IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 25, 2018, accepted August 2, 2018, date of publication August 13, 2018, date of current version September 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2865169

Sliding Window Optimized Information Entropy Analysis Method for Intrusion Detection on In-Vehicle Networks

WUFEI WU[1], (Student Member, IEEE), YIZHI HUANG[1], RYO KURACHI[2], (Member, IEEE), GANG ZENG[2,3], (Member, IEEE), GUOQI XIE[1], (Member, IEEE), RENFA LI[1], (Senior Member, IEEE), AND KEQIN LI[1,4], (Fellow, IEEE)

[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China
[2]Graduate School of Informatics, Nagoya University, Nagoya 464-8601, Japan
[3]Graduate School of Engineering, Nagoya University, Nagoya 464-8601, Japan
[4]Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

Corresponding author: Renfa Li (lirenfa@hnu.edu.cn)

This work was supported in part by the Natural Science Foundation of Hunan Province under Grant 2018JJ3076, in part by the CCF-Venustech Open Research Fund under Grant CCF-VenustechRP2017012, in part by the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China under Grant ICT1800399, and in part by the National Natural Science Foundation of China under Grant 61702172, Grant 61672217, and Grant 61502162.

ABSTRACT With the considerable growth of cybersecurity risks in modern automobiles, cybersecurity issues in the in-vehicle network environment have attracted significant attention from security researchers in recent years. Enhancing the cybersecurity ability of in-vehicle networks while considering the computing resource and cost constraints become an urgent issue. To address this problem, a novel information entropy-based method is proposed in this paper, which uses a fixed number of messages as sliding windows. By improving the sliding window strategy and optimizing the decision conditions, the detection accuracy is increased and the false positive rate is reduced. Experimental results demonstrate that the proposed method can provide real-time response to attacks with a considerably improved detection precision for intrusion detection in the in-vehicle network environment.

INDEX TERMS Controller area network (CAN), cybersecurity, information entropy, in-vehicle network, intrusion detection system (IDS).

I. INTRODUCTION

A. BACKGROUND

Automotive electronic systems are distributed systems that consist of various electronic devices such as electronic control units (ECUs), actors (e.g., engine, motor, brakes, and steering wheel), and sensors (e.g., LIDAR, GPS, position estimator, RADAR, and cameras), which are interconnected with an in-vehicle network such as controller area network (CAN), LIN, and FlexRay. Currently, the number of deployed ECUs in a modern high-end automobile is more than 100 to implement various functionalities [1]. With the continuous improvement in the complexity and connectivity of modern vehicles, the cyber security risks of automobiles have become increasingly prominent [2], [3]. Nevertheless, the vehicle has existed as an independent system for a long time, and existing in-vehicle networks do not provide message encryption and authentication mechanisms during the protocol design phase.

Currently, cyber security attacks in vehicles have increased and have been reported in several papers [2], [4], [5]. For example, Koscher et al. in [4] successfully controlled a wide range of automotive functions, such as disabling brakes and stopping the engine. This attack led to a recall of 1.4 million cars by manufacturers. Attacks on vehicles usually include a series of operations, such as sniffing and fuzzing of CAN bus and reverse engineering the firmware of ECU. Given that an automotive electronic system is safety-critical, its design needs to strictly follow the desired standard specifications, such as ISO 26262 [6]. The cyber security threat to safety function systems is threat to not only information security and privacy but also the user's life and property safety. To avoid serious damage caused by hacking, the intrusion detection capabilities of the in-vehicle networks should be improved.

Countermeasures have been proposed to enhance security performance of in-vehicle networks [7]–[9]. However, security enhancement methods such as message encryption and authentication are limited by cost and resource [10]–[13]. The ever-evolving attack techniques are difficult to block with firewalls [14] because network attacks are becoming increasingly diverse and intelligent during automobiles' life cycle (generally more than 20 years). In order to prevent reverse engineering from attackers, existing protection technologies cannot deal with these attacks effectively. Therefore, intrusion detection is currently considered a good option for enhancing in-vehicle network cyber security owing to its cost effectiveness and high compatibility. In this study, a sliding window optimized information entropy analysis method for in-vehicle network's intrusion detection is proposed. This means a strategy that uses the optimal fixed number of messages as the sliding window.

B. MOTIVATION

Recently, several studies to detect attacks targeted on vehicles are proposed in [15]–[18]. The recently released SAE J3061 guidebook for cyber-physical vehicle systems focuses on designing cyber security aware systems in close relation to the automotive safety standard ISO 26262. The studies related to the current work mainly include [19] and [20].

Previous information entropy-based studies have the following shortcomings. First, the monitoring strategy with fixed time as the sliding window is used in [19] and [20]. Because the CAN bus is an event-triggered in-vehicle network. The information entropy under this monitoring strategy cannot avoid the huge impact of the non-periodic CAN messages and different transmission rates (e.g., 125 and 250 Kbps), thereby resulting in high false positive rate. Second, the size of the sampling window greatly influences the information entropy, but how to determine the size of the sampling window has not been carefully studied in the previous studies. To improve the detection accuracy of the intrusion detection system (IDS) for the in-vehicle network environment, an intrusion detection method with high accuracy and low response time is needed.

C. MAIN CONTRIBUTIONS

To the best of our knowledge, compared with the existing information entropy-based intrusion detection system design, this study is the first to use an observation strategy that utilizes the fixed number of messages as the sliding window. The proposed intrusion detection method does not need to change the existing CAN protocol. Therefore, it can be compatible to existing vehicles on the market. The main contributions of this study can be summarized as follows.

1) We propose a fixed number of messages based sliding window strategy to avoid information entropy interference caused by different baud rate and aperiodic CAN messages.

2) We propose a heuristic algorithm based on simulated annealing. According to this algorithm and the real-life in-vehicle network communication data set [21], we can get the best sliding window parameters for intrusion detection system design in this kind of vehicle.

3) We conduct evaluation experiments based on real-life vehicle data sets, and the results demonstrate that the proposed method compared with existing entropy-based method, which can improve the accuracy of intrusion detection with low response time and false positive rate.

The rest of the paper is organized as follows. In Section II, we introduce the background knowledge and attack model and scenarios. Section III presents our proposed detection mechanism. In Section IV, we describe the proposed algorithms in detail. In Section V, we demonstrate the experiments and evaluation results. Section VI presents the related works. Finally, we elaborate the conclusions in Section VII.

II. PRELIMINARIES

A. CAN OVERVIEW

CAN protocol was invented by Robert Bosch GmbH to meet the specific requirements of in-vehicle environment, such as real-time processing, strong robustness, and cost effectiveness. CAN is a commonly used communication network in current in-vehicle environment, which is a priority-based non-preemptive communication network with a maximum data rate of 1 Mbit/s. At the data link layer, CAN protocol uses broadcast communication to transmit messages. Figure 1 shows the basic data frame structure of CAN 2.0. The blue line in the figure indicates transmission waveform on the CAN physical bus. When Node 3 obtains the bus transmission permission, the other nodes will stop sending and wait until the network is idle again. Once the CAN network is illegally accessed, this arbitration mechanism will be vulnerable to high-priority denial-of-service (DoS) attacks.

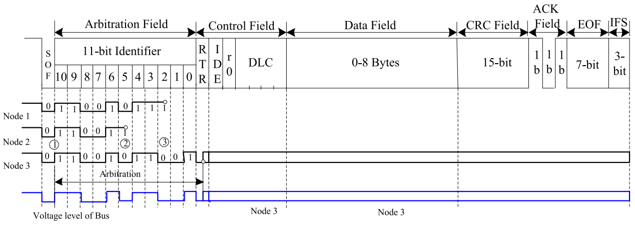


FIGURE 1. Standard CAN message frame.

CAN is a *de facto* standard communication network protocol used in automobiles and industry equipment for 30 years, which is often connected by multiple safety-critical systems. Nevertheless, this component is vulnerable to attack due to its lack of security mechanism design. Therefore, this study focuses on the intrusion detection technology suitable for CAN environment.

B. CAN VULNERABILITY

According to the characteristics of the CAN protocol, the vulnerability of CAN can be described to the next four points [22].

- **Broadcast Transmission:** On the data link layer, the CAN protocol uses broadcast communication to transmit messages, thereby suggesting that all nodes on the bus can receive the messages simultaneously.

- **No Authentication:** Message transmission on CAN bus lacks a design authentication mechanism. The attacker can compromise one or multiple ECUs and restrict them from sending few or all messages.
- **No Encryption:** CAN message lacks an encryption mechanism, and adding encryption technology for CAN will cause additional delay and will reduce the effective utilization rate of network due to the constraints of bandwidth and payload resources.
- **ID-based Priority Scheme:** The arbitration mechanism of CAN is that the message with high priority can continue to transmit data without affecting the bus collision decision time, whereas the low priority message must wait for the next idle state. Therefore, hackers can easily use high-priority IDs to launch DoS attacks.

Figure 2 shows the vulnerabilities of in-vehicle network, the attacks adopted by researchers, and the corresponding countermeasures [7]–[9], [15]–[18]. As shown in Figure 2, an effective countermeasure for frame injection and DoS attacks is to develop an intrusion detection system based on anomaly detection, such as the method proposed in this study. It should be pointed out that the security enhancement system's countermeasure against DoS attacks is limited, especially for high-priority DoS attacks. This means that detection of DoS attacks is easy, but preventing DoS attacks is very difficult.

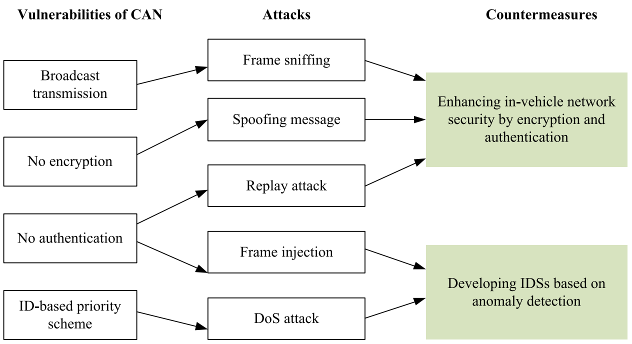


FIGURE 2. In-Vehicle network attacks and corresponding countermeasures.

C. ATTACK HYPOTHESES

1) THREAT MODEL

In our threat model, we assume that attackers can access the CAN bus. Access points include but are not limited to Bluetooth, OBD_II, Wi-Fi, physical access, and USB ports. Once an attacker gains access to the CAN, the attacker can conduct sniffing, spoofing, and DoS attacks. In this study, two attack types are detected in in-vehicle networks. One is DoS attack, and the other is injection attack.

2) ATTACK SCENARIOS

We present two types of attack scenarios based on the threat model. In Figure 3 and Figure 4, the white-box is illustrated as the legitimate messages from the legitimate ECUs. The red-box is illustrated as the spoofing messages from an attacker. Then, the X-axis is the time of message transmission.

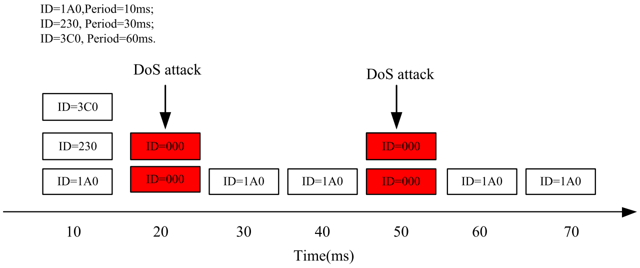


FIGURE 3. Diagram of the DoS attack on the CAN.

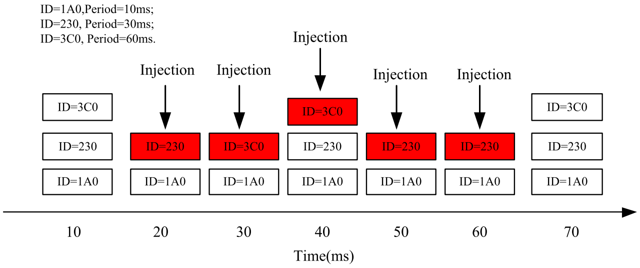


FIGURE 4. Diagram of the injection attack on the CAN.

DoS Attack: As shown in Figure 3, an attacker can inject high-priority messages in a short cycle on the bus. DoS attack against CAN bus can be divided into two categories. One is to corrupt legitimatize messages by an error-frame sending from attackers (such as jamming). The CAN controller can filter out messages that are not received by the ECU because it has a filter and a sub-net mask register. Therefore, sending a specific ID attack to a specific ECU can disable a certain function. Another method is the higher priority messages such as CAN-ID equals to zero. CAN is a broadcast-based bus network based on priority arbitration. Thus the launch of the above-mentioned type of DoS will cause direct CAN bus corruption.

Injection Attack: As shown in Figure 4, an injection attack indicates that the attacker re-sends the latest received message from the node. This type of attack can cause the abnormal high utilization rate of CAN bus to drop back contents of the target signals by injection attacks. The change in the contents of the data segment can invalidate the previously sent messages. The types of injection attacks for CAN networks are divided into two cases. One is the replay attack, which means that the attacker saves the legitimate messages in advance and injects this messages. Another one is the fabrication of the messages in run-time systems.

III. METHODOLOGIES

To improve the detection performance and effectiveness of intrusion detection system design. In this study, we mainly

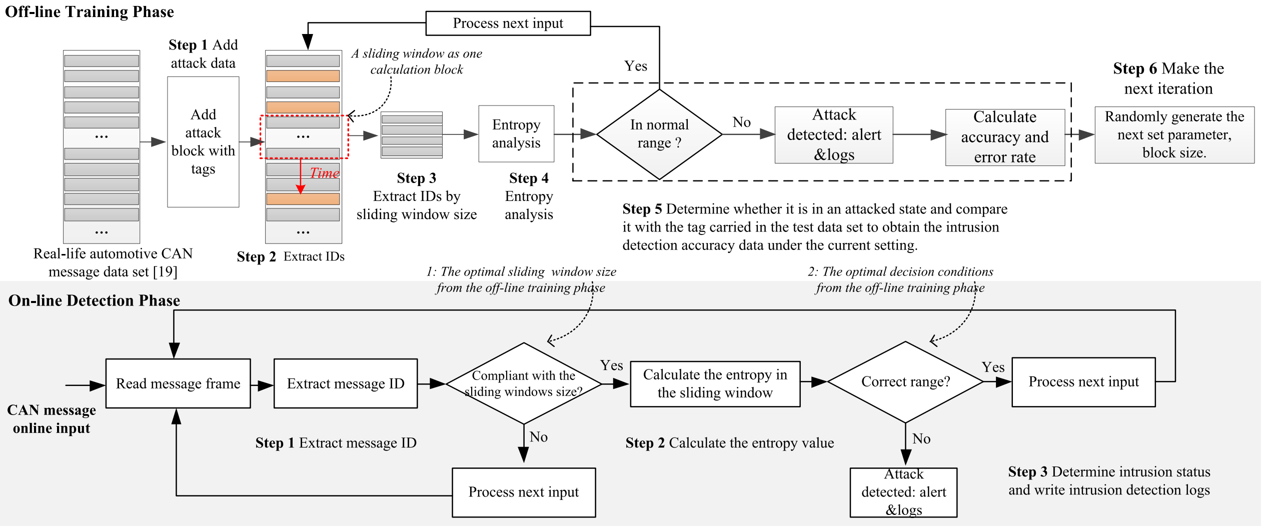

Off-line Training Phase
Process next input
Step 1 Add attack data
Add attack block with tags
A sliding window as one calculation block
Time
Entropy analysis
Yes
In normal range ?
No
Attack detected: alert &logs
Calculate accuracy and error rate
Step 6 Make the next iteration
Randomly generate the next set parameter, block size.
Step 3 Extract IDs by sliding window size
Step 4 Entropy analysis
Step 5 Determine whether it is in an attacked state and compare it with the tag carried in the test data set to obtain the intrusion detection accuracy data under the current setting.
Real-life automotive CAN message data set [19]
Step 2 Extract IDs
1: The optimal sliding window size from the off-line training phase
2: The optimal decision conditions from the off-line training phase
On-line Detection Phase
CAN message online input
Read message frame
Extract message ID
Compliant with the sliding windows size?
Yes
Calculate the entropy in the sliding window
Correct range?
Yes
Process next input
No
No
Step 1 Extract message ID
Step 2 Calculate the entropy value
Process next input
Attack detected: alert &logs
Step 3 Determine intrusion status and write intrusion detection logs

FIGURE 5. Diagram of proposed intrusion detection method.

focus on the optimization design of the sliding window. We define some features for our detection model as follows.

Definition 1: A window is used to monitor the network information entropy, and it can be fixed time or fixed number of messages. Time and number of messages serve as window parameters. We define it as the sliding window.

Definition 2: The time between the start of attack and its detection by IDS can be defined as response time of attack detection $R_t = A_t - D_t$, where A_t represents the start time of attacks, and D_t represents the time when the attack is detected. In this study, the attack start time A_t is obtained by inserting attack blocks with time tags.

A. PROBLEM DESCRIPTION

The automotive electronics system is a safety-critical system, and its correctness requires not only correct action but also response at the right time. Therefore, the response time to intrusion attacks is also an important evaluation index of intrusion detection systems. The main concern of this study is how to implement intrusion detection with low response time in the vehicle network environment. The research problem can be described as how to implement a high-precision, low-latency intrusion detection design in the in-vehicle network environment. The in-vehicle network environment has the following characteristics: resource constraints (including computing, storage, and network bandwidth resources), cost-sensitive, and security-critical.

B. FRAMEWORK OF THE PROPOSED METHOD

As shown in Figure 5, the framework of our proposed method is mainly divided into two phases. One is the off-line training phase. The Simulated Annealing sliding algorithm is used to get an optimal sliding window parameter. The second phase is the on-line detection phase. According to the parameters obtained in the training phase, an intrusion detection system is set up to detect abnormal intrusions.

1) OFF-LINE TRAINING PHASE

The training phase of the method mainly includes the following steps.

- **Step 1:** add attack block (carrying tags for evaluation) to the test data set. More details on the generation of data sets used in the training phase are described in the following Section V.
- **Step 2:** extract the CAN message IDs from the test data set to get a new IDs set.
- **Step 3:** use the sliding window as an information entropy sampling window to analyze the information entropy of the message IDs captured within the sliding window.
- **Step 4:** compare the information entropy obtained in Step 3 with the range of normal information entropy to determine whether there is an attack block.
- **Step 5:** compare the attack result obtained in Step 4 with the tag of the added attack block, and obtain the initial detection accuracy data.
- **Step 6:** select the new sliding window size and normal range for the next iteration. The whole training process uses simulated annealing algorithm and repeats the above Step 1 to Step 6 to obtain the best sliding window size and decision conditions with the highest detection accuracy.

2) ON-LINE DETECTION PHASE

This phase mainly includes the following repeated steps.

- **Step 1:** read the CAN network message online and store in the cache. After the number reaches the set sliding window size, information entropy calculation is performed.

- **Step 2:** compare the information entropy obtained in Step 1 with the normal information entropy range to obtain the intrusion detection result.
- **Step 3:** update system intrusion detection logs and trigger an alert if an attack is detected.

It should be emphasized that each intrusion detection method can only aim at a specific attack model. The method proposed in this study is mainly aimed at attack models that cannot be handled with other security enhancement methods, and it is also a common attack model in in-vehicle networks. Future research can consider adding the third phase, which is the runtime-monitoring/adopting phase. In this phase, the sensitivity of the intrusion detection system can be automatically adjusted based on false positive confirmation.

C. ENTROPY EVALUATION MODEL

In order to detect the information entropy of in-vehicle network, it is necessary to establish a model of information entropy based on the characteristics of CAN bus. The CAN messages have low entropy, with average 11.436 bits [23]. We calculated message ID's entropy according to Shannon entropy definition. Assuming system X, its limited set of possible states is $\{x_1, x_2, \ldots, x_N\}$, then the information entropy of system X is

$$H(X) = -\sum_{i=0}^{N} p(x_i) \log p(x_i), \tag{1}$$

where $p(x_i)$ is the probability of system X in state x_i.

1) ENTROPY OF CAN IDS

For the evaluation of the information entropy of CAN IDs, a CAN system model can be represented by $\Phi = (I, W)$, where $I = \{id_1, id_2, id_3 \ldots, id_n\}$ is a set of different IDs appearing within sliding windows size W. Subsequently, the information entropy value of CAN IDs in sliding windows size W can be expressed as

$$H(I) = -\sum_{id \in I} p_{id} \log p_{id}. \tag{2}$$

Since this paper determines the network state by detecting and monitoring the in-vehicle network, the total number of messages in the sample window is determined, and the total number of messages N_{total} in sample window W can be obtained by Equation (3):

$$N_{total} = \sum_{i=1}^{n} Count_{id_i}. \tag{3}$$

The number of id_i appears in W is obtained by counting. Then the probability of id_i appears in W can be represented as

$$P\,(id_i) = \frac{Count_{id_i}}{N_{total}}. \tag{4}$$

Obviously $\sum_{i=1}^{n} P(id_i) = 1$, $P(id_i) > 0$ $(i = 1, 2, \ldots, n)$. The uncertainty of defining id_i is its self-information. The self-information of id_i is

$$U_{id_i} = \log \frac{1}{P(id_i)} = \log \frac{N_{total}}{Count_{id_i}}. \tag{5}$$

Then, in the sampling windows W, the entropy of IDs on the in-vehicle network is

$$H(I) = E[U(id_i)] = \sum_{i=1}^{n} H_{id_i}, \tag{6}$$

where

$$H_{id_i} = P(id_i)U(id_i) = \frac{Count_{id_i}}{N_{total}} \times \log \frac{N_{total}}{Count_{id_i}}. \tag{7}$$

D. INTRUSION JUDGMENT

To determine whether the network is under attack status, we set the following parameters: u_e means average information entropy, σ_e is the corresponding standard deviation. For each sliding window size W, the detection algorithm compute $H(I)$ according to Equation (6). If $H(I)$ is not in normal range $[u_e - k\sigma_e, u_e + k\sigma_e]$, it is considered to have been attacked, where k is used to set the sensitivity of the deviation σ_e, $0.001 \leq k \leq 2$.

The change of setting of the sliding window has a significant influence on the information entropy. This study focus on how to obtain the best sliding window parameters, the sliding window size W, and deviation σ_e. A simulated annealing based algorithm is used to obtain the best parameters in this study. More details are introduced in Section IV-B.

IV. ALGORITHMS DESCRIPTION

Our goal is to obtain a high-precision real-time intrusion detection method for in-vehicle networks that can be deployed on the current in-vehicle gateway node with a low amount of efforts. For this purpose, the proposed intrusion detection method can be deployed as a software plug-in in a vehicle gateway platform or as a hardware node to be loaded onto an existing CAN bus. First, we design an information entropy-monitoring algorithm based on the fixed number of messages as the sliding window. Second, we use a simulated annealing algorithm to optimize the parameter settings of the sliding window.

A. INFORMATION ENTROPY-BASED INTRUSION DETECTION ALGORITHM

In brief, the entropy measurement algorithm can be expressed as monitoring of the information entropy of CAN bus traffic flow in a single sampling period. The complete design flow of the proposed entropy measurement algorithm is depicted in Algorithm 1. The false negative and false positive rates are main indicators of intrusion detection system, which are measured following [24] in this study. R_A indicates the detection rate of attack packets (detection accuracy), given as,

$$R_A(\%) = \frac{D_A}{T_A} \times 100, \tag{8}$$

and R_N indicates the detection rate of normal blocks (false positive rate), given as,

$$R_N(\%) = \frac{D_N}{T_N} \times 100, \tag{9}$$

where T_A is the number of total attacking blocks, and D_A is the number of attacking blocks detected by the IDS, T_N is the total number of normal blocks, and D_N is the number of normal packets detected by the IDS. An attack block consists of a continuous CAN message from the attack node.

Algorithm 1 Information Entropy-Based Intrusion Detection Algorithm ()

```
Input: $Test\_Data, I \leftarrow \{id_1, id_2, id_3..., id_W\}, k, \sigma_e, W$
Output: $R_A, R_N, R_t$
1.  $IDs \leftarrow Test\_Data\{id_1, id_2, id_3..., id_W\}$
2.  while I in $Test\_Data$ do
3.    while $id_i$ in $I$ do
4.      Calculate $H(id_i)$ according to $\frac{Count_{id_i}}{N_{total}} \times \log \frac{N_{total}}{Count_{id_i}}$;
5.    end while
6.    Calculate $H(I)$ according to $\sum_{i=1}^{n} H_{id_i}$;
7.    if $H(I)$ not in normal range $[u_e - k\sigma_e, u_e + k\sigma_e]$ then
8.      Calculate $R_A$ and $R_N$, based on Equation (8) and
        Equation (9), respectively
9.      $R_t \leftarrow Current_{time} - Attack_{time}$
10.     Write Attack_log
11.   end if
12.   return Detection accuracy $R_A$, false positive rate $R_N$,
      maximum detection response time $R_t$
13. end while
```

The main idea of Algorithm 1 is to calculate the information entropy of all the message IDs that appear in the sliding window, where the *Test_Data* consists of time-line of CAN messages set which has the attack blocks, R_t means the response time of attack detection. Then, the network is monitored in real time against the intrusion in units of sliding windows. The fixed number of messages W is used as the sliding window in this study. The details of entropy measurement algorithm are explained as follows:

1) In Line 1, we specify the input and output parameters of the algorithm. The input *Test_Data* is real messages data from CAN bus received by the monitor node within a sample window [21]. The output is the detection accuracy and response time under this parameter.

2) In Lines 2-6, we calculated the information entropy for each ID in the sliding window. Then get the information entropy of the entire sliding window.

3) In Lines 7-10, we judge whether the entropy is within the normal range, and get the detection result.

The main time complexity is presented in Line 4. The time complexity of Algorithm 1 is $O(|W| \times \log |N|)$.

B. IMPROVED ENTROPY MEASUREMENT ALGORITHM

When designing an intrusion detection system for in-vehicle network environment, the R_A (detection accuracy rate) should be as high as possible, whereas the R_N (false positive rate) should be as low as possible. We employ a Simulated Annealing (SA) algorithm (i.e., Algorithm 2) to optimize parameters used in Algorithm 1, and propose the following energy function for the SA,

$$E() = C_1 \times R_A(\%) - C_2 \times R_N(\%) - C_3 \times R_t, \tag{10}$$

where $E()$ is the energy function representing the detection accuracy and efficiency of the proposed model. This function is based on Algorithm 1 and is in accordance with Equation (8) and Equation (9). Three weighted parameters C_1, C_2, and C_3, are used to assess the characteristics of the proposed intrusion detection system, where these parameters are fixed in the training phase. These parameters can be adjusted to obtain different characteristics of intrusion detection systems. Considering that the vehicles are safety-critical, we set $C_1 = 1$, $C_2 = 0.5$, and $C_3 = 1$, respectively.

The main idea of Algorithm 2 is to quickly determine the best sliding window size W and deviation σ_e by simulated annealing to achieve high-precision and fast intrusion detection. $(\sigma_e, W)\_set\_0$ is an initial solution that is randomly generated, where σ_e is the deviation, k is the sensitivity of the deviation, and u_e is the average information entropy. The goal of Algorithm 2 is to obtain the parameter settings $(\sigma_e, W)\_set\_best$ in the case of maximizing $E()$.

Algorithm 2 Sliding Window Optimization Algorithm by Using Simulated Annealing ()

```
Input:$Data\_set\_with\_attacks, T_{max}$
Output: $(\sigma_e, W)\_set_{best}$
1.  $k\_best \leftarrow k_0$
2.  $\sigma_e\_best \leftarrow \sigma_e 0$
3.  $W\_best \leftarrow W_0$
4.  $e_{best} \leftarrow E((\sigma_e, W)\_set_0)$
5.  $T \leftarrow 0$
6.  while $T < T_{max}$ and $e < e_{best}$ do
7.    $(\sigma_e, W)\_set_{next} \leftarrow neighbor((\sigma_e, W)\_set)$
8.    $e_{next} \leftarrow E((\sigma_e, W)\_set_{next})$,
      Calculated by Algorithm 1 with $(\sigma_e, W)\_set_{next}$ input
9.    if $random() < P(e, e_{next}, temp(T/T_{max}))$ then
10.     $(\sigma_e, W)\_set \leftarrow (\sigma_e, W)\_set_{next}; e \leftarrow e_{next}$
11.     if $e > e_{best}$ then
12.       $(\sigma_e, W)\_set_{best} \leftarrow (\sigma_e, W)\_set; e_{best} \leftarrow e$
13.     end if
14.   end if
15.   $T \leftarrow T + 1$
16. end while
17. return $(\sigma_e, W)\_set_{best}$
```

The sliding window optimization algorithm is described in Algorithm 2, *neighbor*()is a neighbor function that generates the candidates of $(\sigma_e, W)\_set$ randomly. The main time complexity of Algorithm 2 is presented in Line 7. Because the time complexity of Algorithm 1 is $O(|W| \times \log |N|)$, the time complexity of Algorithm 2 is $O(|T| \times \log |N| \times |W|)$.

V. EXPERIMENTS AND EVALUATION

Using the described attack scenarios and detection algorithm, we conduct the following three experiments to evaluate and illustrate the effectiveness of our method. In Experiment 1, we analyze the effects of different sliding window strategies on information entropy analysis. In Experiment 2, the impact of sliding window size on intrusion detection performance, including response time and detection accuracy, is tested and analyzed. In Experiment 3, the effect of attack detection in real-life automotive network traffic test data under the best sliding window parameters is determined according to Algorithm 2.

A. TEST DATA SET

The diversity and uncertainty of automotive attacks bring difficulty in obtaining a data set for the in-vehicle cyber security research. In this study, we use the real-life automotive environment CAN bus network data set provided in [21].

Table 1 shows the data set features. The DoS attack data set in the table is generated by inserting CAN message blocks with $ID = 0x000$ into the real-life normal vehicle data set. One of the attack blocks refers to a continuous attack message. The attack block size setting range is 5-70. To generate a data set containing an injection attack for simulation, we copy the message block from the CAN message sent by the legitimate ECU to the normal vehicle data set and then inject it into the test data set. Considering the uncertainties of car attacks, 1000 DoS attacks and 1000 injection attacks appear in Gaussian distribution throughout the entire test data set.

TABLE 1. Test data set description.

	Count	Time	ID ranges	Attack times
Normal	30000	303ms	0x001-0x7FF	0
DoS	35000	303ms	0x000-0x7FF	1000
Injection	35000	303ms	0x000-0x7FF	1000

B. EXPERIMENT 1: ANALYSIS OF DIFFERENT SLIDING WINDOW STRATEGIES

To evaluate the information entropy analysis effect under different sliding window strategies, we perform two information entropy observation experiments under different strategies. Figure 6 shows the two monitoring results. The X-axis in the figure represents time (the X-axis is uniformly converted to time for comparative analysis), and the Y-axis represents the information entropy value in single sliding window. In Experiment 1, two data sets are used to analyze the effects of different sliding window strategies on information entropy. The data set 1 is the real-life in-vehicle network communication data set [21]. Based on data set 1, data set 2 is generated after adding DoS attack block.

In previous studies, the observation of the bus information entropy is accordance with the fixed time window [19], [20]. Figure 6 (a) shows the entropy analysis when the sample window is set as 100ms. CAN bus is an event-triggered network. Thus, the fixed number of messages is used as the observation window in this study. The average number of messages in 100ms sample windows in our test data set is 50 messages. Therefore, a sliding window size of 50 messages is used in the comparison, as shown in Figure 6 (b). Figure 6 (a) and Figure 6 (b) show the entropy analysis results of the message ID under normal network conditions, and Figure 6 (c) and Figure 6 (d) show the entropy analysis results under attack, where information entropy value in one sliding window is obtained based on Algorithm 1.

Observation: From the comparison of the information entropy analysis experiments under different sliding window strategies (fixed time and fixed message number based), we can draw the following observations:

1) When the fixed time is used as the sliding window, the information entropy of the entire CAN message fluctuates greatly, even if information entropy evaluation is performed on normal data. This strategy of sliding window with a fixed time is not conducive to the intrusion detection of the in-vehicle network.

2) When the fixed number of messages is set as a sliding window, the information entropy of the entire test data set fluctuates slightly when the normal data is analyzed. When analyzing the data set containing the attacks, the information entropy began to fluctuate. As described in Table 2, the average information entropy is 2.987 and the maximum deviation is 2.637.

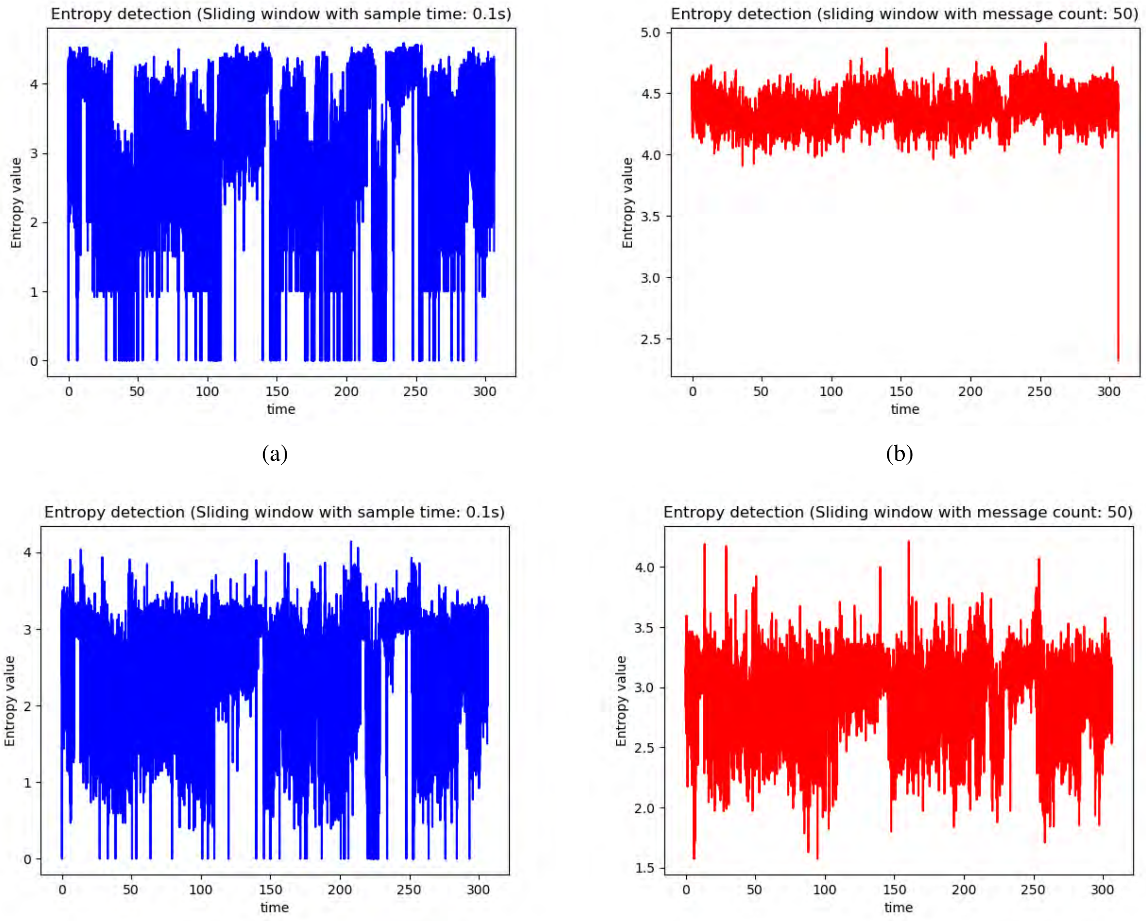
3) The information entropy drops sharply because the number of messages is insufficient in the last sliding window (end of Figure 6 (b)).

The results of Experiment 1 demonstrate that different sliding window strategies greatly impact the intrusion detection effect. We find that using the fixed number of messages as the sliding window, which is conducive to the analysis of intrusion detection design based on information entropy analysis. Compared with the previous method that uses fixed time as the sliding window, the strategy that uses the fixed number of message can effectively reduce information entropy perturbation caused by aperiodic messages. Therefore, a sliding window using the fixed number of messages is adopted in this study to reduce detection interference and improve detection accuracy.

C. EXPERIMENT 2: ANALYSIS OF SLIDING WINDOW SIZE COMPARISON

We attempt to analyze the effect of different size of sliding window on information entropy. We measure the information entropy under different size of sliding window use a real in-vehicle network message data set [21]. The largest intrusion detection response time is the window in which the maximum time is exaggerated. Figure 7 shows the influence of different sliding window sizes on information entropy.

Observation: From the measurement of the information entropy of real-world vehicle data set in different sliding window size, we draw the following observations:



(a)

(b)

(c)

(d)

FIGURE 6. Comparative analysis of information entropy monitoring under different sliding window strategies. (a) Entropy analysis of data set without attacks (based on a fixed-time sliding window strategy). (b) Entropy analysis of data set without attacks (sliding window strategy based on fixed number of messages). (c) Entropy analysis of data set including DoS attacks (based on a fixed-time sliding window strategy). (d) Entropy analysis of data set including DoS attacks (sliding window strategy based on fixed number of messages).

TABLE 2. Influence of different sliding window strategies on information entropy.

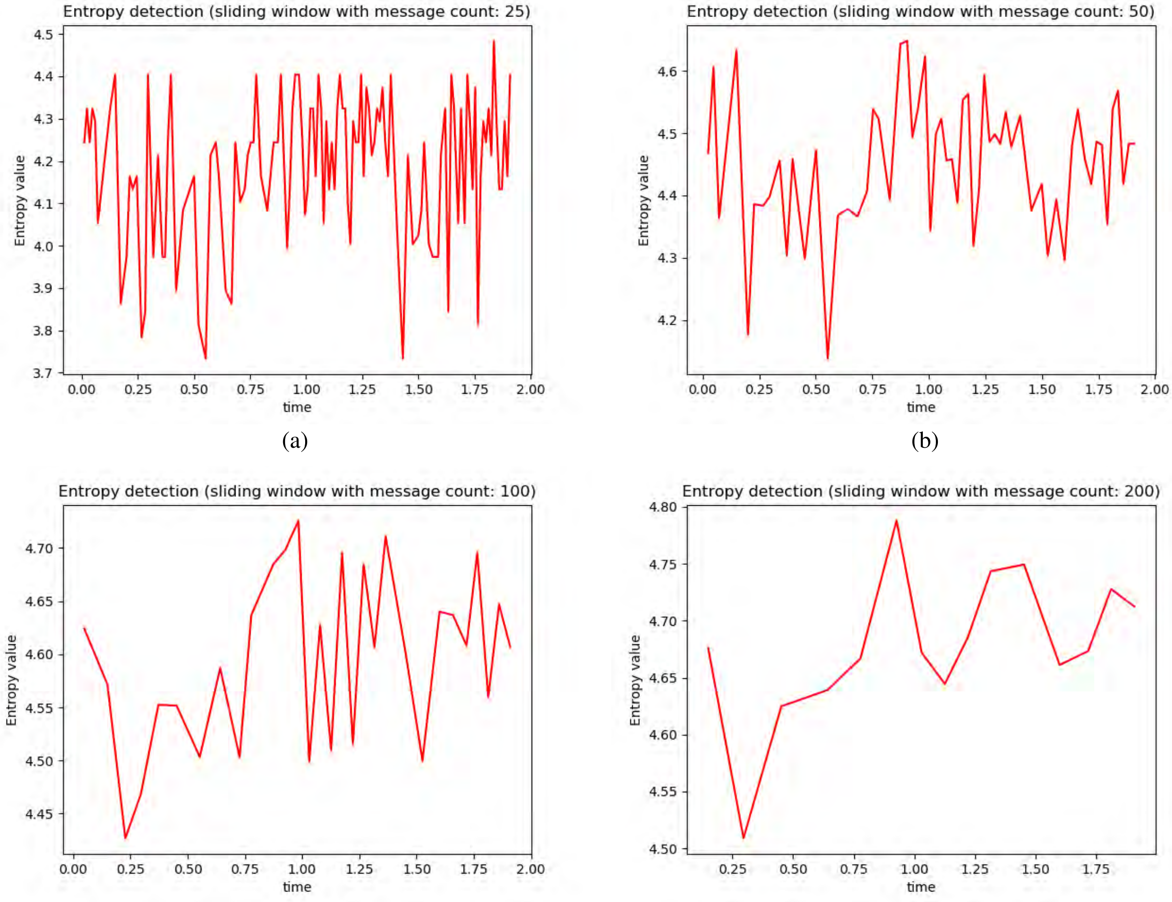
Strategies	Data set type	Maximum H	Minimum H	Average H	Relative difference
Fixed time period	No attacks [21]	4.584	0.0	3.100	4.584
Fixed number of messages	No attacks [21]	4.908	3.907	4.398	1.000
Fixed time period	With attacks [21]	4.142	0.0	2.492	4.142
Fixed number of messages	With attacks [21]	4.211	1.573	2.987	2.637

1) The size of the sliding window plays a major role in reducing the data resource and increasing accuracy. Large observed sliding window corresponds to smooth change in the obtained information entropy and large average value of the obtained information entropy.

2) Large sliding window size corresponds to small information entropy value. Meanwhile, the maximum response time to intrusion detection will be large. As described in Table 3, the maximum response time for intrusion detection gradually increase from 0.054 *ms* to 0.189 *ms*.

3) We obtain the best sliding window parameters for the test data set using Algorithm 2. The size of the sliding window in the proposed intrusion detection system is 60 CAN messages to realize a trade-off between response time and instruction accuracy.

The results of Experiment 2 demonstrate that the sliding window parameters are important to the performance of intrusion detection system, which will affect the accuracy and response time. As described in Table 3, the sliding window size affects not only the detection accuracy but also the



(a)

(b)

(c)

(d)

FIGURE 7. Comparison of information entropy monitoring under different message value sliding window. (a) When the sliding window size is set to 25. (b) When the sliding window size is set to 50. (c) When the sliding window size is set to 100. (d) When the sliding window size is set to 200.

TABLE 3. Influence of different size of sliding windows on information entropy.

Sliding window size	Maximum H	Minimum H	Average H	Relative difference	Maximum $R_t(ms)$
25	4.483	3.733	4.186	0.750	0.054
50	4.648	4.138	4.451	0.345	0.076
100	4.725	4.426	4.595	0.298	0.100
200	4.787	4.508	4.678	0.279	0.189

maximum response time of intrusion detection. The maximum response time refers to the start of the attack in the sliding window until it is detected.

D. EXPERIMENT 3: INTRUSION DETECTION EXPERIMENTS

The best design parameters of sliding window for testing its actual effect are derived using Algorithm 2. Considering the diversity and uncertainty of car attacks, the data set we used in Algorithm 2 is the real-life in-vehicle network communication data set [21]. The sliding window size in the proposed intrusion detection system is 60 CAN messages, σ_e is 0.52, and u_e is 4.186. In accordance with the attack scenario mentioned in Section II-C, we design two attack tests, which are DoS and injection attack on CAN bus. On the basis of the test data set mentioned in Section V-A, we develop the following experiments. The sliding window optimal parameters given in this study are for the test data set used in this study.

First, we select an analysis of the DoS attack data for illustration. Figure 8 shows that, when an attack on the CAN network occurs, the attack data will cause the network state to change. This change is reflected in the information entropy. As shown in Figure 8, when the information entropy value in the sliding window is detected to be out of the normal range, the system will make an intrusion warning.

Then, we simulate the data collection under two attack scenarios to evaluate the effectiveness of our method. Two types of attack data (DoS and Injection attacks) are generated.

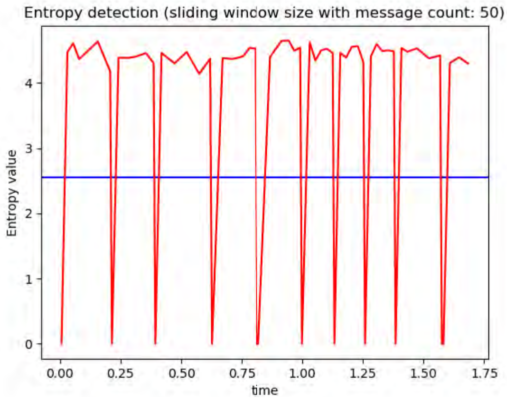


FIGURE 8. When the information entropy value in the observation window is detected to be out of the normal range, the system will make an intrusion warning.

The attack data in the entire data set are in Gaussian distribution. To simulate the DoS attack scenario, we consider the uncertainty of the DoS attack. We add high-priority DoS attacks to the real-life vehicle network data set given the Gaussian distribution. In order to generate the Injection attack scenario, we employ the replay attack, which fabricates the previously sending CAN messages. Subsequently, we conduct intrusion detection on the two attack scenarios.

The experimental results as shown in Figure 9. The X-axis in the figure represents the sensitivity of the deviation k, and the Y-axis represents the information entropy value in single sliding window. During the experiment, we find that the setting of the sensitivity of the deviation k greatly influences the detection accuracy, we can draw the following observations.

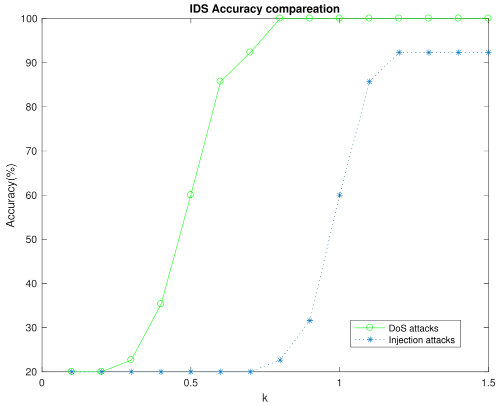


FIGURE 9. Comparative analysis of detection accuracy on DoS and injection attacks under different sensitivity of the deviation k.

Figure 9 shows that the effect of the proposed method on the DoS attack rapidly increase when the sensitivity of the deviation k is 0.5, whereas the same situation occurs for the injection attack when the sensitivity of the deviation k is 0.8. In addition, when the sensitivity of the deviation k increases to a certain extent, the detection accuracy does not increase after reaching a certain value. In Figure 9, we also can see that the detection accuracy of the injection attacks reaches its maximum value requires that the value of k is 1.1, however, the value of k is required to be smaller than 1 for the case of DoS attacks detection. Therefore, considering the unknown type of attack, how to determine the appropriate k value is an unresolved issue in this study, and will serve as an important issue for our future research. The best intrusion detection results of our method are described in Table 4, the detection accuracies of the proposed method for DoS and injection attacks are 100% and 92.3%, respectively. And, the false detection rate is 0% in both cases.

TABLE 4. Detection accuracy.

Attack type	Attack samples	Normal samples	Detection accuracy R_A	False positive rate R_N	$R_t(ms)$
DoS	200	20000	100%	0%	0.081
Injection	200	10000	92.3%	0%	0.081

Lastly, for the same experimental conditions, where the sensitivity of the deviation k is 0.6, σ_e is 0.52, and u_e is 4.186. We implemented the entropy-based intrusion detection method proposed in [20], which used a fixed time as sliding window. Consider the average interval of 60 CAN messages in the data set is 0.11S. Therefore, we used 0.11S as the size of the sliding window in the comparison experiment. We generated test data with different attack block sizes, the intrusion detection effect of the two methods is shown in Figure 10.

Observation: From the detection accuracy comparison experiment under different sliding window strategies (fixed time [20] and our fixed message count based), we draw the following observations:

1) As described in Figure 10 (a), for the DoS attacks, when the attack block is greater than 60 CAN messages, our method and the method of [20] have higher detection accuracy.

2) As described in Figure 10 (b), for the Injection attacks, when the size of the attack block is greater than 30, our method achieves the detection accuracy (92.3%). The method of [20] achieves the detection accuracy (91.0%) when the attack block is greater than 50.

3) The experimental results show that the method of [20] has obvious effect when the attack block is large, and the detection result is not ideal for the smaller attack block.

4) In terms of algorithm computational complexity, since they are all information entropy-based, the two sliding window strategies have the same algorithm complexity. However, in the real-life operation process, it is considered that the CAN network is an event-triggered network, that is, the message distribution in the network in the time domain is not uniform. Therefore, the fixed-time based sliding window strategy will include some invalid periodic calculations, and our sliding window strategy can more efficiently utilize computing resources to analyze message information entropy.

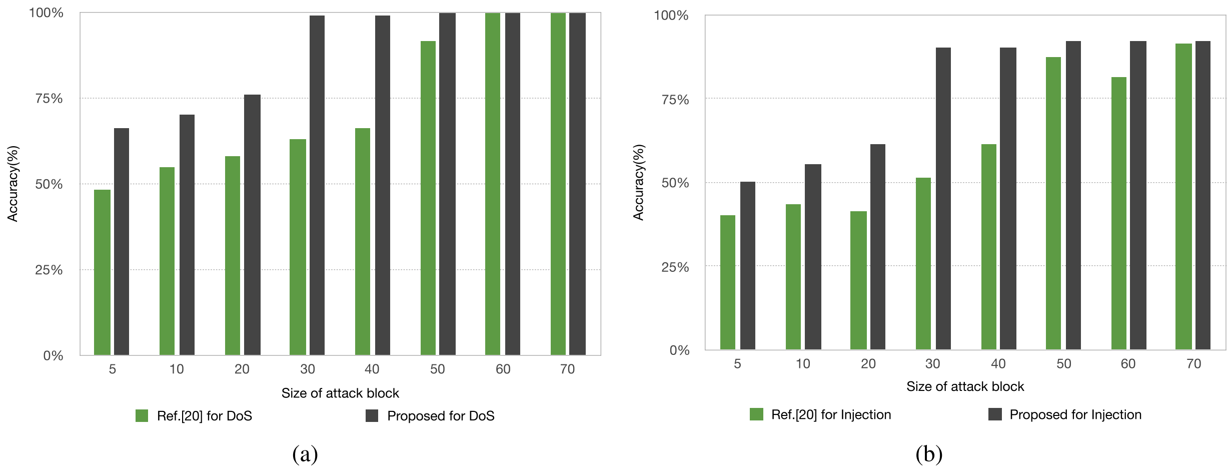


FIGURE 10. Comparative analysis of detection accuracy on DoS and injection attacks under different attack block sizes. (a) Comparison of the DoS attacks. (b) Comparison of the Injection attacks.

Summary: From the intrusion detection experiments simulated for two attack scenarios, we find that the optimized sliding window strategy improves the accuracy of the intrusion detection system based on information entropy. Compared with existing intrusion detection methods for in-vehicle networks [20], Our method has high intrusion detection accuracy for attacks of different size of attack blocks. The accuracy of intrusion detection is 100% for DoS and 92.3% for injection by our method. We also highlight that our intrusion detection method does not require large response time during our experimental evaluation. Therefore, an intrusion method of low cost, high compatibility, and low response time is particularly suitable for abnormal intrusion detection in the automotive network environment.

VI. RELATED WORKS

The security enhancement methods for in-vehicle network mainly include message encryption and authentication, firewall, and intrusion detection. Compared with other methods, intrusion detection method can save valuable bandwidth effectively, and is compatible to a large number of existing vehicles on the market. Researchers in [25] and [26] showed that when an attacker injects legitimate messages to perform a spoofing attack or a DoS attack, the frequencies will increase. Such detection methods are possible with good accuracy and low false positive rate, but only works for periodic traffic. Other popular designs for anomaly detection for in-vehicle networks are based on fingerprint information [18], [27], [28].

Machine-learning-based methods have also been applied to intrusion studies of in-vehicle networks [24], [29]. However, the high demands on their computational performance lead to the inapplicability to the current in-vehicle environment. The method based on information entropy has the characteristics of low cost and strong compatibility. Thus, this method is considered to be suitable for abnormal intrusion detection in the vehicle network environment. The more relevant previous research related to this study is [19] and [20].

In [19], Muter and Asaj introduced the concept of entropy-based attack detection for in-vehicle networks for the first time. They focused on messages with the same ID instead of considering all CAN traffic data. The advantage of this approach is that the ID can be accurately attacked to determine the specific attack type. The disadvantage is that it can easily influence the information entropy of the messages with the same ID due to the change in the operating state of the automobile; thus, it is erroneously detected. The CAN is an event-triggered network. Thus, messages often have different periods (ranging from 0.01 *ms* to a few seconds), which may result in disappearance of some messages in the observation window.

In [20], Marchetti *et al.* evaluated the effectiveness of information-theoretic anomaly detection algorithms applied to networks included in modern vehicles. They found through experiments that the direct use of information entropy-based anomaly intrusion detection to all CAN messages is only effective in the case of a large number of forged CAN message.

Our method, which considers the characteristics of the CAN and the situation of the automobile electronic system environment, can be applied to periodic and aperiodic CAN environments to solve the shortcomings of existing solutions. The accuracy of intrusion detection is effectively improved and setting the optimal sliding window reduces the response time.

VII. CONCLUSION

Intrusion detection method based on information entropy, which considers the constraints of automotive costs and computing performance and the diversity and uncertainty of attacks on automotive networks, has the characteristics of low cost and strong compatibility. Through analyzing the characteristics of CAN network, this study improves the analysis methods of previous information entropy-based IDS, and improves the accuracy of detection against in-vehicle network attacks with low response time. Parameters such as the best sliding window size, standard deviation, and corresponding sensitivity are obtained by a simulated annealing method adopting a sliding window design based on the fixed number of messages. The detection performance of the proposed approach is evaluated through experiments carried out in accordance with the real CAN traffic data set. The experimental results demonstrate that the proposed method can effectively improve the accuracy and effectiveness of intrusion detection for DoS and injection attacks on in-vehicle networks. The research that can be considered in the future is mainly to consider the influence of the state of the vehicle operating state on the information entropy.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the associate editor and anonymous reviewers for their constructive comments, which have helped to improve the quality of the paper.

REFERENCES

[1] NXP. (2017). *Future Advances in Body Electronics*. [Online]. Available: https://automotive.electronicspecifier.com/

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

[3] G. De La Torre, P. Rad, and K.-K. R. Choo, "Driverless vehicle security: Challenges and future research opportunities," *Future Gener. Comput. Syst.*, to be published.

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

[5] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," in *Proc. Black Hat USA*, 2014, pp. 1–90.

[6] *Road Vehicles-Functional Safety*, document ISO 26262, 2011.

[7] W. Wu, R. Kurachi, G. Zeng, Y. Matsubara, H. Takada, and R. Li, "IDHCC: A security-enhanced id hopping can controller design to guarantee real-time," in *Proc. 2nd Workshop Secur. Dependability Crit. Embedded Real-Time Syst. (CERTS)*, Oct. 2017, pp. 14–21. [Online]. Available: http://certs2017.uni.lu

[8] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (CAN) communication protocol," in *Proc. Int. Conf. Cyber Secur. (CyberSecurity)*, Dec. 2012, pp. 1–7.

[9] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," in *Proc. IEEE 68th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2008, pp. 1–5.

[10] F. Zhang, K. Szwaykowska, W. Wolf, and V. Mooney, "Task scheduling for control oriented requirements for cyber-physical systems," in *Proc. Real-Time Syst. Symp.*, Nov. 2008, pp. 47–56.

[11] P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Lightweight authentication for secure automotive networks," in *Proc. Design, Automat. Test Eur. Conf. Exhib.*, 2015, pp. 285–288.

[12] S. Chakraborty, M. A. A. Faruque, W. Chang, D. Goswami, M. Wolf, and Q. Zhu, "Automotive cyber–physical systems: A tutorial introduction," *IEEE Design Test*, vol. 33, no. 4, pp. 92–108, Aug. 2016.

[13] K.-K. R. Choo, M. Bishop, W. Glisson, and K. Nance, "Internet- and cloud-of-things cybersecurity research challenges and advances," *Comput. Secur.*, vol. 74, pp. 275–276, May 2018.

[14] G. Macher, H. Sporer, E. Brenner, and C. Kreiner, "An automotive signal-layer security and trust-boundary identification approach," *Procedia Comput. Sci.*, vol. 109, pp. 490–497, Jan. 2017.

[15] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in *Proc. Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1577–1583.

[16] M. Markovitz and A. Wool, "Field classification, modeling and anomaly detection in unknown CAN bus networks," *Veh. Commun.*, vol. 9, pp. 43–52, Jul. 2017.

[17] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through Hamming distance," in *Proc. AEIT Int. Conf.*, Sep. 2017, pp. 1–6.

[18] K. G. Shin and K. T. Cho, "Fingerprinting electronic control units for vehicle intrusion detection," U.S. Patent 15 472 861, Mar. 29, 2017.

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

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

[21] H. Lee, S. Jeong, and H. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. PST*, 2017, pp. 1–10.

[22] H. Ueda, R. Kurachi, H. Takada, T. Mizutani, M. Inoue, and S. Horihata, "Security authentication system for in-vehicle network," SEI Tech. Rev. 81, 2015.

[23] E. Wang, W. Xu, S. Sastry, S. Liu, and K. Zeng, "Hardware module-based message authentication in intra-vehicle networks," in *Proc. 8th Int. Conf. Cyber-Phys. Syst.*, Apr. 2017, pp. 207–216.

[24] M. J. Kang and J. W. Kang, "A novel intrusion detection method using deep neural network for in-vehicle network security," in *Proc. Veh. Technol. Conf.*, May 2016, pp. 1–5.

[25] C. V. Miller, "Remote exploitation of an unaltered passenger vehicle," in *Proc. Black Hat USA*, 2015, pp. 1–91.

[26] 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.*, Jan. 2016, pp. 63–68.

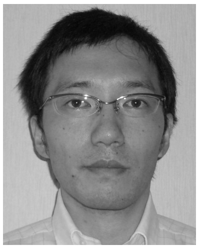
[27] P.-S. Murvay and B. Groza, "Source identification using signal characteristics in controller area networks," *IEEE Signal Process. Lett.*, vol. 21, no. 4, pp. 395–399, Apr. 2014.

[28] K. Cho and K. G. Shin, "Viden: Attacker identification on in-vehicle networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, Dallas, TX, USA, Oct. 2017, pp. 1109–1123, doi: 10.1145/3133956.3134001.

[29] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ECUs using inimitable characteristics of signals in controller area networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4757–4770, Jun. 2018.

WUFEI WU (S'17) is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, China. His research interests include distributed system, embedded computing systems, and cyber-physical systems. He is a Student Member of ACM and CCF.

YIZHI HUANG is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University, China. His research interests include embedded and cyber-physical systems and heterogeneous computing System.



RYO KURACHI (M'18) received the bachelor's degree in applied electronics from the Tokyo University of Science, the master's degree in management of technology from the Tokyo University of Science in 2007, and the Ph.D. degree in information science from Nagoya University in 2012. After a few years working with AISIN AW CO., LTD., as a Software Engineer. He is currently a Designated Associate Professor with the Center for Embedded Computing Systems, Nagoya University. His research interests include embedded systems and real-time systems. Within that domain, he has investigated topics, such as in-vehicle networks and real-time scheduling theory and embedded systems security.

GANG ZENG (M'03) received the Ph.D. degree in information science from Chiba University in 2006. From 2006 to 2010, he was a Researcher, and then an Assistant Professor with the Center for Embedded Computing Systems (NCES), Graduate School of Information Science, Nagoya University. He is currently an Associate Professor with the Graduate School of Engineering, Nagoya University. His research interests mainly include power-aware computing and real-time embedded system design. He is a member of IPSJ.

GUOQI XIE (M'15) received the Ph.D. degree in computer science and engineering from Hunan University, China, in 2014. He was a Post-Doctoral Research Fellow with Hunan University from 2015 to 2017, and with Nagoya University, Japan, from 2014 to 2015. He has been an Associate Professor with the College of Computer Science and Electronic Engineering, Hunan University, since 2017. His current research interests include design automation of automotive cyber-physical systems, embedded and cyber-physical systems, and parallel and distributed systems. He received the Best Paper Award at IEEE ISPA 2016. He is currently serving on the editorial boards of the *Journal of Systems Architecture*, the *Journal of Circuits, Systems and Computers*, and *Microprocessors and Microsystems*.

RENFA LI (M'05–SM'10) is currently a Professor of computer science and electronic engineering, and the Dean of the College of Computer Science and Electronic Engineering, Hunan University, China. He is the Director of the Key Laboratory for Embedded and Network Computing of Hunan Province, China. He is also an expert committee member of the National Supercomputing Center, Changsha, China. His major interests include computer architectures, embedded computing systems, cyber-physical systems, and Internet of Things. He is a member of the council of CCF and a Senior Member of ACM.

KEQIN LI (M'90–SM'96–F'15) is currently a Distinguished Professor of computer science with the State University of New York. He has published over 580 journal articles, book chapters, and refereed conference papers. His current research interests include parallel computing and high-performance computing, distributed computing, energy-efficient computing and communication, heterogeneous computing systems, cloud computing, big data computing, CPU-GPU hybrid and cooperative computing, multicore computing, storage and file systems, wireless communication networks, sensor networks, peer-to-peer file sharing systems, mobile computing, service computing, Internet of Things, and cyber-physical systems. He has received several best paper awards. He is currently or has served on the editorial boards of the IEEE Transactions on Parallel and Distributed Systems, the IEEE Transactions on Computers, the IEEE Transactions on Cloud Computing, the IEEE Transactions on Services Computing, and the IEEE Transactions on Sustainable Computing.