew of the compromised ECU and that of the target ECU. Our elaboration as to why the clock skew of an ECU can be imitated is included in the Appendix. Sagong *et al.* [18] have also demonstrated clock emulation, by which an attacker is able to inject malicious CAN messages without being detected by [10].

III. Background

In this section, we describe the signal inconsistency inherent in ECUs to provide the necessary background to understand VoltageIDS. We also introduce the CAN protocol features in the physical layer, where VoltageIDS analyzes the electrical CAN signal. Finally, we briefly describe how the bus-off attack is performed, which is a new attack on an in-vehicle CAN network. Later, we will demonstrate the ability of VoltageIDS to detect the bus-off attack.

A. Signal Inconsistency

As mentioned above, message authentication is probably impossible due to the short data field length in the CAN data frame. Rather than using message authentication, our approach to ECU identification examined the distinct characteristics of the characteristics of the electrical CAN signals corresponding to CAN messages. The basic idea in our approach stems from the observation that even if two different ECUs send the same message, there would be an inconsistency between the two signals in the physical layer. This is true even if two ECUs are identical products from the same vendor. One of the reasons why ECUs are distinguishable is that they have different cables lengths and wiring resistances that increase with wire length [19]. In fact, many methods leveraged this signal inconsistency for node identification in several applications [15], [19]–[23].

B. CAN Protocol

1) Differential Signaling/Encoding: The CAN protocol uses differential signaling that gives CAN its noise immunity and fault tolerance. In the CAN physical layer, the current flowing in each signal line is equal and opposite in direction, resulting in a field-canceling effect that is the key to low noise operation. This balanced differential signaling reduces noise coupling and allows for high signal rates over twisted-pair cable. Thus, the use of balanced differential receivers and twisted-pair cabling enhances the high noise immunity of the CAN bus. By using differential signaling, the CAN signal is encoded using the non-return to zero (NRZ) bit encoding method, in which 1 and 0 represented by significantly different voltages. In a twisted pair cable, the two shielded wires are denoted CAN-H (High) and CAN-L (Low). In the recessive state (logical 1), both wires are biased at 2.5 V. In the dominant state (logical 0), the CAN-H and CAN-L voltages are approximately 3.5 V and 1.5 V, respectively.

2) CAN Frame Type: CAN networks can be configured to accommodate two different frame formats: the standard frame format (described in CAN 2.0 A and CAN 2.0 B) and the extended frame format (only described in CAN 2.0 B). The difference is that the standard frame format supports an 11 bit identifier, while the extended frame format supports a 29 bit identifier, which is made up of the 11 bit identifier and an 18 bit extended identifier. The CAN standard identifies four frame types: a data frame, remote frame, error frame, and overload frame. All frames begin with a start-of-frame bit that denotes the start of the frame transmission. Unlike existing methods for automotive IDS [10], [12], [15], VoltageIDS covers the other frame types as well as the data frame type, which is one of the strengths of VoltageIDS. This property enables VoltageIDS to detect the bus-off attack that was newly proposed in [11]. When the bus-off attack is performed, the ECUs transmit error frames, thus increasing its error count. When the error count exceeds a predefined limit, the ECU enters its bus-off mode. We will describe the bus-off attack in following subsection. Accordingly, the error frames that are transmitted by the bus-off attacker are identified by VoltageIDS. If only data frames are analyzed, it is hard to detect the bus-off attack because all of the data frames would be normal.

The standard frame format contains fields such as those for the identifier, data length code (DLC), data, and a cyclic redundancy check (CRC) digit. Since the data field contains as many as 8 bytes, the usual approach of applying a cryptographic hash function with an output length exceeding 20 bytes is unrealistic. The identifier field refers to the identifier of the transmitter rather than the receiver. This identifier represents its priority and its meaning such as the engine temperature and the throttle valve angle. Generally, ECUs are assumed not to

share the same identifier, and one ECU has multiple identifiers. It should be noted that manufacturers keep the information about which identifiers match to which functions (i.e., ECUs) confidential.

3) Arbitration: When multiple ECUs try to simultaneously transmit CAN messages, CAN supports a lossless bit-wise arbitration decision process to prioritize such collision messages. The arbitration decision process requires that all ECUs be synchronized to sample every bit of the identifier field on the CAN bus network. If one ECU transmits a dominant bit (0) and another ECU transmits a recessive bit (1) then there is a collision and the ECU transmitting the dominant bit gets priority. Because an arbitration decision ends only within the identifier field, the simultaneous transmission of fields other than the identifier field is considered an error. This enables the higher priority message to be transmitted without any delay. That is why the identifier field is the first field in a data frame and is sometimes known as the arbitration field. VoltageIDS ignores the portion of the signal corresponding to the identifier field because it is not clear whether the signal in question was generated by a single ECU or multiple ECUs. When an arbitration decision occurs, the electrical characteristics of the identifier field are blended. Accordingly, VoltageIDS analyzes the part of the signal corresponding to the field after the identifier field.

C. Bus-Off Attack

Recently, a new attack on the CAN bus network has been suggested [11], which has been named the bus-off attack. As we mentioned above, simultaneous transmission of bits in fields other than the identifier field is considered an error under the CAN standard. The transmit error counter (TEC) of an ECU increases if the dominant state (logical 0) is detected when the ECU tries to transmit the recessive state (logical 1). In other words, the ECU increases its TEC when a bit error occurs. If the TEC of the ECU exceeds the limit of 255, it enters the bus-off mode. Upon entering this mode, the ECU is forced to shut down to prevent distraction during CAN bus communication. As a result, this situation allows the bus-off attacker to perform a kind of DoS (denial of service) attack on the in-vehicle CAN network. As the bus-off attacker exploits this error handling mechanism under the CAN standard, it is difficult to distinguish between errors and an actual attack. No countermeasure to the bus-off attack has been proposed.

IV. System Model

We explain the design of an automotive IDS based on ECU identification results. In addition, we define the adversary model to determine an adversary's goal and abilities. Finally, the attacks on the in-vehicle CAN network which covered by VoltageIDS are described.

A. ECU Identification/Automotive IDS

In this subsection, we describe the extent to which ECU identification can be leveraged for automotive IDS. ECUs are allowed to transmit CAN messages that are related to their functions. For example, the engine ECU transmits only engine-related messages. VoltageIDS first analyzes the electrical characteristics of the CAN message to identify which ECU has transmitted the CAN message. If the ECU identification result indicates one ECU that is trained in advance, VoltageIDS would verify whether the identified ECU can transmit the CAN message. If the verification is not valid, the identified ECU is considered compromised. This property is known as root-cause analysis in [10]. When an adversary transmits seemingly valid CAN messages, it is difficult for him/her to impersonate the electrical CAN signals from a target ECU. Since VoltageIDS analyzes electrical CAN signals in the physical layer not CAN messages in data layer, the adversary needs to mimic the electrical characteristics of the CAN signal and the seemingly valid CAN message simultaneously. However, it is hard to generate electrical signals by only compromising before- or after-market ECUs. The adversary needs to install an additional device that generates the electrical signal. However, the device that is physically located inside a vehicle could be easily recognized by a driver. Accordingly, the adversary who physically accesses the in-vehicle CAN network is outside the scope of our system model. We will describe our adversary model and attack type in detail as follows.

B. Adversary Model

The main goal of adversaries is to transmit malicious CAN messages to intentionally causemalfunction of a vehicle. Adversaries remotely compromise ECUs via numerous attack surfaces and methods [3], [8]. We do not consider adversaries who surreptitiously connect a malicious device to an OBD-II port like CANtact [24], [25] because it requires physical access to a vehicle and it could be easily recognized by a driver.

Automotive manufacturers have hidden or obfuscated information about the set of valid commands (i.e., CAN DB) to prevent adversaries from either injecting malicious commands or stopping/suspending proper CAN messages, even if they succeed on compromising an ECU. Despite their efforts, CAN DB has been easily analyzed. Car hacking has been demonstrated as shown in [3], [5], [7], and [9]. We consider adversaries who know information about the CAN DB of a target vehicle. Accordingly, adversaries easily cause intentional malfunction only if they have a channel to access the in-vehicle CAN network. We classify the ways for adversaries to access an in-vehicle CAN network based on types of ECUs used as access channels as follows.

Before-market ECU. This type of ECU is installed by automotive manufacturers when the vehicles were produced. One example would be to compromise a telematics ECU. Telematics ECUs are being installed in an increasing number of vehicles to provide drivers with convenient functions (e.g., remote engine start using a smartphone). Since these telematics ECUs are connected to an external network such as a cellular network, this channel might be an easy target. In fact, Miller and Valasek demonstrated remote car hacking by exploiting the vulnerabilities of telematics ECUs [5].

After-Market ECU. Many applications with the OBD-II dongles that are plugged into the OBD-II port have been released [4], [26]. These kinds of applications collect information about the condition of a vehicle from the

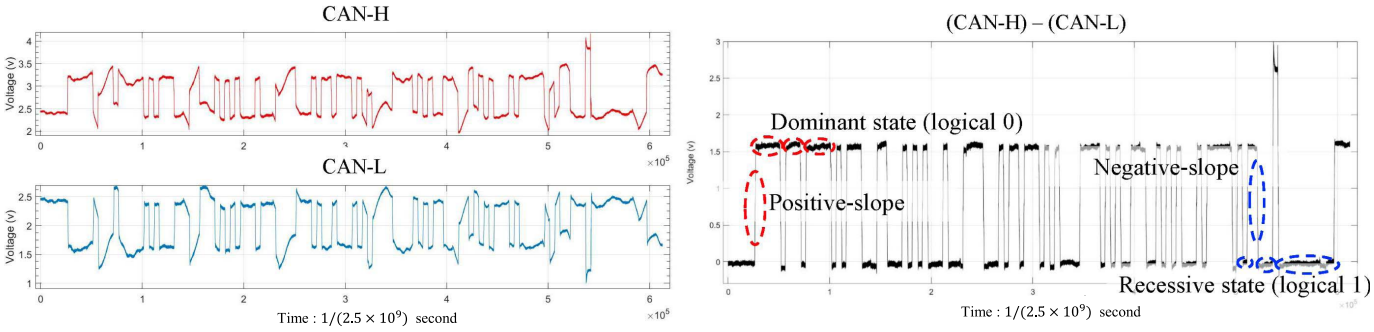


Fig. 1. Example of an electrical CAN signal.

in-vehicle CAN network traffic while providing their designated service, e.g., Pay-per-mile insurance.[1] Because the devices that are plugged to OBD-II port usually provide interconnection between the in-vehicle CAN network and an external network, they might be targets of adversaries. If these devices were compromised by adversaries, they could deliver malicious CAN messages. The demonstrated attacks in [4] and [9] are the type caused by malicious CAN messages via additional devices plugged into the OBD-II port.

C. Attack Type

We describe attacks on an in-vehicle CAN network in our adversary model. We refer to the works of [10] and [11] for the attack types covered by VoltageIDS.

1) Masquerade Attack: Cho and Shin [10] defined the masquerade attack on an in-vehicle CAN network. The objective of masquerade attacks is to manipulate an ECU that is in charge of a safe-critical function, while hiding the fact that the ECU is compromised. To mount a masquerade attack without being detected, an adversary needs to suspend a target ECU and inject malicious CAN message, thus causing malfunction of the target ECU. For example, an adversary who has access to an in-vehicle CAN network weakly compromises (i.e., suspend) a target ECU that is in charge of a safe-critical function. Then, the adversary may inject malicious CAN message by forging the CAN ID through the established channel.

In fact, Greenberg [5] demonstrated car hacking, where a Jeep Cherokee running on a highway was remotely stopped by mounting a masquerade attack [10].

2) Bus-Off Attack: In Subsection III-C, we described the bus-off attack that was suggested by Cho and Shin [11]. In this attack, an adversary who has an remote access to in-vehicle CAN network performs simultaneous transmission of bits in fields other than the identifier field. Due to this simultaneous transmission, a target ECU will enter the bus-off mode. As a result, the bus-off attacker can intentionally suspend the target ECU (i.e., DoS Attack). During the bus-off attack, error frames defined by the CAN standard are detected. Accordingly, error frames need to be analyzed to detect the bus-off attack.

[1] https://www.metromile.com/

V. Our Method

VoltageIDS is composed of three phases: i) signal measurement and preprocessing, ii) feature extraction, and iii) intrusion detection. The intrusion detection phase can be divided according to the types of attacks that are classified in Subsection IV-B. (i.e., masquerade attack and bus-off attack). In addition, We present a modified version of VoltageIDS that adapts to environmental factors.

A. Electrical CAN Signal Measurement and Preprocessing

As described in Section III, the CAN protocol uses differential signaling to ensure robust noise immunity and fault tolerance. The left part of Fig. 1 shows an example of an electrical CAN signal measured at the two signal lines in a CAN bus. Ideally, the two signal lines CAN-H and CAN-L should be passively biased to ≈2.5V in the quiescent recessive state. The dominant state on the bus should take CAN-H to be ≈1V higher, and take CAN-L to be ≈1V lower, thus creating a typical 2 V differential signal. However, the shapes of the two signals differ from their ideal expectation because of noise. The CAN protocol was designed by considering differential signaling, which is a method for electrically transmitting information using two complementary signals. The two signals have equal amplitudes relative to 2.5V (common-mode voltage) and opposite polarities. The right part of Fig. 1 shows signals with most of the noise removed. VoltageIDS reduces the amount of noise by measuring and examining the signal from the differential signaling channel rather than the individual channels, CAN-H and CAN-L.

Next, we recognize several parts of an electrical CAN signal, including its electrical characteristics. The electrical CAN signals can be divided into two states, a dominant state (logical 0) and a recessive state (logical 1). Because the dominant voltage state is actively driven by the transmitter, whereas the recessive state is passively returned to a voltage by a resistor, VoltageIDS considers the part of the dominant state that may include relatively many more electrical characteristics that enable ECUs to be identified. We ensure data independence, by using a 1 bit length dominant state part, regardless of its position. In addition, we focus on the part of the signal in which the state is changed from recessive to dominant or vice versa. These parts of the signal are referred to as the positive-slope and negative-slope parts. This transient signal is known

TABLE I

LIST OF TIME DOMAIN FEATURES. VECTOR x IS THE TIME DOMAIN REPRESENTATION OF THE DATA. N IS THE NUMBER ELEMENTS IN x

Feature	Description
Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x(i)$
Standard Deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\overline{x})}$
Average Deviation	$D = \frac{1}{N}\sum_{i=1}^{N}\lvert x(i)-\mu\rvert$
Skewness	$\gamma = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\overline{x}}{\sigma})^3$
Kurtosis	$\beta = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\overline{x}}{\sigma})^4$
RMS	$RMS = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i))^2}$
Lowest Value	$L = Min\{x(i)\lvert_{i=1..N}\}$
Highest Value	$H = Max\{x(i)\lvert_{i=1..N}\}$
ZCR	$ZCR = \frac{1}{N}\sum_{i=1}^{N}\lvert s(i)-s(i-1)\rvert$ where $s(t)=1$ if the signal has a positive amplitude at time t and 0 otherwise.
Non-negative Count	$NC = \lvert S\rvert$, where $S = \{i\lvert x(i)\geq 0\}$

TABLE II

LIST OF FREQUENCY DOMAIN FEATURES. VECTOR y IS THE FREQUENCY DOMAIN REPRESENTATION OF THE DATA. VECTORS y_m AND y_f HOLD THE MAGNITUDE COEFFICIENTS AND BIN FREQUENCIES RESPECTIVELY. N IS THE NUMBER OF ELEMENTS IN y_m AND y_f

Feature Name	Description
Spec. Centroid	$\mu = \frac{\sum_{i=1}^{N} y_f(i)y_m(i)}{\sum_{i=1}^{N} y_m(i)}$
Spec. Entropy	$H = \sum_{i=1}^{N} w_i \cdot \log_2 w_i$, where $w_i = \frac{y_m(i)}{\sum_{i=1}^{N} y_m(i)}$
Spec. Spread	$\sigma = \sqrt{\sum_{i=1}^{N}((f_i-\mu)^2\cdot w_i)}$
Spec. Skewness	$Skewness = \frac{\sqrt{\sum_{i=1}^{N}(f_i-\mu)^3\cdot w_i}}{\sigma^3}$
Spec. Kurtosis	$Kurtosis = \frac{\sqrt{\sum_{i=1}^{N}(f_i-\mu)^4\cdot w_i}}{\sigma_s^4}$
Spec. Flatness	$Flatness = \sum_{i=f_c}^{N}[y_m(i)\cdot\frac{\prod_{i=1}^{N} y_m(i))^{\frac{1}{N}}}{\frac{1}{N}\cdot\sum_{i=1}^{N} y_m(i)}]$
Spec. Brightness	$Brightness = \sum_{i=f_c}^{N} y_m(i)$, where f_c is the cut-off frequency
Spec. Roll off	$\underset{f\in 1,\ldots,N}{\operatorname{argmin}} \sum_{i=1}^{f} y_m(i) \geq 0.85\cdot\sum_{i=1}^{N} y_m(i)$
Spec. Irregularity	$Irregularity = \frac{\sum_{i=1}^{N-1}(y_m^2(i)-y_m^2(i+1))^2}{\sum_{i=1}^{N-1}(y_m^2(i))^2}$
Spec. flux	$Flux = \sum_{i=1}^{N-1}(N(i)-N(i+1))^2$, where $N(i)$ is the normalized magnitude as $N(i) = \frac{y_m(i)}{\sum_{i=1}^{N} y_m(i)}$

to contain unique properties generated by several passive (e.g., resistance) and active (e.g., capacitance) transmitter components [23], [27], [28]. As a result, VoltageIDS preprocesses the electrical CAN signal to obtain the dominant, positive-slope, and negative-slope portion. Each portion contains distinct electrical characteristics. For additional noise reduction, VoltageIDS computes the average values of each part.

B. Feature Extraction

For the extraction of the electrical characteristics from the preprocessed signals (i.e., the dominant, positive-slope, and negative-slope parts), we consider all possible features that were found to be outstanding in previous work on node identification [20], [21], [29]. We explore a total of 20 scalar features in both the time and frequency domains. Table I and Table II list the selected features in time and frequency domains, respectively. 20 features are extracted from each of the dominant, positive-slope, and negative-slope parts, thus VoltageIDS extracts a total of 60 features. At first glance, it might seem that using all features at our disposal is the best strategy to guarantee high accuracy. However, including too many features may increase the computational cost in practice. We employ a sequential feature selection procedure, which selects a subset of the total features set. Sequential forward selection (SFS) is a bottom-up search procedure that starts from an empty feature set and sequentially adds features to the set. The features are selected by an evaluation function that minimizes the misclassification error (MCE) rate. At each iteration, a feature is selected among the remaining available features of the feature set. Accordingly, the new extended features set should produce a minimum MCE compared with the addition of any other feature. SFS is widely used for its simplicity and speed. Many variants and applications have been proposed based on the SFS algorithm [30]–[34]. The result of feature selection is shown in Subsection VI-C. It is noted that SFS is just one of the methods used to select relevant features from all features [35]–[38].

C. Masquerade Attack Detection

VoltageIDS can detect the masquerade attack by creating a multi-class classifier, where the number of classes equals the number of ECUs in the in-vehicle CAN network. The method performs supervised learning with a training step and a testing step. For the training step, labeled feature sets are used to create a multi-class classifier. VoltageIDS collects the sets of features and labels them with CAN IDs. The ECU IDs refer to the identifiers assigned to ECUs for the arbitration decision as described in Subsection III-B.

Algorithm 1 describes how VoltageIDS creates a multi-class classifier and detects the masquerade attack. In the learning step, 60 features (or only the selected features) and CAN ID id_i from a CAN message are extracted and used to create a labeled set. The labeled sets are used as training data to create the multi-class classifier. We assume that no attack occurs during the training step. In the testing step, 60 features (or only the selected features) are extracted for a given new CAN message. The multi-class classifier predicts the most probable class for the unlabeled data (60 features or selected features). If the prediction result differs from the CAN ID of the CAN message, VoltageIDS detects an intrusion (i.e., masquerade attack).

We evaluated the performance of the supervised classifiers that are provided by the Classification Learner App of MATLAB R2016a [39]. We found that both the Linear SVM (Support Vector Machine) and BDT (Bagged Decision Trees)

Algorithm 1 Masquerade Attack Detection

function Supervised Learning(S: a set of signals)
  for $i = 1$ to $|S|$ **do**
    $id_i \leftarrow$ Decode($s_i \in S$) /* id_i: CAN ID */
/* D_i : dominant-level, N_i : negative-slope, P_i : positive-level, */
    $(D_i, P_i, N_i) \leftarrow$ Preprocess(s_i)
    $[f1_i, f2_i, \ldots f20_i] \leftarrow$ Extract Feature(D_i)
    $[f21_i, f22_i, \ldots f40_i] \leftarrow$ Extract Feature(P_i)
    $[f41_i, f42_i, \ldots f60_i] \leftarrow$ Extract Feature(N_i)
    Multi-labeled Training Set(i) $\leftarrow [f1_i, f2_i, \ldots, f60_i : id_i]$
  end for
  classifier $\leftarrow$ LEARNING(Multi-Labeled Training Set)
  return classifier
end function

function Masquerade Attack Detection (s: signal of a CAN message)
    id $\leftarrow$ Decode(s) /* id : CAN ID */
/* D : dominant-level, N : negative-slope, P : positive-slope, */
  $(D, P, N) \leftarrow$ Preprocess(s)
  $[f1, f2, \ldots f20] \leftarrow$ Extract Feature(D)
  $[f21, f22, \ldots f40] \leftarrow$ Extract Feature(P)
  $[f41, f42, \ldots f60] \leftarrow$ Extract Feature(N)
  prediction $\leftarrow$ Test(classifier, $[f1, f2, \ldots, f60]$
  if prediction$\neq$ id **then**
    return 1 /* Intrusion*/
  else
    return 0 /* No intrusion */
  end if
end function

Algorithm 2 Bus-Off Attack Detection

function Supervised Learning(S: a set of signals)
  for $i = 1$ to $|S|$ **do**
    $id_i \leftarrow$ Decode($s_i \in S$) /* id_i: CAN ID */
    D_i, P_i, $N_i \leftarrow$ Preprocess(s_i)
/* D_i : dominant-level, N_i : negative-slope, P_i : positive-level, */
    $(D_i, P_i, N_i) \leftarrow$ Preprocess(s_i)
    $[f1_i, f2_i, \ldots f20_i] \leftarrow$ Extract Feature(D_i)
    $[f21_i, f22_i, \ldots f40_i] \leftarrow$ Extract Feature(P_i)
    $[f41_i, f42_i, \ldots f60_i] \leftarrow$ Extract Feature(N_i)
    One-labeled Training Set(i) $\leftarrow [f1_i, f2_i, \ldots, f60_i : 1]$
  end for
  classifier $\leftarrow$ LEARNING(One-Labeled Training Set)
  return classifier
end function

function Bus-off Attack Detection (s: signal of a CAN message)
  id $\leftarrow$ Decode(s) /* id : CAN ID */
/* D : dominant-level, N : negative-slope, P : positive-slope, */
  $(D, P, N) \leftarrow$ Preprocess(s)
  $[f1, f2, \ldots f20] \leftarrow$ Extract Feature(D)
  $[f21, f22, \ldots f40] \leftarrow$ Extract Feature(P)
  $[f41, f42, \ldots f60] \leftarrow$ Extract Feature(N)
  Score $\leftarrow$ Test(classifier, $[f1, f2, \ldots, f60]$
  if Score$<$ Threshold **then**
    return 1 /* Intrusion*/
  else
    return 0 /* No intrusion */
  end if
end function

to outperform the other classifiers. We present the evaluation results using the two classifiers in Section VI.

D. Bus-Off Attack Detection

The multi-class classifier can only identify an ECU if the classifier has already been trained on that ECU's electrical characteristics. As described in Subsection IV-B, adversaries would perform simultaneous transmission of bits in fields other than the identifier field during the bus-off attack. Due to simultaneous transmission, the electrical characteristics from multiple ECUs would blend into a CAN signal. However, this signal will be identified as one of the ECUs if the electrical CAN signal was tested with the multi-class classifier. For the bus-off attack detection, VoltageIDS needs to identify this signal as unknown because it is difficult to learn all possible combinations of ECUs in advance. The method we adopt for novelty detection entails the identification of new or unknown data that were not used to train the classifier [40]–[43]. In this regard, VoltageIDS performs a simple threshold-based approach for detecting unknown signals, which is a convenient extension of our basic classification-based model. As the use of an SVM with a radial basis function (RBF) kernel is known to be effective for novelty detection, we apply it to the one-class classification technique [44]. The original purpose of one-class classification is to identify objects of a specific class amongst all objects. This is accomplished by learning from a training set containing only the objects of that class. However, we utilize the technique in a different way. For blended signal (unknown signal) detection, VoltageIDS creates a classifier for one-class classification, by considering all signals from legitimate ECUs to belong to a single class. CAN signals with a classification score (i.e., degree of matching with the legitimate ECUs) that is lower than a specified threshold are considered unknown. Algorithm 2 describes how VoltageIDS creates a one-class classifier and detects the bus-off attack. Unlike the multi-class classifier in Algorithm 1, the one-class classifier does not need labels. All sets of features have the same label in Algorithm 2.

E. Incremental/Decremental Learning

The voltage signal is known to be sensitive to environmental factors, such as variation in temperature and time. As expected, VoltageIDS is also not effective during temperature variation. Thus, we designed VoltageIDS to be robust against environmental factors by adopting incremental learning, a method

TABLE III
COMPONENTS OF EXPERIMENTAL SETUP

Components	Specification	Explanation
Oscilloscope	HDO6104 (Lecroy) Bandwidth: 1 GHz Sampling rate: 2.5 GS/s Vertical resolution: 12 bits	Measures the electrical signal corresponding to CAN messages in the physical layer of the CAN protocol
CAN node (ECU)	Arduino Uno Microcontroller: ATmega328 Clock Speed: 16 MHz CAN shield V1.2 / V2.0	Constructs a CAN bus prototype in which CAN messages are transmitted
Actual vehicle	2010 HYUNDAI YF SONATA 2.0 GDi 2014 KIA ALL NEW SOUL 1.6 GDi	Provides the experimental environment

in which input data are continuously used to update the existing classifier. In other words, the method represents a dynamic supervised learning technique. For this purpose, we selected a method that enables the SVM classifiers to be incrementally updated [45]. In Subsection VI-G, it is shown that VoltageIDS with incremental learning is robust against temperature changes. Decremental learning, which enables old data to be excluded from the classifier, could also be applied to VoltageIDS.

VI. EVALUATION

This section presents our evaluation of VoltageIDS on a CAN bus prototype and on actual vehicles. A series of experiments was carried out to justify the use of the electrical CAN signal for ECU identification.

A. Experimental Setup

Table III describes the components of the experimental setup and their specifications.

1) CAN Bus Prototype: We set up a CAN bus prototype to simulate a CAN bus network with 12 CAN nodes (ECUs). According to [46] and [47], 3–20 ECUs per CAN bus are shown in various modern vehicles (Audi A8, Honda Accord, Jeep Cherokee, Infiniti Q50, etc.). In addition, it has been reported as of 2017 that vehicles have approximately 25 ECUs, while luxury cars have approximately 50 [2]. It should also be noted that not all ECUs are on a CAN bus. Some ECUs intended for non-safety functions (e.g., body control or infotainment) are included on other networks such as LIN, MOST, or FlexRay [2]. Moreover, a vehicle has multiple in-vehicle CAN subnetworks (i.e., multiple CAN buses) to accommodate many ECUs under the bandwidth-limited CAN bus [48]. It is also noteworthy that many automotive manufacturers have been trying to integrate several ECUs into a single ECU to reduce the number of ECUs installed in a vehicle, i.e., to improve efficiency [49]. We believe that 12 CAN nodes are sufficient to simulate a CAN bus subnetwork where VoltageIDS is evaluated. Each node consists of an Arduino UNO board and a CAN shield from SeedStudio [50], [51]. The evaluation was performed at a 500-kbps baud rate, which is the most common speed of CAN bus networks in actual vehicles. Because the high-speed CAN (ISO 11898-2) network requires only two 120-Ω terminal resistors, we removed most of the resistors from CAN shield PCBs, save for two resistors.

2) Actual Vehicle: VoltageIDS was evaluated on two vehicles: a 2014 KIA SOUL, and a 2010 HYUNDA SONATA.

3) Electrical CAN Signal Acquisition: Electrical CAN signals were acquired by the oscilloscope from the CAN bus prototype or from actual vehicles. As we described in Section III, the CAN protocol uses differential signaling (i.e., CAN-H and CAN-L) to ensure robust noise immunity and fault tolerance. The oscilloscope probe was attached to the CAN-H output of two lines, and the ground clip was attached to the other CAN-L. Therefore, (CAN-H) − (CAN-L) values were obtained as shown in the right part of Fig. 1. These analogue signals can be decoded to CAN messages by the NRZ encoding rule as described in Subsection III-B. It is noted that the 9-pin D-sub type mail connector is the most common for mechanical implementation of CAN, in which Pin 2 and Pin 7 are for CAN-L and CAN-H, respectively.

4) Classification Algorithm: We used an SVM and a BDT with the default options provided by MATLAB Apps. In particular, we used a linear function as the kernel function of SVM, and 30 decision trees were used for BDT. We collected more than 150 data observations (i.e., electrical CAN signals) per ECU. The classifier was first trained with randomly selected data from these 150 data observations, and the classification model was tested with the other data per ECU. In the next subsection, we analyze how much data would be necessary for the classifier to be trained. Since the accuracy depends on the selected data, the training and testing phases were repeated 10 times for a fair evaluation. All the results we show in this section are averaged results from 10 repetitions.

5) Performance Metric: We used the standard multi-class classification metrics such as ***precision***, ***recall***, and ***F-score*** in our evaluation [52]. We computed the true positive rate (TPR_i) for each ECU_i, meaning the rate at which ECU_i has been correctly identified. Similarly, we computed the false positive rate (FPR_i) and false negative rate (FNR_i) for ECU_i, as these rates indicate wrongly accepted and wrongly rejected identifications, respectively. We then computed the precision and recall for ECU_i using the following equations:

$$\text{Precision} : \mathrm{Pr}_i = \frac{\mathrm{TPR}_i}{\mathrm{TPR}_i + \mathrm{FPR}_i}$$

$$\text{Recall} : \mathrm{Re}_i = \frac{\mathrm{TPR}_i}{\mathrm{TPR}_i + \mathrm{FNR}_i}$$

High precision and high recall indicate low FPR and low FNR, respectively. The importance of both rates is highly dependent upon the particular application. In automotive IDS, a FP state is when a CAN message is identified as an attack but the CAN message is actually normal. In other words, a FP is a false alarm. High FPR (or low precision) can cause a driver to feel uncomfortable because of false alarms. On the other hand, a FN state occurs when the IDS fails to catch an attack. A CAN message is identified as acceptable when the CAN message is actually an attack. High FNR (or low recall) can cause issues in safe driving. Thus, precision and recall are both important metrics when an automotive IDS is designed. We computed an F-score, the harmonic mean of precision and

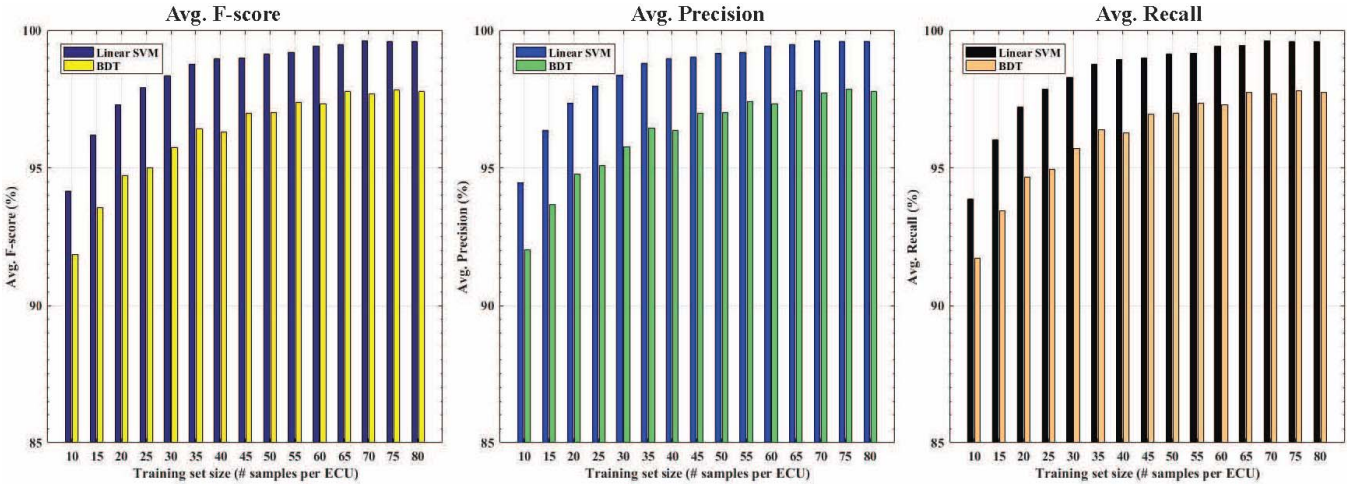


Fig. 2. Average F-score, Precision, and Recall as a function of training set size.

recall for a single metric as follows.

$$\text{F-Score} : \text{F–Score}_i = \frac{2 \times \text{Pr}_i \times \text{Re}_i}{\text{Pr}_i + \text{Re}_i}$$

We note that that it provides a good measure of the overall classification performance since precision and recall represent a tradeoff. To obtain the overall performance of VoltageIDS, we computed the average values in the following way:

$$\text{Avg. Precision} : \text{AvgPr} = \frac{\sum_{i=1}^{n} \text{Pr}_i}{n}$$

$$\text{Avg. Recall} : \text{AvgRe} = \frac{\sum_{i=1}^{n} \text{Re}_i}{n}$$

$$\text{Avg. F–Score} : \text{AvgF–Score} = \frac{2 \times \text{AvgPr} \times \text{AvgRe}}{\text{AvgPr} + \text{AvgRe}},$$

where n is the number of ECUs (i.e., classes).

B. Size of Training Set

We first evaluated the performance of VoltageIDS according to the size of the training set. We varied the size of the training set and used 100 inputs per ECU for testing. In this experiment, we considered CAN signals that only come from our CAN bus prototype. Fig. 2 shows the average F–scores, precision, and recall as a function of the training set size. We can see that the scores become higher as the size of the training set increases. When 70 samples per ECU are used for the training, VolageIDS using Linear SVM achieves an F–score of at least 99.7%, which is sufficient to correctly identify ECUs. As most CAN messages are short-periodic (i.e., 10ms - 1000ms), it does not take much time to construct a classifier with 70 samples per ECU.

C. Feature Selection

In machine learning, feature selection is the process of selecting a subset of relevant features for use in model construction. Feature selection techniques are used for three reasons: simplification of models, shorter training times, and enhanced generalization by reducing overfitting. Feature selection needs to be distinguished from feature extraction even though they are both related to the reduction of dimensionality.

TABLE IV

LIST OF SELECTED FEATURES FOR LINEAR SVM AND BDT, RESPECTIVELY

Algorithm	Feature	Type of Signal	Algorithm	Feature	Type of Signal
Linear SVM	Max	Positive Slope	BDT	Max	Positive Slope
	Max	Negative Slope		Max	Negative Slope
	Standard deviation	Dominant Level		Standard deviation	Dominant Level
	ZCR	Negative Slope		ZCR	Negative Slope
	Non-negative count	Negative Slope		Non-negative count	Negative Slope
	Standard deviation	Positive Slope		Kurtosis	Dominant Level
	Skewness	Dominant Level		RMS	Negative Slope
	Spectral irregularity	Negative Slope		Min	Negative Slope
				Spectral Kurtosis	Positive Slope

Feature extraction like principal component analysis (PCA) projects the original feature space on a new feature space with reduced dimensionality. Since either linear or non-linear combination is performed for the transformation to a low-dimensional space in feature extraction, it is required to calculate all original features before combination. In terms of computation overhead, the small number of features to be calculated is necessary because VoltageIDS analyzes CAN messages and its electrical CAN signals in real time. We leverage a sequential feature selection procedure which selects a subset of features by sequential addition (forward search). Fig. 3 shows the misclassification error (MCE), which is the number of misclassified observations divided by the number of total observations on the test set as a function of the number of features. We can see that the MCE remains constant when the number of features exceeds eight and nine for both Linear SVM and BDT, respectively. Therefore, selecting the highest number of possible features does not always achieve superior performance and sometimes causes an overfitting problem. There is little difference between the selected features in the two lists included in Table IV.

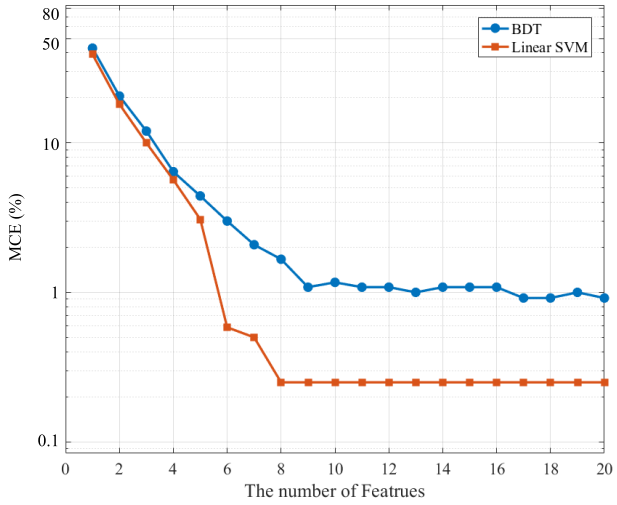


Fig. 3. Misclassification error (MCE) as a function of the number of features.

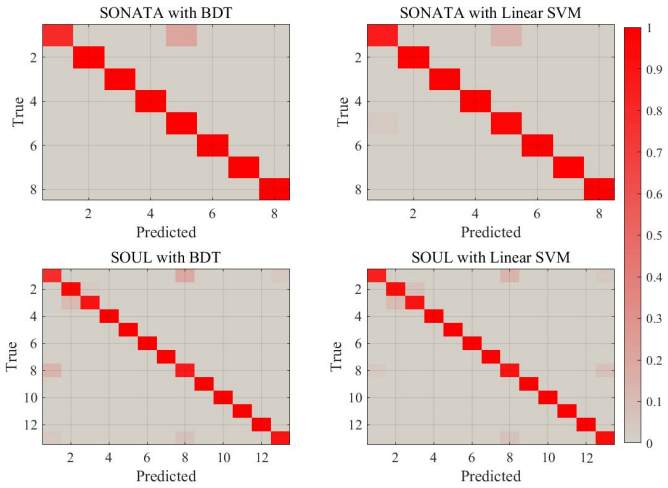


Fig. 4. Identification result using BDT and Linear SVM on diagnostic CAN IDs from actual vehicles: confusion matrix as a heat map.

D. Masquerade Attack Detection on Actual Vehicle

In this subsection, we present our evaluation of VoltageIDS for the detection of masquerade attacks on actual vehicles. For the evaluation on actual vehicles, we checked the number of CAN IDs assigned to in-vehicle CAN networks. As one of the in-vehicle CAN subnetworks, we selected the diagnostic CAN (DCAN) bus network, which is easily accessed via the OBD-II port. We then measured the electrical CAN signals in the DCAN bus network. In the 2010 Hyundai Sonata and 2014 Kia Soul, 27 and 45 distinct CAN IDs were assigned to CAN messages, respectively.

Fig. 5 shows the identification result on the data from the DCAN bus network of the two vehicles. The confusion matrix is not clear, which implies a low average F-score. For the Sonata, linear SVM and BDT output an average F-score of 54.76% and 59.54%, respectively. For the Soul, the corresponding output was 54.24% and 60.96%, respectively. The main reason for these low average F-scores is that multiple CAN IDs are assigned to a single ECU. If two CAN messages with different CAN IDs were to originate from the same ECU, VoltageIDS would not be able to distinguish between them correctly. In fact, the information regarding which message originates from which ECU is not public. Instead, automotive manufacturers try to keep this information confidential. As an alternative, we considered that each ECU has a particular ID for the diagnostic protocol [8]. Each ECU has a pair of IDs: a request ID and response ID. The response ID always exceeds the request ID by 8. For example, if we transmit a diagnostic packet with a request ID of 0x760, the ECU corresponding to it would respond with 0x768. We refer to this type of ID as the diagnostic ID to distinguish it from the normal CAN ID. We also assume that ECUs use only a single diagnostic CAN ID. However, the diagnostic CAN ID used in functional addressing (e.g., OBD-II PIDs) can be shared by multiple ECUs, we fixed the request SID (Service Identifier) to obtain CAN data from only a single ECU per diagnostic CAN ID.

By querying a diagnostic service with 11 bit identifiers from 0x700 to 0x7FF, we obtained 8 and 13 distinct diagnostic CAN IDs from the Sonata and Soul, respectively. Fig. 4 shows the identification result on the diagnostic CAN IDs. As can be seen in the figure, the result we obtained is clearer than that for normal CAN IDs. For the Sonata, linear SVM and BDT output an average F-score of 96.31% and 95.01%, respectively. For the Soul, they output average F-scores of 97.20% and 96.57%, respectively. We suspect that F-scores that are lower than average compared to the result from our CAN bus prototype arise because this experiment was conducted outside. The temperature variations during the experiment were relatively large. Because the voltage is generally known to decrease with increasing temperature, the temperature change would cause a shift in the characteristics of the electrical CAN signals from the ECUs. It should be noted that it took 3 to 4 hours to measure the electrical CAN signal from the vehicles. The results of our evaluation on the extent to which the characteristics are affected by temperature change are presented in Subsection VI-G. However, the results from actual vehicles within a small temperature variation is acceptable to correctly identify ECUs. Thus, we conclude that VoltageIDS can detect intrusion from a masquerade attack, even on actual vehicles. In addition, we expect that VoltageIDS would have output a higher average F-score, given the exact information about the CAN IDs according to the ECUs. The result we obtained for feature selection on the actual vehicles differed from that on the CAN bus prototype, despite the size of the learning datasets and the identical number of features in both cases. This implies that feature selection of VoltageIDS needs further study.

E. Masquerade Attack Detection for Vehicle in Motion

We evaluated the extent to which the electrical characteristics are affected by the motion of a vehicle. Unlike idling vehicles, transmitting diagnostic messages to the in-vehicle CAN network while driving might be dangerous (i.e., we were not sure if the vehicle might behave unexpectedly). Many warning signs actually turned on when we transmitted diagnostic packets into the in-vehicle CAN network. Thus, we would not have been able to drive the vehicle while ignoring these signs. As an alternative safe

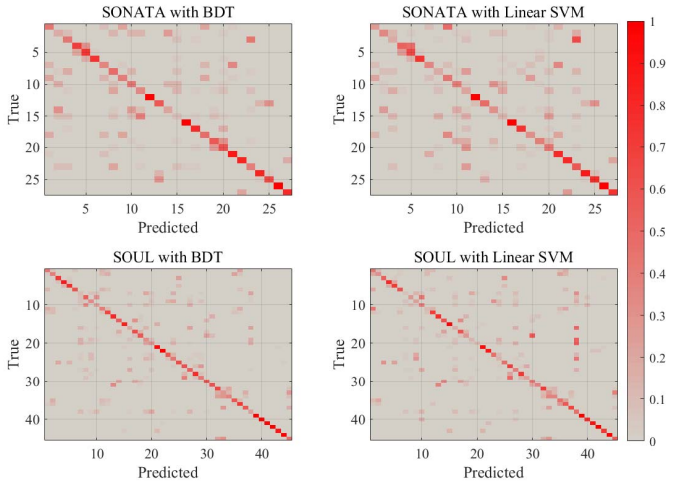


Fig. 5. Identification result using BDT and Linear SVM on CAN IDs of actual vehicles: confusion matrix as a heat map.

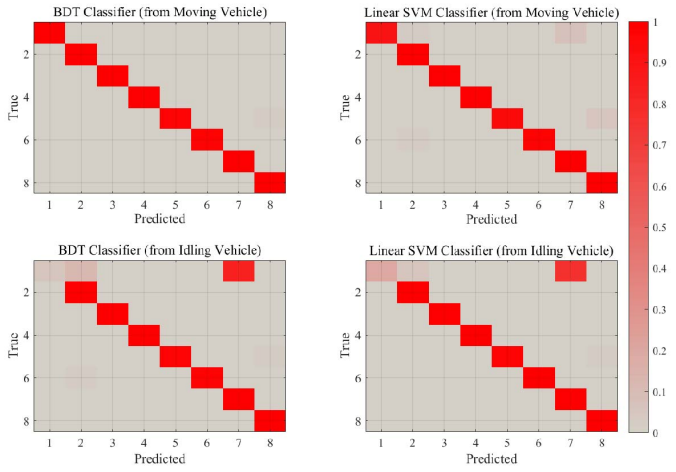


Fig. 6. Identification result on a moving vehicle: confusion matrix as a heat map.

method, we needed to find a subset of normal CAN IDs for which a single ECU is guaranteed to have only a single CAN ID. We created the multi-class classifier with the diagnostic CAN IDs. We then tested CAN messages with normal CAN IDs to check which normal CAN ID corresponds to a diagnostic CAN ID. By using this classification, we found 8 CAN IDs out of a total of 45 CAN IDs for the 2014 Kia Soul. We measured the electrical CAN signal corresponding to these 8 CAN IDs instead of diagnostic CAN IDs while driving. Once again, we did not have exact information about which message originates from which ECU. Our aim was to ensure that the signal would be indiscriminately affected by a variety of conditions e.g., high/low speed driving, driving around a corner, and driving on uneven roads. Therefore, we also measured the signal slowly for about 2 hours.

The top of Fig. 6 shows the identification result on the vehicle in motion. Linear SVM and BDT output average F-scores of 97.84% and 99.61%, respectively. Thisimplies that ECUs can be correctly identified even in moving vehicles. Next, we created another classifier with a set of signals that was

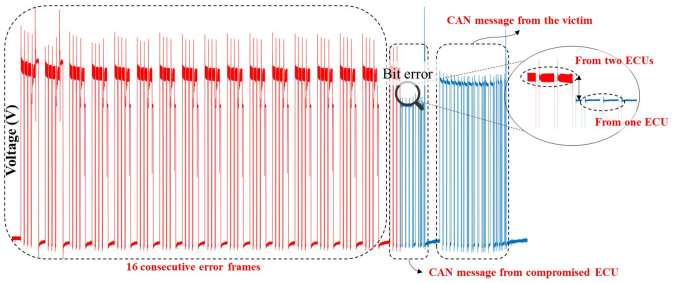


Fig. 7. Example of 16 consecutive error frames and CAN messages due to the bus-off attack.

measured when the vehicle was idling. Then, we again tested the signals measured from the moving vehicle. The bottom of Fig. 6 shows the confusion matrix using the classifier created from electrical CAN signals from the idling vehicle. Both algorithms output average F-scores of 90.77% and 90.01%, respectively. This result indicates that the characteristics of ECUs from a moving vehicle do not change much compared to those from an idling vehicle. In fact, only one ECU shows different characteristics in Fig. 6. We infer that the change in characteristics might stem from the specific location of the ECU in the vehicle. To the best of our knowledge, this is the first experiment that was conducted on a moving vehicle, compared to other methods in [10], [12], and [15].

F. Bus-Off Attack Detection

As we mentioned in Subsection III-C, it is difficult to distinguish between errors and the bus-off attack. However, this difference can be clearly recognized in the physical layer. In the bus-off attack, the compromised ECU simultaneously transmits the same bits as the target ECU until a bit error occurs. Hence, the electrical CAN signal originates from two ECUs in the case of the bus-off attack, and its voltage level is different from the electrical CAN signal generated by a single ECU. Fig. 7 shows an example of a series of error frames due to the bus-off attack. After 16 consecutive error frames, the signal of the compromised ECU is first shown before the signal of the target because the victim ECU enters error-passive mode. Fig. 7 clearly shows the difference between the voltage levels of two ECUs (i.e., the red part shows the first 16 frames and the beginning part of the next CAN message from the compromised ECU) and the voltage level of the signal from the single victim ECU (i.e., the blue part shows the lower voltage part of the CAN message from the compromised ECU and CAN message from the victim). We note that the signal from the identifier field can come from more than two ECUs, even under normal conditions. This occurs because ECUs are allowed to simultaneously transmit the same data during the arbitration decision procedure. This implies that a signal from the location other than the identifier field should be examined for detection of a bus-off attack. Because the three bits after the identifier field (i.e., the RTR, IDE, and reserved bit) are always dominant (i.e., 0), the DLC field located after the three bits is where the bit error of the bus-off attack can occur. VoltageIDS could be modified to effectively detect the bus-off attack such that it examines the part corresponding to these three bits. Recall that we focus on the dominant level, negative

TABLE V
ERROR RATES IN THE BUS-OFF ATTACK DETECTION

SVM score with RBF kernel		FPR (%)
Mean	Standard Deviation	(threshold of -0.5)
-20.6737	1.372e-6	0.00

slope, and positive slope. For each portion, 20 features are used. It is possible that a stuff bit does not reside in between the three bits after the ID field. It is noted that a bit with opposite state (dominant or recessive) is inserted as a stuff bit after five consecutive states to ensure a sufficient number of transitions required to maintain synchronization. If more than two consecutive states before RTR are dominant, a recessive state may exist amongst these three bits. Otherwise, there may be no state change and neither a negative-slope nor positive-slope would exist. VoltageIDS uses only 20 features from the dominant-level signal. We generated 100 pairs of signals causing a bit error for the bus-off attack. Then, VoltageIDS performed the procedure for the bus-off attack detection. While legitimate signals all have scores of around 0 from the results of SVM with RBF kernel classification, the 100 pairs of signals causing a bit error all have scores around -20.6737, as shown in Table V. It should be noted that a score of 0 indicates complete matching to a legitimated one. VoltageIDS needs a threshold that clearly separates the bus-off attack signal and the legitimate signals. Table V shows that FPR is 0 when the threshold is set as -0.5. In our experiment, we obtained the result in which every legitimate signal has score below -0.1. FPR refers to the rate at which a valid signal is considered to be malicious. Thus, we conclude that VoltageIDS is able to detect a bus-off attack.

G. Effect of Temperature

Methods using device fingerprinting are expected to be affected by the shift in fingerprinting characteristics due to temperature variation [10], [12], [15], [20], [21], [23], [29], [53], [54]. Most of these researchers did not evaluate their methods on the shift in the characteristics in terms of temperature changes. We tested the effect of temperature variation on electrical CAN signals. Generally, it is well known that the voltage decreases as temperature increases. It should be noted that the change in voltage is not solely affected by the change in temperature. We measured the electrical CAN signals of 13 diagnostic CAN IDs from the 2014 Kia Soul on two different days: 01/16/2017 and 02/4/2017. We intentionally waited for a day when the temperature difference between the two days was approximately 10 degrees Celsius. The classifier was created from the electrical CAN signals that were measured on the first day (01/16/2017). We tested the classifier with the other electrical CAN signals that were measured on the second day (02/4/2017). The left of Fig. 8 shows the effect of temperature variations on the identification result, for which the linear SVM outputs an average F-score of 37.65%. Interestingly, certain ECUs are very sensitive to temperature change, while other ECUs are robust to it. This difference is inferred to be due to the different locations of ECUs insdie the vehicle. THus, they can have different exposures and sensitivities to temperature variations.

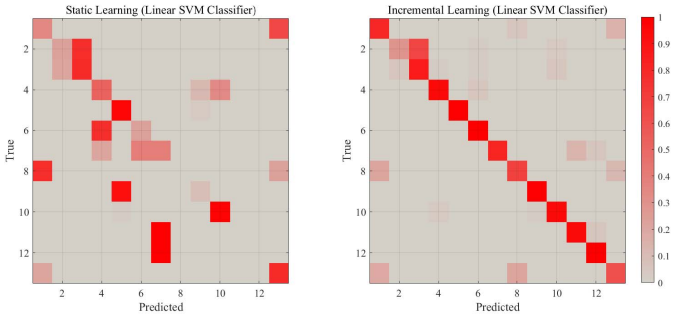


Fig. 8. Identification result affected by temperature variation: confusion matrix as a heat map.

In addition, we used VoltageIDS with incremental learning, which enables classifiers to be adaptively updated by adopting the implementation of [55]. This approach involved continuously using the input signals to update the classifier. Hence, the input signal was classified by the classifier that had been trained with all test signals before examining the input signal. The right of Fig. 8 shows the confusion matrix in which linear SVM with incremental learning outputs an average F-score of 84.89%. Compared to the previous result, VoltageIDS with incremental learning is robust against temperature variation.

To simulate the bus-off attack in which two ECUs simultaneously transmit a pair of messages causing a bit error, we implemented an alternative method. This involves plugging two CAN shields into an Arduino board. This method enables simple transmission-time synchronization.

H. Low-Performance Oscilloscope

We repeated the experiments with low-performance oscilloscopes to show that VoltageIDS is effective, even at low cost. In addition, a low-performance oscilloscope can be helpful to enhance the execution time of VoltageIDS. For example, the smaller number of samples obtained using a low sampling rate requires less time to handle it. In VoltageIDS, the sampling rate and the vertical resolution are the most important factors that contribute to the performance of oscilloscopes. We used 8 bit vertical resolution, which is common in most oscilloscopes. The sampling rate was determined by testing VoltageIDS as a function of the sampling rate because its range is too wide to select a particular rate. This evaluation can help select an oscilloscope with suitable specification. We measured the electrical CAN signal at the sampling rate of 2.5 GS/s and 8 bit vertical resolution. The signal was preprocessed to remove some of the measured samples and achieve the same effect as a lower sampling rate. Fig. 9 shows the average F-scores as a function of sampling rate. For 2.5 GS/s, linear SVM and BDT output average F-scores of 98.94% and 97.28% on the CAN bus prototype, respectively, and 94.14% and 91.71% on the actual vehicle. Because of the small difference in average F-scores compared to the results with 12 bit vertical resolution, an 8 bit vertical resolution is expected to be sufficient to enable VoltageIDS to identify the ECUs. The average F-scores decreased with decreasing sampling rate. For 250 MS/s, however, VoltageIDS still achieves acceptable performance. Linear SVM and BDT

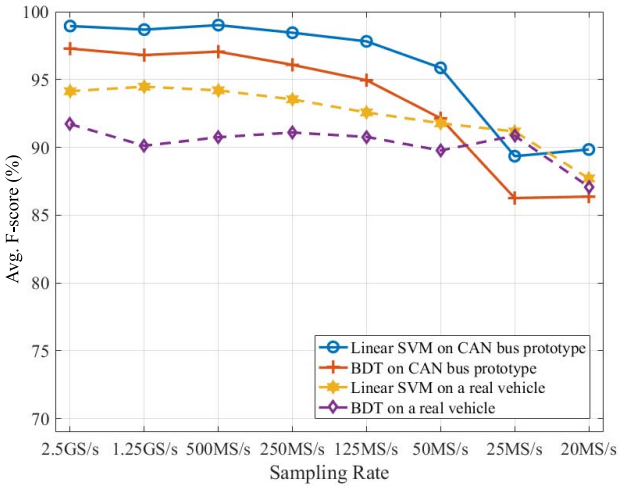


Fig. 9. Average F-score as a function of sampling rate.

output average F-scores of 98.45% and 96.08% on the CAN bus prototype, respectively, and 93.54% and 91.09% on the actual vehicle, respectively.

Based on this result, we surveyed oscilloscopes that enable VoltageIDS to be implemented at reasonable costs. We once again measured the electrical CAN signals on our CAN bus prototype using the oscilloscopes in [56] and [57]. These are on sale for less than $200. We then performed VoltageIDS on their output signals. For the oscilloscope of [56], the signals were measured at a sampling rate of 250 MS/s with 8 bit vertical resolution and a bandwidth of 70 MHz. Linear SVM and BDT output average F-scores of 95.43% and 94.54%, respectively. For the oscilloscope of [57], the signals were measured at a sampling rate of 20MS/s with 5-bit vertical resolution and a bandwidth of 10 MHz in which Linear SVM and BDT output average F-scores of 90.01% and 79.79%, respectively. Unfortunately, even though the oscilloscope of [57] has better accuracy, it was impossible to use the maximum when we measured the electrical CAN signals.

VII. Discussion

In this section, we discuss the limitations of VoltageIDS and our future research.

A. Multiple CAN IDs Per ECU

One limitation VoltageIDS is unable to overcome is that the information relating to which CAN ID belongs to which ECU is not publicly known, which degrades the performance of VoltageIDS. We believe that this problem should be addressed by vehicle manufacturers.

B. Potential Attack

According to [46], the vehicle power source (i.e., battery charging status) has a non-negligible effect on the electric signals generated by ECUs. As time goes by, the electric signal-based IDS including VoltageIDS would incorrectly detect intrusions. It is necessary to learn frequently based on power source condition variations. However, frequent learning could be a critical vulnerability, which means that adversaries have lots of chances to inject malicious signals. During learning phase of VoltageIDS, an adversary may inject malicious signals. This causes VoltageIDS to learn the wrong features. To handle attacks performed during the learning phase, a context-aware updates approach that update learning data by checking context information [46] should be applied in VoltageIDS. For example, the status of the car battery charge is one the important pieces of contextual information that affects the electric signals generated by the ECU.

A new version of the in-vehicle network protocol. CAN-FD [58] and automotive Ethernet [59] protocol is being developed to replace the current version of CAN protocol. They have higher bit rates and sufficient data payload size, which are commonly required by authentication approaches. Accordingly, it is expected that the authentication approaches will be a common way to secure these communication protocols. However, an automotive IDS will also be necessary, which is able to complement authentication methods limitations, such as node compromise or performance delay due to computational overhead. Because CAN-FD is an extended version of the CAN protocol, the electric signal is generated in the same way as the CAN protocol. It is reasonable to expect that VoltageIDS can detect intrusion, even on CAN-FD protocol. Unfortunately, VoltageIDS cannot be applied to automotive Ethernet protocol. However, the work proposed in [22] might work at the automotive Ethernet protocol because it is designed to identity the Ethernet card based on electric signals. Thus, we expect that VoltageIDS will play an important role even when a new protocol becomes commonly used for in-vehicle networks.

VIII. Conclusion

We have presented VoltageIDS, which is able to detect in-vehicle CAN network intrusions based on the inimitable characteristics of electrical CAN signals. As the proposed method only requires the installation of a monitoring unit on the CAN bus network without any modification of the current ECUs, it can be directly and transparently applied to current vehicles. We also provided a series of experimental results to show the feasibility of VoltageIDS in practice. In particular, VoltageIDS would be the first method evaluated on moving as well as idling vehicles. The method is also shown to be capable of detecting the recently introduced bus-off attack [11]. To the best of our knowledge, this is the first method that is able to address this attack properly. Therefore, we conclude that VoltageIDS is a feasible and effective approach for securing an in-vehicle CAN network.

Appendix

Cho and Shin [10] presented clock-based intrusion detection system (CIDS) for intrusion detection in in-vehicle CAN networks. CIDS leverages the fact that each ECU has a unique clock skew that can be used as a pattern to be examined for intrusion detection. They succeeded in calculating the clock offset and clock skew of ECUs based on the fact that every ECU periodically transmits CAN messages with constant length. They also demonstrated that CIDS is effective on idling vehicles without any modification of existing ECUs. It is assumed that an adversary is not able to imitate the clock

TABLE VI
REVISED NOTATION

Notation	Description
ECU R	The ECU works as CIDS which calculates $skew_x$
ECU A	The ECU compromised by an adversary
ECU V	The victim ECU
$skew_x$	The clock skew of messages ECU X sends
$a_{X,t}$	The arrival timestamp of the message ECU X sends at its local time t
$O_{X,t}$	The clock offset of the message ECU X sends at time t
$O_X[k]$	The average clock offset of N message ECU X sends at k step

skew of a target ECU. Even though the clock skew is known to be difficult to imitate, we found that imitation is possible in CAN communication. Before providing an explanation as to how to imitate the clock skew, we first explain how CIDS works.

In CIDS, ECU R computes the clock offset and clock skew based on the difference between the true and expected arrival times. In addition to implementing effective noise removal, they calculated the averaged clock offsets O[k] as follows:

$$\mathrm{O}[k] = \sum_{i=2}^{N} [a_i - (a_1 + (i-1)\mu_T[k-1])],$$

where a_i is the arrival timestamp of the i-th CAN message from the ECU. $\mu_T[k-1]$ is the average timestamp interval of the $(k-1)$-th step, which is considered as $\mu_T[k-1] = T$. For CIDS to be effective, the following two assumptions are required:

1) ECUs periodically transmit CAN messages at a constant frequency.
2) The data lengths of CAN periodic messages are constant over time.

However, adversaries do not have to comply with these assumptions. For instance, adversaries may transmit a CAN message at a constant frequency that is different from the target ECU. Because CIDS is not able to check this modification, imitating the clock skew is possible without being detected. As a result, we found that CIDS cannot detect clock skew-imitating intrusions. If an adversary delays message transmission by the difference between the clock offset of the compromised ECU and that of the target ECU, CIDS incorrectly decides that the clock skew is valid.

Table VI contains the notations we revised to explain the vulnerability of CIDS based on our analysis. As skew_R is assumed to be 0, the clock offset at time t between ECU V and ECU R is $\mathrm{O}_{V,t} = \mathrm{skew}_V \times t$. The relative clock offset at time t between ECU V and ECU A is $(\mathrm{skew}_V - \mathrm{skew}_A) \times t$. Accordingly, our idea is to delay transmitting messages by the ratio of the relative clock offset for imitation of the skew of ECU V. The exact time to be delayed d is calculated as follows.

$$\begin{aligned} \mathrm{skew}_A \times (t+d) &= \mathrm{skew}_V \times t \\ \Longrightarrow d &= \frac{(\mathrm{skew}_V - \mathrm{skew}_A) \times t}{\mathrm{skew}_A} \end{aligned}$$

This implies that the clock offset at $(t+d)$ between ECU A and ECU R is the same as the clock offset at t between ECU V and ECU R (i.e., $\mathrm{O}_{A,(T+d)} = \mathrm{O}_{V,T}$). Because CIDS utilizes the average clock offset of N messages, the clock offset $\mathrm{O}_A[k]$ corresponding to the delayed messages sent by ECU A is calculated as follows.

$$\begin{aligned} &\mathrm{O}_A[k] \\ &= \frac{1}{(N-1)} \sum_{i=2}^{N} a_{A,i(T+d)} - (a_{A,(T+d)} + (i-1)\mu_{T+d}[k-1]) \\ &= \frac{1}{(N-1)} \sum_{i=2}^{N} ((i(T+d) + \mathrm{O}_{A,i(T+d)}) \\ &\quad - (T + d + \mathrm{O}_{A,T+d} + (i-1)(T+d))) \\ &= \frac{1}{(N-1)} \sum_{i=2}^{N} (\mathrm{O}_{A,i(T+d)} - \mathrm{O}_{A,(T+d)}) \\ &= \frac{1}{(N-1)} \sum_{i=2}^{N} (\mathrm{O}_{V,iT} - \mathrm{O}_{V,T}) = \mathrm{O}_V[k] \end{aligned}$$

As a result, CIDS is not able to detect the delayed messages. Instead, it incorrectly decides that the messages came from ECU V rather than ECU A.

REFERENCES

[1] *Growing Number of ECUS Forces New Approach to Cars Electrical Architecture*. Accessed: Aug. 3, 2017. [Online]. Available: http://www.newelectronics.co.uk/electronics-technology/growing-number-of-ecus-forces-new-approach-to-car-electrical-architecture/45039/

[2] AMPG Body Electronics Systems Engineering Team, "Future advances in body electronics," NXP, Eindhoven, The Netherlands, 2017. [Online]. Available: https://www.nxp.com/docs/en/white-paper/BODYDELECTRWP.pdf

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Secur. Symp.*, San Francisco, CA, USA, 2011, pp. 1–16.

[4] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures," in *Proc. 9th USENIX Workshop Offensive Technol.*, 2015, pp. 1–9.

[5] A. Greenberg, "Hackers remotely kill a jeep on the highway—With me in it," *Wired*, vol. 7, p. 21, Jul. 2015.

[6] H. J. Jo, W. Choi, S. Y. Na, S. Woo, and D. H. Lee, "Vulnerabilities of Android OS-based telematics system," *Wireless Pers. Commun.*, vol. 92, no. 4, pp. 1511–1530, 2017.

[7] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 447–462.

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

[9] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Apr. 2015.

[10] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 911–927.

[11] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1044–1055.

[12] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee. (2016). "Identifying ECUs using inimitable characteristics of signals in controller area networks." [Online]. Available: https://arxiv.org/abs/1607.00497

[13] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive can networks—Practical examples and selected short-term countermeasures," in *Proc. Int. Conf. Comput. Safety, Rel., Secur.*, 2008, pp. 235–248.

[14] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2008, pp. 220–225.

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

[16] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp. (4)*, Jun. 2011, pp. 1110–1115.

[17] M. Müter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *Proc. 6th Int. Conf. Inf. Assurance Secur. (IAS)*, Aug. 2010, pp. 92–98.

[18] S. U. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran. (2017). "Cloaking the clock: emulating clock skew in controller area networks." [Online]. Available: https://arxiv.org/abs/1710.02692

[19] K.-D. Kang, Y. Baek, S. Lee, and S. H. Son, "An analysis of voltage drop as a security feature in controller area network," in *Proc. IEMEK Symp. Embedded Technol.*, 2016, pp. 1–2.

[20] A. Das, N. Borisov, and M. Caesar. (2015). "Exploring ways to mitigate sensor-based smartphone fingerprinting." [Online]. available: https://arxiv.org/abs/1503.01874

[21] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi, "Accelprint: Imperfections of accelerometers make smartphones trackable," in *Proc. NDSS*, 2014, pp. 1–16.

[22] R. M. Gerdes, M. Mina, S. F. Russell, and T. E. Daniels, "Physical-layer identification of wired Ethernet devices," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 4, pp. 1339–1353, Aug. 2012.

[23] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase," in *Proc. Wireless Opt. Commun.*, Jul. 2003, pp. 13–18.

[24] *Cantact, The Open Source Car Tool*. Accessed: Dec. 20, 2017. [Online]. Available: http://linklayer.github.io/cantact/

[25] E. Evenchick, "Hopping on the can bus: Automotive security and the CANard toolkit," in *Proc. Black Hat Asia*, 2015, pp. 1–31.

[26] L. Guan *et al.*, "From physical to cyber: Escalating protection for personalized auto insurance," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. (CD-ROM)*, 2016, pp. 42–55.

[27] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2009, pp. 25–36.

[28] K. B. Rasmussen and S. Capkun, "Implications of radio fingerprinting on the security of sensor networks," in *Proc. 3rd Int. Conf. Secur. Privacy Commun. Netw. Workshops (SecureComm)*, Sep. 2007, pp. 331–340.

[29] S. Davidson, D. Smith, C. Yang, and S. Cheah, "Smartwatch user identification as a means of authentication," Dept. Comput. Sci. Eng., Univ. California San Diego, La Jolla, CA, USA, 2016.

[30] M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000.

[31] D. P. Muni, N. R. Pal, and J. Das, "Genetic programming for simultaneous feature selection and classifier design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 106–117, Feb. 2006.

[32] M. Last, A. Kandel, and O. Maimon, "Information-theoretic algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 22, nos. 6–7, pp. 799–811, 2001.

[33] S. Nakariyakul and D. P. Casasent, "An improvement on floating search algorithms for feature subset selection," *Pattern Recognit.*, vol. 42, no. 9, pp. 1932–1940, 2009.

[34] J. Schenk, M. Kaiser, and G. Rigoll, "Selecting features in on-line handwritten whiteboard note recognition: SFS or SFFS?" in *Proc. 10th Int. Conf. Document Anal. Recognit. (ICDAR)*, Jul. 2009, pp. 1251–1254.

[35] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.

[36] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop Mach. Learn.*, 1992, pp. 249–256.

[37] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[38] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 1–8.

[39] *Classification Learner, Mathworks*. Accessed: Feb. 14. 2017. [Online]. Available: https://www.mathworks.com/help/stats/classificationlearner-app.html?s_tid=gn_loc_drop

[40] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, 2009.

[41] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.

[42] M. Markou and S. Singh, "Novelty detection: A review—Part 1: Statistical approaches," *Signal Process.*, vol. 83, no. 12, pp. 2481–2497, 2003.

[43] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, Mar. 2001.

[44] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.

[45] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proc. NIPS*, 2000, pp. 1–7.

[46] K.-T. Cho and K. Shin. (2017). "Viden: Attacker identification on in-vehicle networks." [Online]. Available: https://arxiv.org/abs/1708.08414

[47] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *Proc. Black Hat USA*, 2014, pp. 1–94.

[48] I. Foster and K. Koscher, "Exploring controller area networks," in *Proc. USENIX Assoc.*, 2015, pp. 1–6.

[49] *The Competitiveness of Automobiles Depends on ECU Integration*. Accessed: Aug. 3, 2017. [Online]. Available: http://english.etnews.com/20131113200002

[50] *Arduino Uno*. Accessed: Aug. 3, 2017. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardUno

[51] *Can-Bus Shield V1.2*. Accessed: Aug. 3, 2017. [Online]. Available: http://wiki.seeed.cc/CAN-BUS_Shield_V1.2/

[52] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[53] D. Formby, P. Srinivasan, A. Leonard, J. Rogers, and R. Beyah, "Who's in control of your control system? Device fingerprinting for cyber-physical systems," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016, pp. 1–15.

[54] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing speech from gyroscope signals," in *Proc. USENIX Secur.*, 2014, pp. 1053–1067.

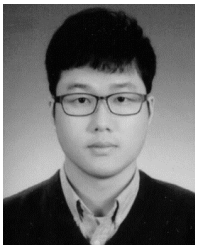
[55] C. P. Diehl and G. Cauwenberghs, "SVM incremental learning, adaptation and optimization," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2003, pp. 2685–2690.

[56] *IDSO Series, Hantek*. Accessed: Aug. 3, 2017. [Online]. Available: http://www.hantek.com/en/ProductDetail_10165.html

[57] *Bitscope Micro*. Accessed: Aug. 3, 2017. [Online]. Available: http://www.bitscope.com/product/BS05/

[58] F. Hartwich and R. Bosch, "Can with flexible data-rate," in *Proc. ICC*, 2012, pp. 1–9.

[59] P. Hank, S. Müller, O. Vermesan, and J. Van Den Keybus, "Automotive Ethernet: In-vehicle networking and smart mobility," in *Proc. Conf. Design, Autom. Test Eur.*, 2013, pp. 1735–1739.



Wonsuk Choi received the B.S. degree in mathematics from the University of Seoul, Seoul, South Korea, in 2008, and the M.S. degree in information security from Korea University, Seoul, in 2013, where he is currently pursuing the Ph.D. degree in information security with the Graduate School of Information Security. His research interests include applied cryptography, healthcare security, and authentication and key exchange in sensor networks.

Kyungho Joo received the B.S. degree from the College of Information and Communication, Korea University, Seoul, South Korea, in 2016, where he is currently pursuing the M.S. degree in information security with the Graduate School of Information Security. His research interests include vehicular-IT security, wireless security, and applied cryptography.

Hyo Jin Jo received the B.S. degree in industrial engineering and the Ph.D. degree in information security from Korea University, Seoul, South Korea, in 2009 and 2016, respectively. He is currently a Post-Doctoral Researcher with the Department of Computer and Information System, University of Pennsylvania, Philadelphia, PA, USA. His research interests include cryptographic protocols in authentication, applied cryptography, security and privacy in ad hoc networks, and smart car security.

Moon Chan Park received the B.S. degree in mathematics from the University of Seoul, Seoul, South Korea, in 2013, and the M.S. degree in information security from Korea University, Seoul, in 2015, where he is currently pursuing the Ph.D. degree in information security with the Graduate school of Information Security. His research interests include applied cryptography and software security.

Dong Hoon Lee (M'06) received the B.S. degree from the Department of Economics, Korea University, Seoul, in 1985, and the M.S. and Ph.D. degrees in computer science from The University of Oklahoma, Norman, in 1988 and 1992, respectively. Since 1993, he has been with the Faculty of Computer Science and Information Security, Korea University. He is currently a Professor and the Director of the Graduate School of Information Security, Korea University. His research interests include the design and analysis of cryptographic protocols in key agreement, encryption, signatures, embedded device security, and privacy-enhancing technology.