IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 30, 2020, accepted February 13, 2020, date of publication February 24, 2020, date of current version March 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975893

Normal and Malicious Sliding Windows Similarity Analysis Method for Fast and Accurate IDS Against DoS Attacks on In-Vehicle Networks

SHUJI OHIRA[1], (Student Member, IEEE), ARAYA KIBROM DESTA[1], ISMAIL ARAI[2], (Member, IEEE), HIROYUKI INOUE[3], (Member, IEEE), AND KAZUTOSHI FUJIKAWA[2], (Member, IEEE)

[1]Graduate School of Science and Technology, Nara Institute Science and Technology, Ikoma 630-0192, Japan
[2]Information Initiative Center, Nara Institute of Science and Technology, Ikoma 630-0192, Japan
[3]Graduate School of Information Sciences, Hiroshima City University, Hiroshima 731-3194, Japan

Corresponding author: Shuji Ohira (ohira.shuji.ok2@is.naist.jp)

ABSTRACT Controller Area Network (CAN) is a de facto standard of in-vehicle networks. Since CAN employs broadcast communication and a slower network than other general networks (e.g. Ethernet, IEEE802.11), it is inherently vulnerable to Denial-of-Service (DoS) attacks. As a countermeasure against DoS attacks on CAN, a method for detecting a DoS attack using the entropy in a sliding window has been proposed. This method has a good advantage in terms of effectiveness and the small computational overhead. However, this method may only be effective against DoS attacks under naive conditions such as some higher priority messages. In addition, if an adversary can adjust the entropy of the DoS attack to its normal value, the conventional method cannot detect a DoS attack in which the adversary manipulates the entropy. We found this type of DoS attack, which is called an *entropy-manipulated attack*. In this paper, we propose a method that can detect an entropy-manipulated attack by using the similarity of two sliding windows. We confirmed that the proposed method detected the DoS attack in 100% of the cases in our experiment, and we showed that the detection time is up to 93% (14 μs) shorter than the conventional method.

INDEX TERMS Automotive security, controller area network, DoS attack, intrusion detection system, simulated annealing.

I. INTRODUCTION

The security risk of cyberattacks on modern vehicles has become a concern due to the spread of vehicles connected to the internet [1], [2]. These attacks abuse the security hole of In-Vehicle Infotainment (IVI) and vulnerability of Controller Area Network (CAN) [3] which is a de facto standard form of in-vehicle networks. Miller and Valasek successfully controlled a variety of automotive functions through IVI, and they exploited the vulnerabilities of CAN. As a result, 1.4 million automobiles were recalled because of this vulnerability. Therefore, cybersecurity countermeasures for automobiles are urgently required. In addition, software in the automotive industry has become open source software (e.g. Automotive Grade Linux Distribution: AGL [4]) for enhancing the reusability of source codes. If the software has vulnerabilities, there is a danger of malware infection which exploits the vulnerabilities related to cybersecurity [5]. Thus, the intrusion detection capabilities of the in-vehicle networks should be improved to avoid serious damage from hacking.

Countermeasures such as encryption and authentication have been proposed to prevent spoofing, sniffing, and replay attacks in next-generation in-vehicle networks (e.g. CAN with Flexible Data rate: CAN FD, In-Vehicle Ethernet). These proposals could disable traditional hacking, in which an adversary sends a spoofed message to a specific Electronic Control Unit (ECU). However, even in CAN bus applied to encryption and authentication, these proposals cannot disable a Denial-of-Service (DoS) attack that sends many messages to the in-vehicle network, because DoS attacks can delay the encrypted normal messages. To disable DoS attacks, a security solution different from encryption and authentication is necessary. In order to prevent a DoS attack of one type in which an adversary sends messages to flood the buffer of

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

receiving ECU, Intrusion Prevention System (IPS) defenses have been proposed [6]–[8]. However, these methods do not affect other DoS attacks, in which an adversary sends many messages of the highest CAN ID priority. Therefore, it is necessary to have an IPS that can prevent all DoS attacks. To prevent all DoS attacks, firstly, we must consider a method to detect all DoS attacks.

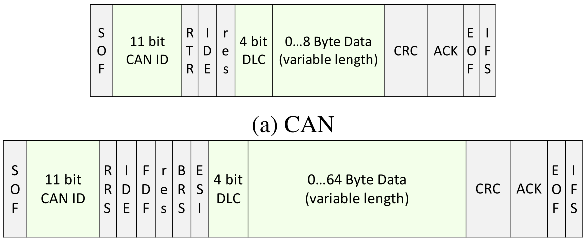
Time-intervals Intrusion Detection System (IDS) [9] has been proposed to detect spoofing attacks and DoS attacks on CAN. This IDS detects DoS attacks with the cutoff of the time interval to 0.2 milliseconds for detecting DoS messages. However, in the case of over 0.2 milliseconds of the time interval of the DoS attack's messages, the IDS cannot detect DoS attacks. In addition, in different baud rates such as CAN and CAN FD, the IDS cannot be adapted to the CAN buses because the time intervals of DoS attacks are different. Furthermore, some methods [9]–[11] may only be effective against the DoS attacks under naive environments, such as some higher priority messages [12].

A machine learning approach for IDS has been proposed. The approach can widely detect attacks such as spoofing, replaying, and DoS attacks with high accuracy. The approach (especially deep learning) is too expensive to implement the training function in the vehicle, although the cost of inferring is reasonable. Also, a secure Over-the-Air (OTA) update [13] has been proposed in modern automotive. Therefore, the cost of training the additional communication of the OTA update should be reasonable.

Also, an entropy-based IDS [11] using the entropy of a fixed number of messages, called a sliding window, has been proposed against the DoS attacks and the replay attacks. This method has a good advantage in terms of effectiveness and the small computational overhead [14]. However, the entropy might be manipulated by adversaries to avoid the IDS. To solve this problem, we propose a method that can detect DoS attacks of all types using the Simpson coefficient as the similarity of two sliding windows. One of the two sliding windows is composed of CAN IDs (Window IDs: *WIDs*) which are actually received, and the other is composed of normal CAN IDs (Criterion IDs: *CIDs*) which serves as a criterion to calculate the similarity. Also, *CIDs* is generated as an optimized parameter using the Simulated Annealing (SA) algorithm in the proposed method.

The main contributions of this study can be summarized as follows.

1) We found a DoS attack called the *entropy-manipulated attack*, which bypasses the conventional method (entropy-based IDS [11]) by adjusting the entropy of messages. These consist of messages of random CAN ID.
2) The proposed method (similarity-based IDS) achieved a detection precision of 100.0% against the above type of DoS attacks, while the detection precision is 68.3% in the entropy-based IDS.
3) We showed that the detection time is up to 93% (14 μs) shorter than the entropy-based IDS.



(a) CAN

(b) CAN FD

FIGURE 1. CAN and CAN FD message format.

The rest of the paper is organized as follows. In Section II, we explain the background and attack model. In Section III, we propose a DoS attack detection method based on the similarity of sliding windows. In section IV, we evaluate the similarity-based IDS. In Section V, we discuss the result of evaluation from Section IV. Finally, we elaborate on our conclusions.

II. PRELIMINARIES

A. CAN VULNERABILITY

1) CAN

CAN [3] is a de facto standard form of in-vehicle networks. With the increase of ECUs in cars, the wiring of automobiles has become complicated. In order to solve this wiring harness problem, a CAN was designed with noise resistance and capable of event-oriented mutual communication. For enhancing CAN capacity, CAN FD was released in 2012 [15] as an extended CAN in response to the demand for a capability in automobiles to send more data. Figure 1 shows the standard CAN and CAN FD message frame. The green area in Figure 1(a) and (b) are data that can be observed in the application layer programs (e.g. can-utils [16]). In Figure 1, the 11-bit identifier is called CAN ID, which is used for prioritizing CAN messages. The 4-bit Data Length Code (DLC) contains the number of bytes of data being transmitted. The DLC has a length of 4 bits, and this can have an integer value between 0 and 15. In the case of a CAN FD, a 4-bit DLC is taken from a corresponding table. For example, if DLC is 15 (0 × 1111), the size of the data field is 64-bytes in the CAN FD. Then, 64-byte data fields have various data (e.g. sensor values, gear position, turn signal). Also, unlike CAN FD, the CAN's 8-bytes of data fields have various data. The rest of the fields shaded in gray are handled in the physical layer.

CAN uses a bitwise arbitration ID to control the priority of messages. If logical "0" and logical "1" are transmitted on the bus at the same time, "0" takes precedence. Due to this characteristic, a low CAN ID (e.g. 0 × 000) of destination information is handled as an ID with a high priority. If an adversary exploits the bitwise arbitration, the adversary can easily achieve a DoS attack on CAN. In addition, due to the features of broadcast communication, no authentication and, low bandwidth, CAN is vulnerable against sniffing, spoofing, replay attacks, and DoS attacks. Also, Cook and

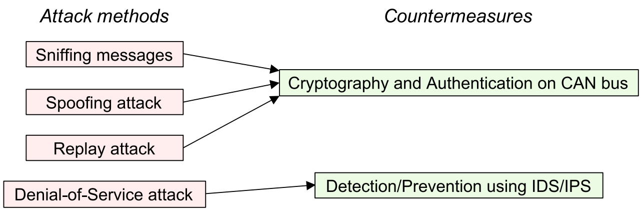
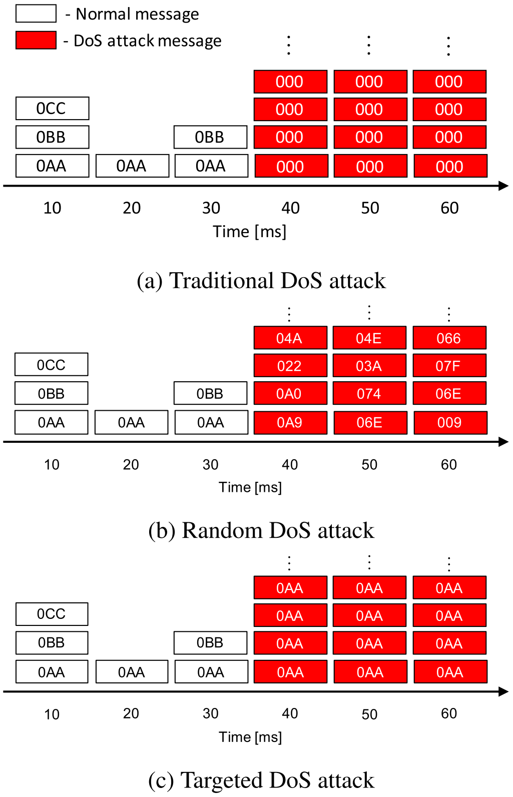


FIGURE 2. **CAN vulnerabilities and the countermeasures.**



(a) Traditional DoS attack

(b) Random DoS attack

(c) Targeted DoS attack

FIGURE 3. **The classification of DoS attacks on CAN.**

Freudenberg [17] claimed that the major drawback of CAN is that message latency is non-determinant (due to the existence of some error frames and retransmissions), and latency increases with the amount of traffic on the bus. Therefore, bus utilization should not exceed 30% of the CAN bus capacity to assure that low priority messages do not experience an unacceptable delay. It implies that even a compromised in-vehicle computer with limited resources can delay CAN message with DoS attacks.

CAN attack methods and countermeasures are summarized in Figure 2, which shows that encryption and authentication methods could disable a sniffing, spoofing, and replay attack in next-generation in-vehicle networks (e.g. CAN FD). However, to disable a DoS attack, a security solution different from encryption and authentication is necessary.

2) ATTACK MODEL

There are three types of DoS attacks on CAN according to the report [6] of Humayed *et al.* as follows. (Also, shown in Figure 3.)

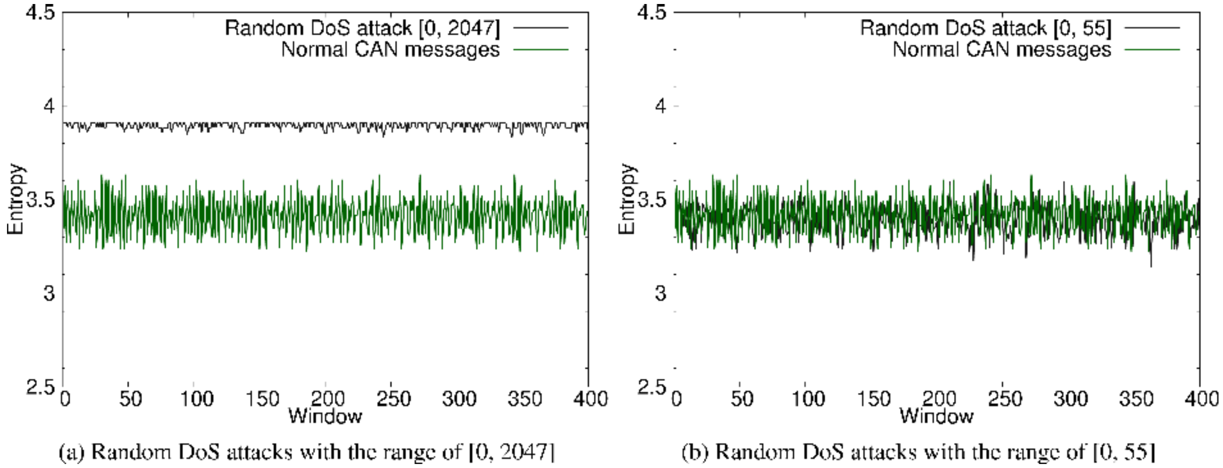
a) Traditional DoS attack
An adversary can easily interfere with a CAN bus by using bitwise arbitration. Since the lower CAN ID has a higher priority, the adversary would use CAN ID 0 × 000 for the DoS attack. As a result, the adversary can induce unexpected behavior of the vehicle. Though it is not difficult to detect the Traditional DoS attack, IDSs must detect it as soon as possible.

b) Random DoS attack
A Random DoS attack is the most appropriate attack for broadcasting incorrect values without investigating an in-vehicle network. Messages with a random CAN ID from 0 to 2047 can be transmitted within one second, even on a low-speed network. The difficulty of the detection of the Random DoS attack is the same with Traditional DoS attacks and should be detected as soon as possible too. An entropy-based IDS can detect Random DoS attacks of both extremes of entropy. However, the entropy-based IDS cannot detect an entropy-manipulated attack in which an adversary adjusts the entropy of a DoS attack to a normal value. For example, in case of the sliding window $W = 30$, entropies of two sliding windows are the same value if a sliding window includes 10 messages of normal CAN IDs 0 × 0AA, 0 × 0BB, and 0 × 0CC, respectively, and a sliding window includes 10 messages of the DoS message's IDs 0 × 000, 0 × 001, and 0 × 002, respectively. If an adversary exploits the entropy-manipulated attack, the adversary can bypass the entropy-based IDS. The conditions for an entropy-manipulated attack are as follows.

1. The entropies of the Random DoS attack and the normal messages are the same values.
2. The CAN IDs of the Random DoS attack are lower than the CAN IDs of normal messages.

Here, we describe that the realization of the entropy-manipulated attack. Firstly, after an adversary intrudes an ECU from some external network, the adversary sniffs the messages of CAN. Next, the adversary can calculate the window and the entropy in the CAN using the same algorithm to the entropy-based IDS. Finally, the entropy-manipulated attack is executed using the window and entropy. IDS should assume adversaries know the algorithm inside it [18].

Second, we describe the possibility, which entropy-manipulated attacks are interrupted by other legitimate CAN messages. The average total number of CAN IDs in six car models is 53 IDs. In addition, most of the CAN messages in the six car models have a CAN ID of 0 × 053 or more. Therefore, if an adversary exploits CAN IDs of high priority from 0 × 000 to 0 × 053 for a Random DoS attack, the adversary can imitate the same entropy as legitimate data. In other words, the adversary cannot be interrupted by other legitimate CAN messages so that the attacks are higher priority than the legitimate CAN messages. Therefore,



(a) Random DoS attacks with the range of [0, 2047]

(b) Random DoS attacks with the range of [0, 55]

FIGURE 4. Vulnerability of the entropy-based IDS.

we can conclude that the entropy-manipulated attack is reproducible by an adversary.

c) Targeted DoS attack
A Targeted DoS attack influences buses and ECUs. In this research, we assume an attack on one ECU and define it as a DoS attack using one ID flowing on the bus. This DoS attack could have life-threating consequences for the driver and passengers. However, entropy-based IDS can be used to achieve to detect this DoS attack.

B. RELATED WORKS

1) INTRUSION DETECTION SYSTEM

A lot of works have been done on IDSs on CAN. IDSs on CAN using deep learning have been proposed [19], [20]. The approach is too expensive to implement the training function in the vehicle although the cost of inferring is reasonable. Also, a secure OTA update [13] has been proposed in modern automotive. Therefore, the cost of training the additional communication of the OTA update should be reasonable.

Time-intervals IDS [9] has been proposed to detect spoofing attacks and DoS attacks on CAN. This IDS detects DoS attacks with the cutoff of the time interval to 0.2 milliseconds for detecting DoS messages. However, in the case of over 0.2 milliseconds of the time interval of the DoS attack's messages, the IDS cannot detect DoS attacks. In addition, in different baud rates such as CAN and CAN FD, the IDS cannot be adapted to the CAN buses because the time intervals of DoS attacks are different. Furthermore, some methods [9]–[11] may only be effective against the DoS attacks under naive environments such as some highest priority messages [12].

Detection methods based on electrical fingerprint information have been proposed [21], [22]. However, in order to perform electrical fingerprint information-based anomaly detection, some additional hardware such as the A/D converter is necessary. Furthermore, if an original ECU is compromised, the IDS cannot detect a malicious message using CAN ID assigned in compromised ECU itself.

There are various other related studies, but these studies are not superior to the entropy-based IDS in terms of effectiveness to DoS attacks and the small computational overhead. Hence, we will summarize the comparison in Section V-C.

2) ENTROPY-BASED IDS

The IDSs using entropy have been proposed [23], [24]. Wu *et al.* pointed out that because these IDSs use fixed-time messages (sliding windows) for calculating entropies, they cannot be applied to different transmission rates. Therefore, they proposed a novel entropy-based IDS [11], showing that the entropy-based IDS can fastly detect DoS attacks by using a sliding window (a fixed number of messages) for calculating the entropy. The definition of entropy in the method is as follows, where $I = \{id_1, id_2, id_3, \ldots, id_n\}$ is a set of different CAN IDs appearing within sliding windows W. Equation (1) is expressed as the entropy of CAN IDs in sliding windows W.

$$H(I) = -\sum_{id_i \in I} P(id_i)\log(P(id_i)) \tag{1}$$

Next, we explain the $P(id_i)$ in Equation (1). Since the method determines the network state by monitoring CAN messages per window W, the total number of messages in the arbitrary network state is the same to window W. Thus, the total number of messages N_{total} in the sliding windows W can be obtained by Equation (2):

$$N_{total} = \sum_{i=1}^{n} Count_{id_i} \tag{2}$$

The $Count_{id_i}$ is the number of id_i appearing in W. Then the probability of id_i appearing in W can be represented as

$$P(id_i) = \frac{Count_{id_i}}{N_{total}} \tag{3}$$

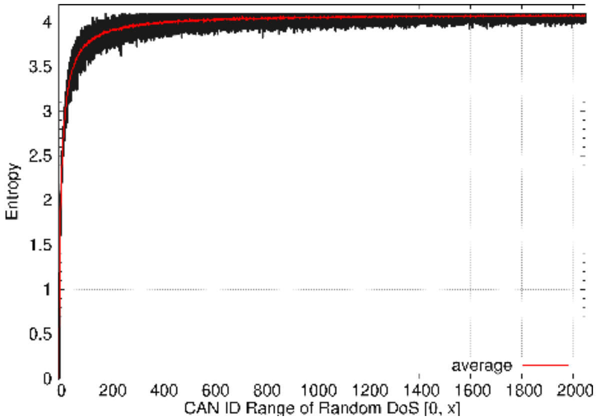


FIGURE 5. Entropy of Random DoS attack under different CAN ID range.

The definition of Equation (1), (2), and (3) is based on the entropy-based IDS [11]. The problem with the entropy-based IDS [11] is that it has a much higher FP rate against the entropy-manipulated attack. We show the example of the entropy-manipulated attack. Figure 4 illustrates our test car's temporal change of entropy against two Random DoS attacks. Figure 4(a) and (b) are randomly generated by Random DoS attacks with CAN IDs of a range of [0, 2047] and [0,55] respectively. From Figure 4(a), we confirm that the entropy-based IDS can distinguish the normal CAN messages and Random DoS attacks of the extremes of entropy. Figure 4(b) shows that the entropies of normal messages and Random DoS attacks are the same value. This attack is an entropy-manipulated attack. Also, Figure 5 shows the entropy of Random DoS attacks under the various ranges of CAN IDs. It shows that an adversary can inject a Random DoS attack using arbitrary entropy. If the entropy of a Random DoS attack and the entropy of normal traffic are the same value, like Figure 4 (b), an adversary can bypass the entropy-based IDS. This problem is caused by that the entropy defined by [11] is based only on the randomness of the CAN ID. In other words, an adversary can configure sliding windows with the same randomness with completely different CAN IDs which are higher priority than normal CAN messages. Therefore, to detect the entropy-manipulated attacks and the other DoS attacks, we consider an approach that can detect the entropy-manipulated attacks based on whether a CAN IDs' set in a sliding window is a normal range.

III. PROPOSED SIMILARITY-BASED IDS

In Section II-B.2, we confirmed that the entropy-based IDS could not detect the entropy-manipulated attack. Therefore, we aim to detect the entropy-manipulated attack and the other DoS attacks with the similarity-based IDS. The entropy-based IDS focuses only on the degree of randomness of CAN IDs in a sliding window, but the IDS does not focus on the individual values of CAN IDs in a sliding window. In other words, an adversary can configure sliding windows with the same randomness with completely different CAN IDs which are higher priority than normal CAN messages. Therefore, the entropy-manipulated attack bypasses the entropy-based IDS.

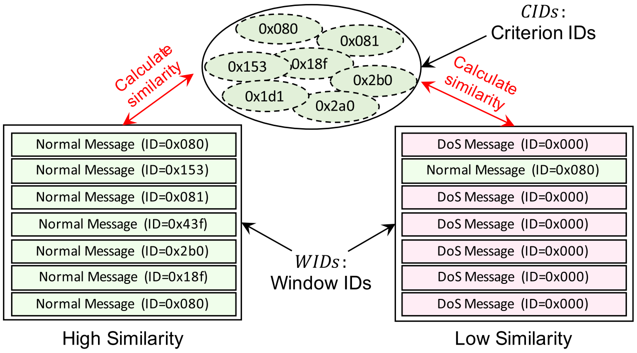


FIGURE 6. Example of ***WIDs*** and ***CIDs***.

In order to detect anomalies based on the degree of randomness of CAN IDs and the individual values of CAN IDs in a sliding window, we propose an IDS based on the similarity of the sliding windows rather than the entropy of the sliding windows. Our similarity-based IDS calculates the similarity in Window IDs (called *WIDs*) and Criterion IDs (called *CIDs*) using the Simpson coefficient, which expresses the similarity between the sets. Figure 6 shows an example of *WIDs* and *CIDs*. As shown in Figure 6, our similarity-based IDS detects DoS attacks based on the similarity between *WIDs* and *CIDs*. In addition, in order to optimize the anomaly detection precision and detection time, we use the SA algorithm. The SA algorithm is used to obtain a good local solution, so that the entropy-based IDS [11] indicated the effectiveness of the SA algorithm.

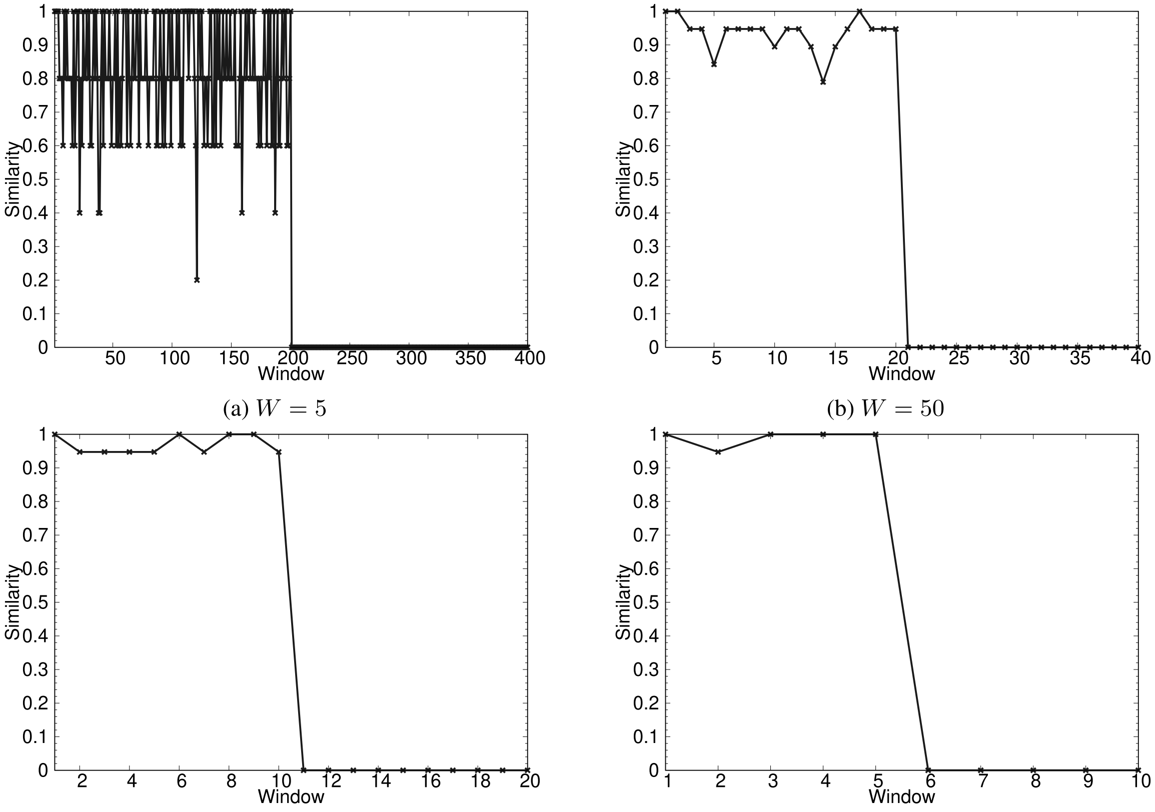
Detection time is an important evaluation metric because fast detection can conduct intrusion preventions (e.g. ID-Hopping Mechanism) rapidly. Therefore, we consider decreasing the detection time.

A. DEFINITION OF SIMILARITY IN CAN

Our similarity-based IDS calculates the similarity in *WIDs* and *CIDs* using the Simpson coefficient (often called the Overlap coefficient), which expresses the similarity between the sets. If the two sets have an intersection, the Simpson coefficient of the two sets is higher than the Jaccard coefficient and the Dice coefficient. Also, since some non-cyclic messages are sent in CAN, similarity decreases even in normal messages. Therefore, if a part of the two sets is not shared such as non-cyclic messages, the Simpson coefficient that the decreasing of similarity is most low is the most suitable similarity in CAN.

The definition of similarity in CAN is as follows. Equation (4) is used to calculate the similarity between *CIDs* and *WIDs* in CAN, where *CIDs* are a set of bases to calculate the similarity, and *WIDs* are a set of CAN IDs in a sliding window, W, of a fixed number of messages.

$$\mathrm{overlap}(CIDs, WIDs) = \frac{|CIDs \cap WIDs|}{\min(|CIDs|, |WIDs|)} \tag{4}$$



(a) $W = 5$

(b) $W = 50$

(c) $W = 100$

(d) $W = 200$

FIGURE 7. Comparison of similarity under different sliding window message sizes.

If we use normal CAN messages to calculate the similarity, *CIDs* and *WIDs* may possess several elements of the same CAN ID. Due to this fact, *CIDs* and *WIDs* are multisets. Moreover, Equation (4) can be transformed into Equation specified in (5) because $|CIDs|$ and $|WIDs|$ are always same as $|W|$.

$$\text{overlap}(CIDs, WIDs) = \frac{|CIDs \cap WIDs|}{|W|} \tag{5}$$

We use Equation (5) to calculate the similarity in the proposed similarity-based IDS. Also, note that $|W|$ is a constant value in the On-line detection phase, if the denominator of Equation (5) is incremented whenever a CAN message is received, the Overlap coefficient can be calculated in $\mathcal{O}(1)$. It is a smaller value than $\mathcal{O}(\log(|N|))$ of the entropy calculation of the entropy-based IDS, where N is the number of unique CAN ID included in a sliding window.

Next, we measure similarity under different sliding windows as a preliminary experiment, in which we used preliminary CAN messages that are composed of 1000 messages of both normal and DoS attacks. These preliminary CAN messages of the first half are normal ones, and the rest is the DoS attack messages of CAN ID 0 × 000. We plot similarity under different sliding windows in Figure 7. We use CAN IDs optimized by the Off-line learning phase as *CIDs* to calculate similarity. Figure 7 shows how different sliding windows W affect the similarity value. For example, when $W = 5$, the similarity value is between 0.2 and 1.0, but when W increases to 50, the similarity value changes to a range of values between 0.8 and 1.0. Furthermore, when $W = 100$ or 200, the similarity value reaches almost 1.0. As shown in Figure 7 (b), we have confirmed that when $W = 50$, the distance of similarities between normal CAN messages and DoS attacks is great. Hence, the Off-line learning phase optimally selects a sliding window size in the range of [5, 50].

B. FRAMEWORK OF THE SIMILARITY-BASED IDS

The similarity-based IDS has two phases, an Off-line learning phase and an On-line detection phase (Figure 8). In the first phase, the SA algorithm is used to collect optimal parameters; in the later phase, we detect anomalies by using the optimal parameters collected in the first phase.

a) Off-line learning phase
The Off-line learning phase mainly includes the following steps.

- **Step 1:** Extract the CAN IDs from learning traffic.
- **Step 2:** Calculate the similarity between *WIDs* and *CIDs*. *WIDs* are a set of CAN IDs of a certain window. *CIDs* are a set of normal CAN IDs defined for intrusion detection. Also, *CIDs* serves as a criterion to calculate the similarity.
- **Step 3:** Judge whether the similarity (details are shown in Section III-A) calculated in Step 2 falls

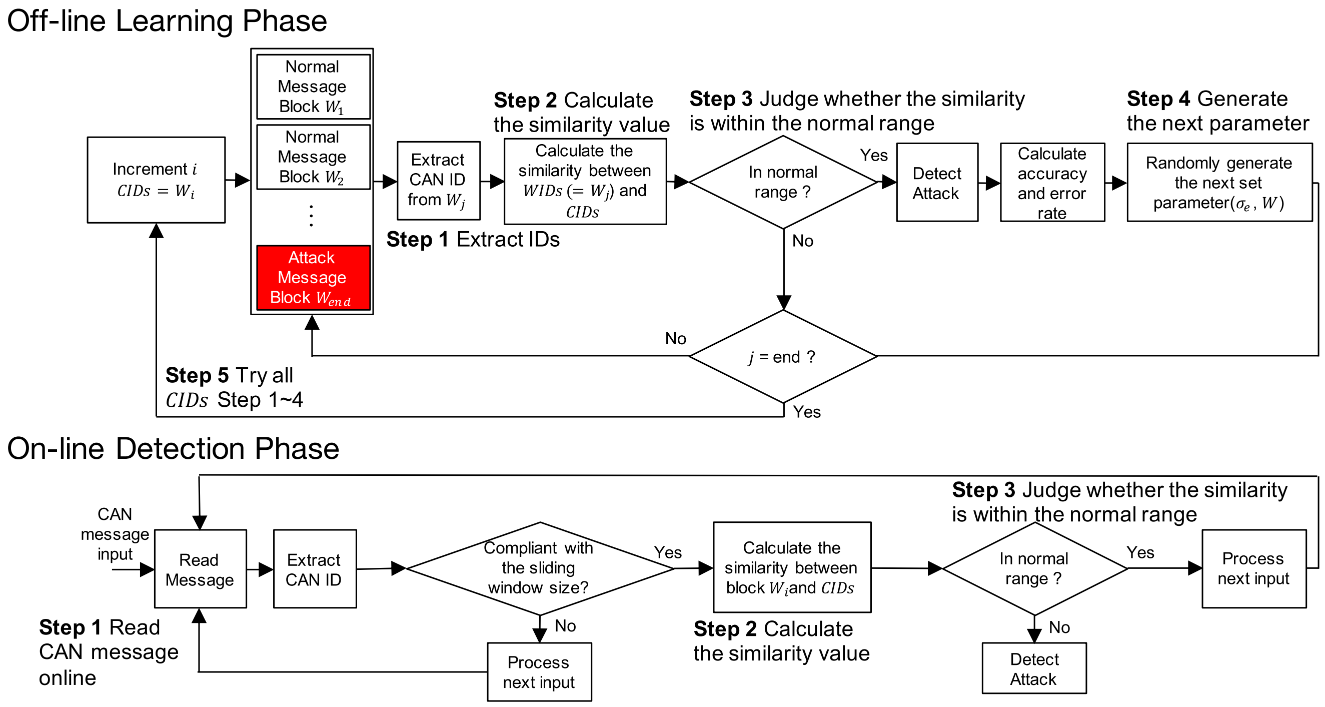

Off-line Learning Phase
Increment i $CIDs = W_i$
Normal Message Block W_1
Normal Message Block W_2
⋮
Attack Message Block W_{end}
Extract CAN ID from W_j
Step 1 Extract IDs
Step 2 Calculate the similarity value
Calculate the similarity between $WIDs$ (= W_j) and $CIDs$
Step 3 Judge whether the similarity is within the normal range
In normal range ?
Yes
No
Detect Attack
Calculate accuracy and error rate
Step 4 Generate the next parameter
Randomly generate the next set parameter(σ_e, W)
No
j = end ?
Yes
Step 5 Try all $CIDs$ Step 1~4
On-line Detection Phase
CAN message input
Read Message
Extract CAN ID
Step 1 Read CAN message online
Compliant with the sliding window size?
Yes
No
Process next input
Calculate the similarity between block $W_i$and $CIDs$
Step 2 Calculate the similarity value
Step 3 Judge whether the similarity is within the normal range
In normal range ?
Yes
No
Process next input
Detect Attack

FIGURE 8. Flow of the similarity-based IDS.

within the normal range randomly generated by the SA algorithm.

- **Step 4:** Select the new deviation of the normal range and the sliding window for the next loop.
- **Step 5:** Try all the *CIDs* in the CAN messages for learning after that determine the *CIDs* to calculate the best score. Steps 1-4 are designed based on the SA algorithm. Step 5 is designed to try all the *CIDs* through steps 1-4 for high accuracy.

b) On-line detection phase

The On-line detection phase mainly includes the following steps.

- **Step 1:** Read a CAN message on-line and collect messages until the number of messages is the same as the size of the sliding window.
- **Step 2:** Calculate the similarity between *WIDs* and *CIDs*.
- **Step 3:** Judge whether the similarity calculated in Step 2 falls within the normal range generated in the Off-line learning phase.

C. ON-LINE DETECTION PHASE

The proposed algorithm used for the On-line detection phase is depicted in Algorithm 1. In the algorithm, I is a set of CAN IDs from one sliding window W. The remaining parameters are optimized in the Off-line learning phase. During the On-line intrusion detection phase, the in-vehicle network is monitored in real-time in units per sliding window, W.

The details of the On-line detection phase are as follows:

1) In line 1, we define the average of similarity $u_s = 0.8$. This average has been measured by the result of the average similarity of six car models.

Algorithm 1 Similarity-Based Intrusion Detection Algorithm (On-Line Detection Phase)

```
Input: $I \Leftarrow \{message_1, message_2, \ldots, message_W\}, k, \sigma_s, W, CIDs$
1: $u_s \Leftarrow 0.8$
2: while True do
3:     $WIDs \Leftarrow$ extract_CANID($I$)
4:     Calculate similarity $S$ according to overlap($WIDs$, $CIDs$) based on Equation (5)
5:     if $S$ not in normal range $[u_s - k\sigma_s, u_s + k\sigma_s]$ then
6:         Detect DoS attacks
7:     end if
8: end while
```

2) In lines 2-4, we calculate the similarity value in sliding window W.

3) In lines 5-7, we judge whether the similarity value is within the normal range.

As described in Section III-A, the time complexity in calculating similarity is $\mathcal{O}(1)$. Thus, the time complexity of Algorithm 1 is $\mathcal{O}(|W|)$.

D. OFF-LINE LEARNING PHASE

As a parameter for evaluating our similarity-based IDS, we used precision TP/(TP + FP) (True Positive: TP, False Positive: FP). We selected this precision because it gives the main indicators in the IDS. The detection rate of attack messages R_A (TP rate) is calculated according to equation (6).

$$R_A(\%) = \frac{D_A}{T_A} \times 100 \tag{6}$$

where T_A is the total number of DoS attack blocks, D_A is the detected number of DoS attack blocks. And, the detection error rate of attacks R_N (FP rate) is calculated according to equation (7).

$$R_N(\%) = \frac{D_N}{T_N} \times 100 \tag{7}$$

where T_N is the total number of normal message blocks, D_N is the number of normal message blocks detected incorrectly as attacks by the IDS. Also, if the number of normal messages is greater than the number of DoS messages, the block is labeled as normal.

Algorithm 2 Modified Similarity-Based Intrusion Detection Algorithm for Off-Line Learning Phase

```
Input: $Test\_Data$,
    $I \Leftarrow \{message_1, message_2, \ldots, message_W\}, k, \sigma_s, W,$
    $CIDs$
Output: Precision $\frac{R_A}{R_A+R_N}$
$u_s \Leftarrow 0.8$
while $I$ in $Test\_Data$ do
    $WIDs \Leftarrow$ extract_CANID($I$)
    Calculate similarity $S$ according to overlap($WIDs$, $CIDs$) based on Equation (5)
    if $S$ not in normal range $[u_s - k\sigma_s, u_s + k\sigma_s]$ then
        if The window include number of malicious messages greater than number of normal messages then
            $D_A + = 1$
        else
            $D_N + = 1$
        end if
    end if
end while
Calculate TP rate $R_A$ and FP rate $R_N$, based on Equation (6) and and Equation (7)
return Precision $\frac{R_A}{R_A+R_N}$
```

The proposed algorithm used for the Off-line learning phase is depicted in Algorithm 2 and 3. Algorithm 2 is the algorithm added to calculate precision to Algorithm 1 for the Off-line learning phase. The *Test_Data* of input parameters in Algorithm 2 represents the CAN message chronologically sequenced, including the DoS attack blocks. We employ the SA algorithm to optimize parameters in the Algorithm 3. The energy function used in the SA algorithm is as follows.

$$\mathrm{E}() = C_1 \times R_A(\%) - C_2 \times R_N(\%) - C_3 \times W \tag{8}$$

where E() represents the efficiency of the TP rate, the FP rate, and the detection time. E() is based on Equations (6), (7), and sliding window W. Three weighted parameters C_1, C_2, C_3 are used to adjust the weights to the characteristics of IDS. To get high precision and fast detection time, we set $C_1 = 1.0$, $C_2 = 0.5$, $C_3 = 2.0$, which are the same values as in the entropy-based IDS [11] and the sliding window W is in the range of [5, 50].

Algorithm 3 Sliding Windows Optimization Algorithm Using SA (Off-Line Learning Phase)

```
Input: $Learning\_Data\_with\_DoS\_attack$,
    $I \Leftarrow \{message_1, message_2, \ldots, message_W\}$
Output: $(\sigma_s, W)\_set_{max}, CIDs_{max}$
function neighbor($\sigma_s$, $W$)
    return random($\sigma - 0.5, \sigma + 0.5$), random($W - 10, W + 10$)
end function
function probability($e1$, $e2$, $T$)
    return exp $(-(\frac{e2-e1}{T}))$
end function

while $I$ in $Learning\_Data\_with\_DoS\_attack$ do
    $CIDs \Leftarrow$ extract_CANID($I$)
    $k \Leftarrow 0.8$
    $\sigma_s\_best \Leftarrow \sigma_{e0}$
    $W\_best \Leftarrow W_0$
    $e_{best} \Leftarrow$ E($(\sigma_s, W)\_set_0, CIDs$)
    $T \leftarrow 10000$
    $cool \leftarrow 0.99$
    while $T > 0.0001$ and $e_{best} > e$ do
        $(\sigma_s, W)\_set_{next} \Leftarrow$ neighbor($(\sigma_s, W)\_set$)
        $e_{next} \Leftarrow$ E($(\sigma_s, W)\_set_{next}, CIDs$) Calculated by Algorithm_2 ($Learning\_Data\_with\_DoS\_attack$, $k$, $(\sigma_s, W)\_set_{next}$, $CIDs$)
        $p =$ probability($e$, $e_{next}$, $T$)
        if random() $< p$ then
            $(\sigma_s, W)\_set \Leftarrow (\sigma_s, W)\_set_{next}$; $e \Leftarrow e_{next}$
            if $e > e_{best}$ then
                $(\sigma_s, W)\_set_{best} \Leftarrow (\sigma_s, W)\_set$; $e_{best} \Leftarrow e$
            end if
        end if
        $T \Leftarrow T \times cool$
    end while
    $precision_{best}$
    $\Leftarrow$ Algorithm_2 ($Learning\_Data\_with\_DoS\_attack$, $k$, $(\sigma_s, W)\_set_{best}$, $CIDs$)
    if $precision_{max} < precision_{best}$ then
        $precision_{max} \Leftarrow precision_{best}$
        $(\sigma_s, W)\_set_{max} \Leftarrow (\sigma_s, W)\_set_{best}$
        $CIDs_{max} \Leftarrow CIDs$
    end if
end while
return $(\sigma_s, W)\_set_{max}, CIDs_{max}$
```

The *Learning_Data_with_DoS_attack* of input parameters in Algorithm 3 represents the CAN messages of time sequence, including the DoS attack blocks. The I is a set of one sliding window W. Algorithm 3 optimizes σ_s, W, and *CIDs* to achieve high precision and fast detection. $(\sigma_s, W)\_set$ is randomly generated, where σ_s is the deviation, k is the sensitivity deviation, and u_s is the average similarity value. The sensitivity deviation k is 0.8 the same as the average similarity value. The purpose of Algorithm 3 is to obtain

TABLE 1. Data set description.

Name	Normal messages	H_{ave}	H_{dev}
HCR Lab [25]	1000	3.05408	0.08345
Car model A	1000	3.28269	0.06991
Car model B	1000	2.09613	0.04416
Car model C	1000	3.25292	0.07083
Car model D	1000	3.41602	0.11795
Car model E	1000	2.88480	0.12877

the parameter settings that can effectively maximize E(). The details of the Off-line learning phase are as follows:

1) In lines 1-6, functions neighbor() and probability() are defined and later used in lines 17 and 19 respectively.
2) In lines 8-27, we execute the SA algorithm to optimize σ_s, W.
3) In lines 28-33, we calculate $precision_{best}$ using $(\sigma_s, W)\_set_{best}$, and then compare $precision_{best}$ with $precision_{max}$, thereby getting the parameters $(\sigma_s, W)\_set_{best}$ with the highest precision among all *CIDs*.

Since the time complexity of Algorithm 1 is $\mathcal{O}(|W|)$, the time complexity of Algorithm 3 is $\mathcal{O}(|S| \times |T| \times |W|)$, where $|T|$ is the temperature in the SA algorithm, $|S|$ is the number of normal blocks in *Learning_Data_with_DoS_attack*.

IV. EXPERIMENTS AND EVALUATIONS

A. DATA SET

In this study, we evaluate the precision in detecting the three types of DoS attacks. We use a data set of the real CAN messages provided in [25] and a data set of the five car models (A, B, C, D, E) which we logged during driving and stopping. We evaluate the similarity-based IDS using the DoS attack messages (1000 messages) added to the data sets without DoS attack messages.

Table 1 shows the dataset description. We describe the average value H_{ave} and the standard deviation H_{dev} of the entropy at the sliding window $W = 60$, which was regarded as the optimal parameter in the entropy-based IDS [11]. As shown in Table 1, the entropies depend on the car model. Therefore, when these normal entropies and the entropies of Random DoS attacks are the same value, the precision of the entropy-based IDS decreases.

We evaluate the precision and detection time of similarity-based IDS with the three aspects. First, the precision of similarity-based IDS is compared from the precision of entropy-based IDS against three types of DoS attacks. Second, we evaluate the precision of similarity-based IDS against entropy-manipulated attacks. Finally, we conduct the experiment, in which the similarity-based IDS detect DoS attacks, to measure the detection time in resource-restricted environments.

B. PRECISION EVALUATION AGAINST THREE TYPES OF DOS ATTACKS

In this section, we evaluate the precision of the similarity-based IDS against each of the DoS attacks. In actual automobiles, the FP rate in the IDS should be low. In other words, the similarity-based IDS is expected to have a high TP rate and a low FP rate. Therefore, we selected a TP rate, an FP rate, and the precision (TP/(TP + FP)) as the evaluation indicators of the similarity-based IDS. Table 2 shows the evaluation indicators of the entropy-based IDS and the similarity-based IDS against Traditional, Random, and Targeted DoS attacks. Also, we evaluate the entropy-based IDS and the similarity-based IDS using only the HCR Lab data set.[1] Table 2 only shows the precision against the Random DoS attack with the range of [0, 31] in both methods, because this range is suitable as the example of entropy-manipulated attacks.

Table 2 shows that the entropy-based IDS can detect DoS attacks in which an adversary uses a single CAN ID at the high TP rate and high precision. However, the precision of the entropy-based IDS against Random DoS attacks is 68.3%. Also, the average and standard deviation of entropies of Random DoS attacks are $H_{ave} = 3.08535$, $H_{dev} = 0.07863$, and these are almost the same as the average and standard deviation of the HCR Lab's data set in Table 1. Hence, we confirm that the entropy-based IDS cannot correctly classify a Random DoS attack.

On the other hand, Table 2 shows that the similarity-based IDS can detect all DoS attacks with a high TP rate and precision. Therefore, it is confirmed that the similarity-based IDS has superior precision against Random DoS attacks, as compared with the entropy-based IDS, and has the same precision as other methods for the other types of DoS attacks. Also, this evaluation made it clear that the similarity-based IDS with the below parameters can detect all DoS attacks in the HCR Lab's data set.

$$
\begin{aligned}
W &= 25,\\
\sigma_s &= 0.52499,\\
CIDs &= \{0\times 80, 0\times 80, 0\times 81, 0\times 81, 0\times 153, 0\times 164,\\
&\quad 0\times 165, 0\times 165, 0\times 18\mathrm{f}, 0\times 18\mathrm{f}, 0\times 220,\\
&\quad 0\times 260, 0\times 2\mathrm{a}0, 0\times 2\mathrm{b}0, 0\times 316, 0\times 316,\\
&\quad 0\times 329, 0\times 370, 0\times 382, 0\times 43\mathrm{f}, 0\times 440,\\
&\quad 0\times 4\mathrm{b}0, 0\times 4\mathrm{b}1, 0\times 545, 0\times 5\mathrm{a}2\}
\end{aligned}
$$

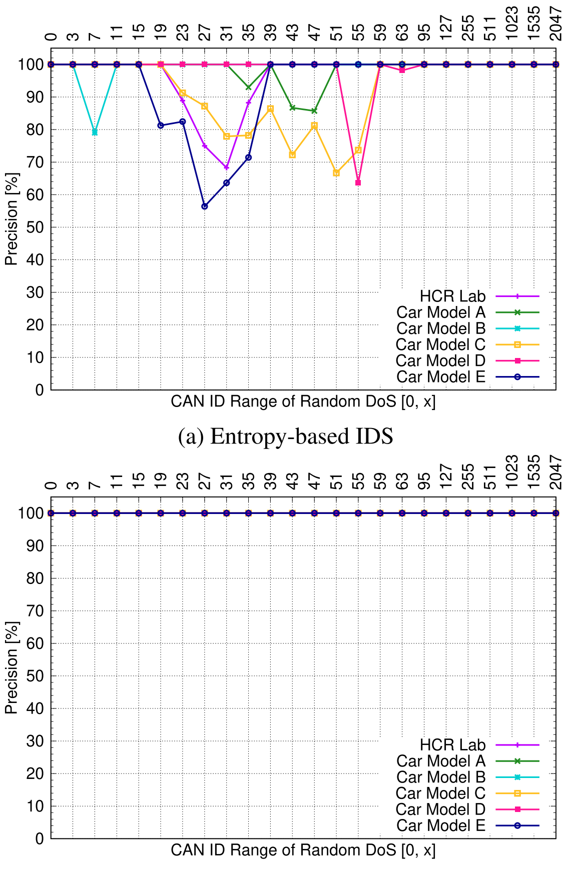
C. PRECISION EVALUATION IN VARIOUS RANGES OF CAN ID

In this section, we compare the similarity-based IDS with the entropy-based IDS when these IDSs are used under an entropy-manipulated attack. We also compare the two, in Figure 9, for their precision against entropy-manipulated attacks. We confirm that the entropy-based IDS has a range in which the precision decreases, whereas the similarity-based IDS can detect all ranges. Figure 9 shows the precision against entropy-manipulated attacks of various ranges,

[1] We must hide specific CAN IDs of real vehicle data except the data set of HCR Lab because they are not in public from the companies.

TABLE 2. Comparison of the precision at each DoS attack.

	Entropy-based IDS [11]			Similarity-based IDS		
	R_A[%]	R_N[%]	Precision[%]	R_A[%]	R_N[%]	Precision[%]
Traditional DoS	100.0	0.0	100.0	100.0	0.0	100.0
Random DoS[0, 31]	94.6	44.0	68.3	100.0	0.0	100.0
Targeted DoS	100.0	0.0	100.0	100.0	0.0	100.0



(a) Entropy-based IDS

(b) Similarity-based IDS

FIGURE 9. Comparison of precision against entropy-manipulated attack.

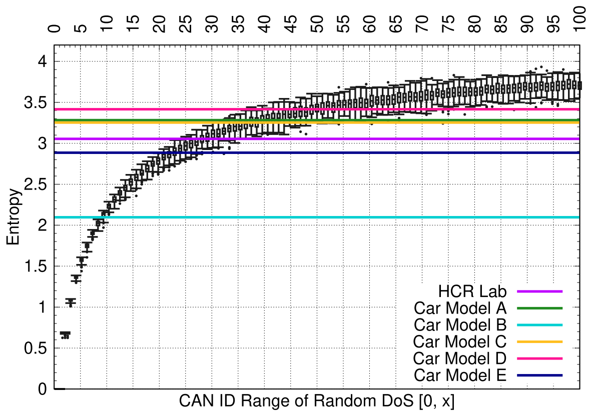


FIGURE 10. Entropy of entropy-manipulated attack under different CAN ID range [0, 0]-[0, 100].

TABLE 3. The experimental environment.

CPU	Broadcom BCM 2837 1.2GHz 64bit quad-core armv7l
RAM	1GB
OS	Debian 8.0
CAN Interface	PiCAN 2

while Table 2 shows the precision against only the entropy-manipulated attack with the range of [0, 31] in both methods.

We show the variation of the entropy in the entropy-manipulated attack under different CAN ID ranges in Figure 10, which shows a correspondence between the entropy and the CAN ID range of entropy-manipulated attacks. Figure 10 plots the 600 entropies of entropy-manipulated attacks in each range in the box-and-whisker plots. Furthermore, we depict the lines which express the average of entropies on the six car models with Figure 10. Focusing on the car model D in Figure 9 (a), the precision of the entropy-based IDS has decreased in the range [0, 55]. Next, when focusing on the x-axis [0, 55] in Figure 10, the average entropy of car model D is within the entropy of the entropy-manipulated attack in the range [0, 55]. Because the entropy of the entropy-manipulated attack and the average normal entropy are the same value, the entropy-based IDS cannot detect the entropy-manipulated attack with high precision. The same applies to other car models. Hence, we confirmed that the precision decreases when the average entropy used in detecting DoS attacks is within the range of the entropy of the entropy-manipulated attack.

D. DETECTION TIME EVALUATION

In this section, we compare the detection time of the similarity-based IDS and of the entropy-based IDS in the On-line detection phase. Also, since the Off-line learning phase has nothing to do with real-time detection, we evaluate only the On-line detection phase.

First, we describe the experimental environment (see Table 3). We assumed Raspberry Pi, a low-spec evaluation board, to implement our similarity-based IDS on resource-restricted on-board computers. We also implemented the entropy-based IDS [11] for the comparison between entropy- and similarity-based IDS. Thus, the conventional one was conducted in the same environments of similarity-based IDS for this evaluation.

Next, we define the evaluation time in Figure 11. The evaluation time T_1 shows the time after the start of the DoS attack until the anomaly is detected. In other words, T_1 is the time that increases in proportion to the sliding windows. The evaluation time T_2 shows the time from receiving W messages as a sliding window until the time of detecting the

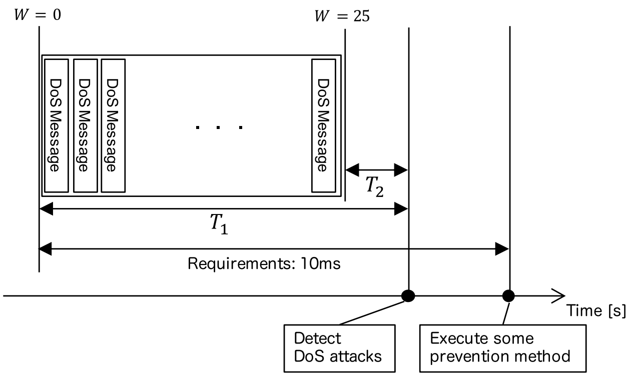
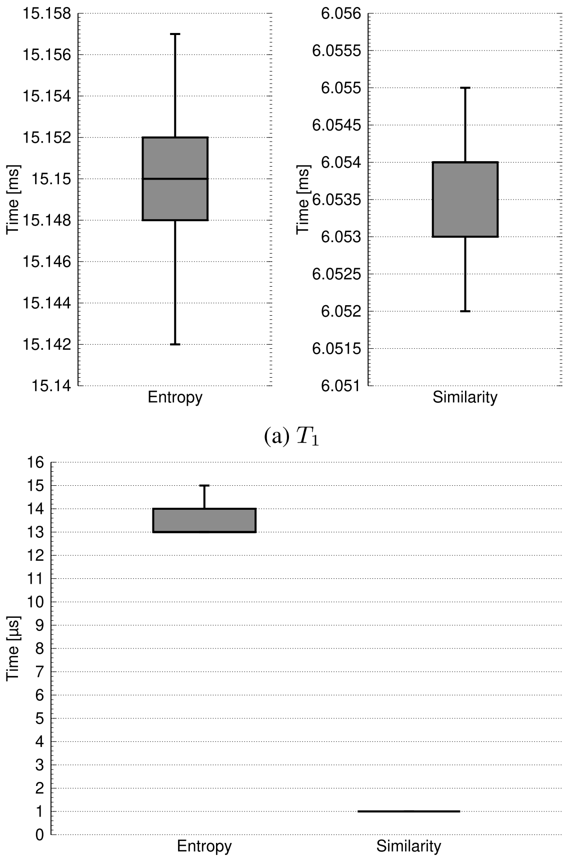


FIGURE 11. Definition and requirements for detection time in CAN.



(a) T_1

(b) T_2

FIGURE 12. Comparison of detection time.

anomaly. In other words, T_2 is an indicator to compare the calculation times of the entropy and the similarity.

We show the actual requirement in the detection time of similarity-based IDS. In some cars, it is impossible to send messages more often than 10 ms apart due to the load requirements placed by vehicle manufacturers [26]. Therefore, if we can prevent the DoS attacks until 10 ms after starting attacks, the messages may be sent with the correct intervals. Hence, we define a requirement as which the T_1 must be shorter than 10 ms in the similarity-based IDS.

Figure 12 shows the box-and-whisker diagrams as the results of T_1 and T_2 measured 1000 times each. Also, the median of T_1 is 6.054 ms in the similarity-based IDS. In the figure, the detection times T_1 of the similarity-based IDS are lower by one order to the detection times of the entropy-based IDS. As shown in Figure 12 (a), the T_1 ranges of the similarity-based IDS and of the entropy-based IDS are 6.052-6.055 ms and 15.142-15.157 ms, respectively. This result is caused by the difference that is an optimized sliding window $W = 25$ in the similarity-based IDS, whereas an optimized sliding window $W = 60$ in the entropy-based IDS.

As shown in Figure 12 (b), the T_2 ranges of the similarity-based IDS and of the entropy-based IDS are 13-15 μs and 1 μs, respectively. The median of T_2 is 14 μs in the entropy-based IDS. We confirmed that the similarity-based IDS could detect an attack up to 93.33% (14 μs) faster than the entropy-based IDS.

V. DISCUSSION

In Section IV, the similarity-based IDS can detect all DoS attacks in 100.0% precision and with a faster time than the conventional entropy-based IDS by up to 93.33% (14 μs). We discuss these results in this section.

A. PRECISION

We found this DoS attack called the entropy-manipulated attack, which bypasses the conventional entropy-based IDS by adjusting the entropy of messages. The proposed similarity-based IDS can detect the entropy-manipulated attack because it can distinguish between the CAN IDs of the entropy-manipulated attack and the normal CAN IDs. As an experimental result, the similarity-based IDS achieved a detection precision of 100.0% against the entropy-manipulated attacks, while the detection precision is 68.3% in the entropy-based IDS. Since an adversary use CAN IDs with higher priority than normal messages in the entropy-manipulated attacks, the similarity decreases between normal messages and the DoS attack. Therefore, as shown in Figure 9(b), the similarity-based IDS can detect entropy-manipulated attacks with 100.0% precision.

However, the similarity-based IDS may not be able to detect a Replay DoS attack which is a combination of replay attacks and DoS attacks, because an adversary can inject the same CAN IDs with normal CAN messages. Since the entropy-based IDS is effective against a Replay attack, a hybrid implementation of the similarity-based IDS and the entropy-based IDS would be effective against Replay DoS attacks.

B. DETECTION TIME

Due to load requirements placed by the vehicle manufacturers [26], we defined the requirement as which the T_1 must be shorter than 10 ms in the similarity-based IDS. As the result of Section IV-D, the T_1 ranges of the similarity-based IDS and of the entropy-based IDS are 6.052-6.055 ms and 15.142-15.157 ms, respectively. The T_1 of the similarity-based IDS meets the 10 ms of the requirement, whereas the entropy-based IDS's T_1 does not meet the requirement. Thus, we confirmed that the similarity-based

TABLE 4. Comparison of the related works.

	Rule-based IDS	Time-interval IDS [9]	Entropy-based IDS [11]	Similarity-based IDS
Traditional DoS	100%	100%	100%	100%
Random DoS	100%	100%	68.3%	100%
Targeted DoS	0%	0%	100%	100%
Different Bandwidth	Applicable	Not Applicable	Applicable	Applicable
Threshold	-	Not Optimized	Optimized	Optimized
Time Complexity	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\lvert W\rvert \times \log(N))$	$\mathcal{O}(\lvert W\rvert)$

IDS is superior to the entropy-based IDS in actual requirements.

As shown in Figure 12 (b), the T_2 ranges of the similarity-based IDS and of the entropy-based IDS are 13-15 μs and 1 μs, respectively. Note that the time complexity of the entropy-based IDS is $\mathcal{O}(|W| \times \log(|N|))$, and the computational complexity of the similarity-based IDS is $\mathcal{O}(|W|)$ in the On-line detection phase. Hence, we showed that the detection time is up to 93% (14 μs) shorter than with the entropy-based IDS in Section IV-D.

Incidentally, in ID-Hopping Mechanism [6]–[8] which avoids Targeted DoS attacks, the average overhead required for AES encryption to generate a one-time ID and for newly setting the CAN ID register are 20.23 μs and 0.2 μs respectively. Therefore, the impact of achieving rapid IDS 14 μs faster than the entropy-based IDS is worth to cancel the overhead for utilizing the conventional IDS and the ID-Hopping Mechanism together.

C. COMPARISONS

Some IDSs proposed so far has a good advantage in term of effectiveness to DoS attacks and the small computational overhead. In the following, these IDSs and the similarity-based IDS are compared. Table 4 shows a comparison of the related works. Incidentally, to compare the similarity-based IDS and various methods, we newly define one of the IDS called Rule-based IDS which detects an attacker based on a white-list or black-list of CAN ID.

First, we compare each IDSs based on three types of DoS attacks. Rule-based IDS can detect Traditional and Random DoS attacks using a white-list or black-list because Traditional DoS attacks consisted of messages of CAN ID 0 × 000. However, Targeted DoS attacks are bypassed because rule-based IDS judges attack with the white-list. Time-interval based IDS can detect Traditional DoS attacks because time-interval based IDS detects anomaly interval of messages of CAN ID 0 × 000. When Random DoS attacks have messages of same CAN IDs assigned to the CAN, the time-interval of the CAN IDs is shorter than the regular time-interval. Hence, the time-intervals IDS detects Random DoS attacks. However, the time-interval IDS cannot detect Targeted DoS attacks using a non-cyclic CAN ID. The time-interval IDS monitors the CAN IDs which ECUs send periodically. In other words, the time-interval IDS does not monitor non-cyclic CAN IDs. As we confirmed in Section IV-B, the entropy-based IDS can detect Traditional and Targeted DoS attacks. However, the entropy-based IDS has a problem that the FP rate is high against Random DoS attacks. While our similarity-based IDS using the similarity of sliding windows has a high precision against both Random DoS attacks and the other DoS attacks. As we mentioned in Section II-B.2, the entropy-based IDS bypasses entropy-manipulated attacks which is a kind of Random DoS attack, whereas the similarity-based IDS can detect all DoS attacks. From the comparison above, we confirmed that the similarity-based IDS could only detect DoS attacks of all types.

Secondly, we describe the applicability in different bandwidth of CAN. The rule-based IDS can be used in different bandwidth of CAN because this IDS does not use intervals of messages to detect attacks. Similar to the rule-based IDS, the entropy-based IDS and the similarity-based IDS are applicable because the entropy and the similarity are calculated based on CAN IDs of a fixed number of messages. On the other hand, since the time-interval of messages varies in each bandwidth, time-interval IDS cannot be used in different bandwidth of CAN. Thus, we confirmed that the rule-, entropy-, and similarity-based IDSs have advantages in terms of different bandwidth.

Thirdly, we mention whether each IDSs optimize a threshold to judge DoS attacks. Also, the rule-based IDS detects the attacks based on a white-list or black-list of CAN ID, so that this IDS does not have a threshold to judge attacks. The time-interval IDS has a threshold to detect DoS attacks with high accuracy. An expert must manually select this threshold before the IDS is implemented on the actual CAN bus. In addition, the threshold is experimental rather than theoretical; it is possible that there is an optimized threshold to detect DoS attacks. Both the entropy- and the similarity-based IDSs automatically optimize a threshold to judge attacks using SA. Therefore, there is an advantage to determine a threshold in the rule-, entropy-, and similarity-based IDSs.

Finally, we discuss the time complexity. The rule-based IDS uses a white-list or a black-list to detect intrusions so that the time complexity of this IDS is $\mathcal{O}(1)$. The time-interval IDS calculates the time-interval when received some messages. Since this IDS only needs calculating the time-interval and comparing whether the time-interval is normal to detect attacks, the time complexity is $\mathcal{O}(1)$. Next, the time complexity of the entropy-based IDS's On-line detection phase is $\mathcal{O}(|W| \times \log(N))$. As mentioned in Section III-C, the time complexity of the similarity-based IDS is $\mathcal{O}(|W|)$. Thus, the rule-based IDS and Time-interval IDS have advantages in terms of the time complexity. On the other hand, we evaluated the actual detection time of the similarity-based

IDS. As a result, we confirmed that our method satisfies the requirement from the vehicle manufacturers [26]. Therefore, we conclude that the similarity-based IDS can be operated in the actual environment.

From these comparisons among related works, we confirm that the similarity-based IDS can detect the DoS attacks of all types. In addition, it was confirmed that the similarity-based IDS has advantages in terms of applicability in CAN of different bandwidth, determining the threshold, and the detection time.

VI. CONCLUSION

The growing number of vehicles connected to the internet causes a security risk of cyberattacks such as DoS attacks on modern automobiles. It requires a security solution that can prevent DoS attacks. To prevent all DoS attacks, firstly we must consider a method to detect all DoS attacks. In this research, we proposed an optimized DoS attack detection method based on the similarity of sliding windows that is capable of detecting every type of DoS attack. In addition, we have solved the entropy-based IDS' problem of a higher FP rate occurring when the entropy-manipulated attack is executed. Furthermore, our similarity-based IDS has lower computational complexity than the entropy-based IDS. We confirmed that the similarity-based IDS detected a DoS attack in 100% of the cases in our experiment, and we showed that the detection time is up to 93.33% (14 μs) shorter than with the entropy-based IDS. We release the source code [27] hoping to promote research on countermeasures against DoS attacks.

For future work, we plan to implement a hybrid method based on similarity and entropy against Replay DoS attacks. Furthermore, We will apply our method to the case of a specific ECU message being interrupted by a DoS attack (*bus-off attack*) [28] which abuses the error handling protocol of CAN. Finally, since we proposed the similarity-based IDS which can rapidly detect all DoS attacks, we will evaluate the serviceability of using the similarity-based IDS and the ID-Hopping mechanism together.

REFERENCES

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, pp. 1–91, Aug. 2015.

[2] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to CAN bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[3] R. B. GmbH. (Jul. 2019). *CAN Specification Version 2.0*. Accessed: Aug. 7, 2019. [Online]. Available: http://esd.cs.ucr.edu/webres/can20.pdf

[4] A. G. Linux. (Jul. 2019). *Automotive Grade Linux*. Accessed: Aug. 7, 2019. [Online]. Available: https://www.automotivelinux.org/

[5] D. Oka and C. Gay, "Open-source software in your car—What can go wrong?" in *Proc. SCIS*, 2019, pp. 1–8.

[6] A. Humayed and B. Luo, "Using ID-hopping to defend against targeted DoS on CAN," in *Proc. 1st Int. Workshop Safe Control Connected Auton. Vehicles (SCAV)*, 2017, pp. 19–26.

[7] W. Wu, R. Kurachi, G. Zeng, Y. Matsubara, H. Takada, R. Li, and K. Li, "IDH-CAN: A hardware-based ID hopping CAN mechanism with enhanced security for automotive real-time applications," *IEEE Access*, vol. 6, pp. 54607–54623, 2018.

[8] S. Woo, D. Moon, T.-Y. Youn, Y. Lee, and Y. Kim, "CAN ID shuffling technique (CIST): Moving target defense strategy for protecting in-vehicle CAN," *IEEE Access*, vol. 7, pp. 15521–15536, 2019.

[9] 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 *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2016, pp. 63–68.

[10] B. Groza and P.-S. Murvay, "Efficient intrusion detection with Bloom filtering in controller area networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 1037–1051, Apr. 2019.

[11] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks," *IEEE Access*, vol. 6, pp. 45233–45245, 2018.

[12] R. Kurachi, H. Takada, N. Adachi, H. Ueda, and Y. Miyashita, "DDCAN: Delay-time deliverable CAN network," in *Proc. IEEE 19th Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Jul. 2019, pp. 36–41.

[13] M. S. Idrees, H. Schweppe, Y. Roudier, M. Wolf, D. Scheuermann, and O. Henniger, "Secure automotive on-board protocols: A case of over-the-air firmware updates," in *Proc. Int. Workshop Commun. Technol. Vehicles*. Oberpfaffenhofen, Germany: Springer, Mar. 2011, pp. 224–238.

[14] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, to be published.

[15] R. B. GmbH. (Jul. 2019). *CAN with Flexible Data-Rate Specification Version 1.0*. Accessed: Jul. 8, 2019. [Online]. Available: https://can-newsletter.org/assets/files/ttmedia/raw/e5740b7b5781b8960f5%5efcc2b93edf8.pdf

[16] Github. (Aug. 2019). *Linux-Can/Can-Utils: Linux-CAN/SocketCAN User Space Applications*. Accessed: Jul. 8, 2019. [Online]. Available: https://github.com/linux-can/can-utils

[17] J. Cook and J. Freudenberg, "Controller area network (CAN)," *EECS*, vol. 461, pp. 1–5, Oct. 2007.

[18] 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 *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, 2019, pp. 229–244.

[19] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0155781.

[20] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–6.

[21] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 787–800.

[22] K.-T. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2017, pp. 1109–1123.

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

[24] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Proc. IEEE 2nd Int. Forum Res. Technol. Soc. Ind. Leveraging Better Tomorrow (RTSI)*, Sep. 2016, pp. 1–6.

[25] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2017, pp. 57–5709.

[26] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Syst.*, vol. 35, no. 3, pp. 239–272, Jan. 2007.

[27] S. Ohira. (Jul. 2019). *Similarity-Based IDS*. Accessed: Jul. 8, 2019. [Online]. Available: https://github.com/shuji-oh/similarity_CAN_IDS

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

SHUJI OHIRA (Student Member, IEEE) received the B.E. degree in information sciences from Hiroshima City University, in 2018. He is currently pursuing the master's degree with the Graduate School of Science and Technology, Nara Institute of Science and Technology. His research interests include embedded system security and cyber-physical systems security.

ARAYA KIBROM DESTA received the M.E. degree from the Graduate School of Information Science, Nara Institute of Science and Technology, in 2019, where he is currently pursuing the Ph.D. degree. His research interests include intrusion detection systems and cyber-physical systems security.

ISMAIL ARAI (Member, IEEE) received the M.E. and Ph.D. degrees in engineering from the Graduate School of Information Science, Nara Institute of Science and Technology, in 2004 and 2008, respectively. From 2008 to 2011, he was a Postdoctoral Fellow with the Research Organization of Science and Engineering, Ritsumeikan University. He was an Assistant Professor, a Lecturer, and an Associate Professor with the Department of Electrical and Computer Engineering, National Institute of Technology, Akashi College, from 2011 to 2013, from 2013 to 2015, from 2015 to 2016, respectively. He has been an Associate Professor affiliated with the Information Initiative Center, Nara Institute of Science and Technology, since 2016. His research topics include transportation data analysis, pedestrian navigation systems, cybersecurity, and open data. He is also a member of the ACM, IPSJ, and IEICE.

HIROYUKI INOUE (Member, IEEE) received the Ph.D. degree in engineering from the Nara Institute of Science and Technology, in 2000. He has been an Associate Professor with the Graduate School of Information Science, Hiroshima City University, since 2010. His research interests include technologies for embedded security, especially automotive network security, and network protocol of the Internet.

KAZUTOSHI FUJIKAWA (Member, IEEE) received the M.E. and Ph.D. degrees in information and computer sciences from Osaka University, in 1990 and 1993, respectively. He is currently a Professor with the Information Initiative Center, Nara Institute of Science and Technology. His research focuses on multimedia communication systems, digital libraries, ubiquitous computing, and mobile networks. He is a member of ACM and IPSJ.