IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 March 2023, accepted 29 March 2023, date of publication 6 April 2023, date of current version 13 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3265018

RESEARCH ARTICLE

Multiple Observation HMM-Based CAN Bus Intrusion Detection System for In-Vehicle Network

CHEN DONG, HAO WU, (Member, IEEE), AND QINGYUAN LI
State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China

Corresponding author: Hao Wu (hwu@bjtu.edu.cn)

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2022JBQY004, and in part by the National Natural Science Foundation of China under Grant 62221001, and in part by the Frontiers Science Center for Smart High-speed Railway System.

ABSTRACT As modern vehicles become more intelligent and connected, the number of ECUs and communication interfaces with external networks (such as 3G/4G and Bluetooth) has increased significantly, which raises potential network security risks. Due to the absence of effective security measures, there is a frequent occurrence of cybersecurity incidents targeting vehicles, particularly the in-vehicle CAN bus network. As the main bus in the vehicle, the CAN bus faces important challenges in its safety due to the lack of security mechanisms. Therefore, we propose a CAN bus intrusion detection system based on multiple observation HMM for in-vehicle networks to enhance the security of the vehicle. Specifically, the proposed algorithm builds a multiple observation HMM-based on the ID and data fields of normal CAN bus traffic. According to the established HMMs, we calculate the existence probability of the frame under the defined time window as the detection threshold. When the existence probability of the frame to be detected exceeds the normal threshold range, it is considered abnormal. Furthermore, we establish four common attack models based on the collected real vehicle data and evaluate the performance of the proposed algorithm in these attack scenarios. The experimental results show that the proposed method has better detection performance than other frame-by-frame anomaly detection methods in four attack scenarios.

INDEX TERMS CAN bus intrusion detection, HMM, multi-observation sequence, in-vehicle networks.

I. INTRODUCTION

With the sustained development of Intelligent Transportation Systems (ITS) in recent years, intelligent and connected vehicles (ICVs) have become a promising paradigm of modern vehicles, which leads to the increasing complexity of in-vehicle networks (IVNs) [1]. More than 100 electronic control units (ECUs) are deployed in IVNs, supporting various functions such as brake control, transmission control, body control, etc., and realize information exchange through the in-vehicle bus [2]. Furthermore, aiming at sharing more information with the environment to aid decision-making [3], [4], ICVs have added a series of external wireless interfaces such as Bluetooth, WiFi, and cellular network, but also increase remote attack surfaces. As a result, hackers can access to IVNs without direct contact with the vehicle, obtain private information of the vehicle, and even cause fatal accidents, which will pose a great threat to vehicle safety.

As the most widely used in-vehicle communication bus, the controller area network(CAN) is facing increasing unprecedented security threats. However, the CAN bus has not been designed with enough security mechanisms to deal with today's cyberattacks, as the vehicle was once considered to be a closed mechanical system. For example, since ECUs broadcast data through the CAN bus, all ECUs connected to the bus monitor these data in real time, which means that an attacker only needs to control one ECU to eavesdrop easily. In recent years, attacks against the CAN bus have frequently appeared. In 2015, Miller and Valasek [5] compromised the Jeep Cherokee via a vulnerability in the vehicle's entertainment system and then could bleed the brakes by sending messages to the CAN bus remotely. In 2019, Cai et al. [6]

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

demonstrated that an attacker could exploit vulnerabilities in BMW vehicles by sending messages to the CAN bus remotely to move the driver's seat and reset ECUs. In the 2022 global automotive cybersecurity report [7], Upstream Security stated that the number of automotive hacking attacks has increased by more than 2.25 times in 2021 compared to 2018.

Due to the rapidly growing issues of vehicle cybersecurity, extensive research has been conducted to address the risks and threats specifically on the CAN bus. In general, security solutions for the CAN bus are divided into three aspects: message authentication and encryption [8], intrusion detection system [9], and overall security frameworks. The main protection methods of encryption and authentication are authentication of frame messages or key encryption. However, due to the limited communication capacity of the CAN bus, encryption and authentication algorithms will increase the burden on the CAN bus and even affect the bus real-time communication. The overall security protection scheme, on the other hand, cannot be implemented in a short period as it requires to modify the existing hardware and software of the vehicle. Under current experimental conditions, intrusion detection is gradually becoming an important issue in vehicle network security research. Nonetheless, the current state-of-the-art intrusion detection methods for the CAN bus encounter significant challenges. Many methods aim to detect attacks over a period rather than identifying a single frame as an attack frame. Additionally, several intrusion detection schemes require a considerable amount of attack data, which is often challenging to collect in real-world scenarios. To this end, we propose a multi-observation HMM-based, frame-by-frame CAN intrusion detection system.

Specifically, our main contributions can be summarized as follows.

(1) We propose a multiple observation HMM-based CAN bus Intrusion Detection System(MOHIDS) for in-vehicle networks. The scheme comprehensively considers the ID characteristics and data domain characteristics of frames and only depends on the bus data during normal vehicle operation.

(2) We proposed a frame-by-frame detection algorithm, which improves the detection granularity and shortens the attack response time without changing the existing vehicle structure. In addition, our proposed algorithm is able to accurately locate specific attack frames rather than attack blocks.

(3) We construct four common attack scenarios and evaluate the proposed algorithm by real vehicle data and the attack platform for the CAN bus. The evaluation results show that the proposed method can effectively detect four attack scenarios and is superior to other single frame detection algorithms.

The remainder of this paper is organized as follows. In Section II, we discuss the related work on security cases and intrusion detection systems for the CAN bus. Section III introduces the basic characteristics, security vulnerabilities, and attack scenarios of the CAN bus. Section IV describes the design of MOHIDS. In Section V, we describe the experimental setup for evaluation and present the experiments and results. Section VI discusses MOHIDS's limitations and possible directions for future work.

II. RELATED WORK

A. CAN SECURITY ANALYSIS

In 2010, Koscher et al. [10] carried out a very comprehensive analysis on the CAN security of modern vehicles, providing the first experimental guidance research for the safety risks of modern vehicles. They thoroughly evaluated the security vulnerability of the CAN bus and took such attacks as sniffing, fuzzy testing, and reverse engineering against the important ECU of the vehicle. In the experiment, the author successfully controlled the key components of CAN bus, such as the engine, radio equipment, instrument system, lock, and wiper, and then analyzed the safety characteristics of various components in detail. In 2012, Lin et al. [11] divided attack nodes into strong attackers and weak attackers according to the difference in attack capabilities, and established an attack model to describe four attack scenarios including replay attack and forgery attack. Finally, a security mechanism with low bus load and low time cost was presented. In 2015, Woo et al. [12] has proved through experiments that it is possible for an attacker to carry out a remote wireless attack on the vehicle CAN bus by implanting malicious software into the smartphone application. When the victim used Bluetooth or WiFi to connect the smartphone to the target vehicle, the author completed four types of attack experiences: destruction of the dashboard, engine stop, handle control, and acceleration through the implanted malicious program. In 2022, Jo et al. [13] systematically analyzed the existing research on in-vehicle CAN attacks, and then summarized the common attack surfaces, different types of attackers, and corresponding multiple attack scenarios of car CAN.

B. INTRUSION DETECTION SYSTEM(IDS)

Current intrusion detection technologies of the CAN bus can be divided into three categories: (1) frequency-based intrusion detection techniques, (2) data field-based intrusion detection techniques, and (3) hybrid-based intrusion detection techniques.

(1) frequency-based intrusion detection techniques: Frames are usually transmitted on the CAN bus periodically, which allows the transmission interval or frequency to help establish a baseline for detection. In 2016, Song et al. [14] proposed a light-weight detection algorithm for CAN bus intrusion detection based on the periodic characteristics of CAN bus frames, but complex attack scenarios were not considered when analyzing intrusion data, thus the detection sensitivity was low and attacks against non-periodic frames could not be detected. In 2020, Ezeobi et al. [15] proposed a specification-based detection for the CAN bus which focuses on real-time schedule ability response time. They used the analysis of the real-time scheduling of the normal runtime bus to build the detection model and define that messages with non-compliant response times are considered anomalous.

In 2021, Han et al. [16] proposed a detection algorithm based on the periodic event-triggered interval of CAN ID. This approach applied the statistical values of the event trigger interval for each CAN ID to machine learning models including decision trees, random forests, and XGB, and evaluates performance against multiple attack types and two types of vehicles, but did not give the specific details of the attack.

(2) data field-based intrusion detection techniques: In addition to frequency-based detection methods, there are some complex algorithms that consider other features such as changes in data fields and semantic changes in messages. In 2016, Taylor et al. [17]used Long Short-Term Memory (LSTM) networks to detect message sequences in CAN networks. The algorithm can detect abnormal behavior with a low false positive rate, but it did not clearly provide applicable attack scenarios. Furthermore, since the algorithm treated the data sequence of each ID as independent, it lacked consideration of the correlation between IDs. In 2016, Kang et al. [18] proposed an intrusion detection system based on deep neural network(DNN) for CAN bus data whose features are taken from 8 bytes of the data domain. The proposed algorithm can provide a real-time response to attacks, with an average detection rate of about 98%. However, the types of attacks that can be handled are not detailed.

(3) hybrid-based intrusion detection techniques: In 2017, Markovitz and Wool [19] proposed a novel domain-aware intrusion detection system for the CAN bus. They used a greedy algorithm to separate unknown messages into different fields and developed an intrusion detection system by semantic analysis. The algorithm was evaluated based on real CAN messages and achieves low false positives. In 2021, Xie et al. [20] proposed an enhanced deep learning GAN model based on complex CAN message blocks. Compared with other intrusion detection models based on GAN, the proposed algorithm can effectively detect tampering attacks, but it can not achieve frame-by-frame detection, resulting in a longer response time. In recent years, researchers have proposed several frame-by-frame detection schemes to achieve low response latency for real-time detection. In 2021, Khan et al. [21] proposed a two-stage intrusion detection scheme based on bloom filter and bi-directional LSTM. In the first stage, the input data is sent to a classifier based on Bloom filters. If the computed state falls within the normal range, it is subsequently forwarded to a detector based on LSTM. The proposed algorithm achieved good performance in terms of accuracy and false alarm rate, but the two-stage detection process increased the detection time, which imposes certain requirements on the computing performance of the vehicular hardware. In 2021, Sun et al. [22] proposed an intrusion detection model named CNN-LSTM with Attention Model (CLAM)for the in-vehicle CAN. The CLAM model utilized one-dimensional convolution (Conv1D) to extract features from the input signal, followed by bi-directional LSTM to capture temporal dependencies. Finally, a soft attention mechanism was employed to enhance the detection accuracy. The model achieved exceptional anomaly detection performance in five typical attack scenarios. However, the training process imposes a high demand for the quantity and comprehensiveness of attack samples, which could pose challenges in practical applications.

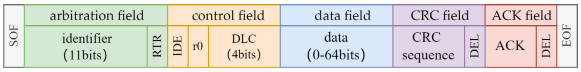

FIGURE 1. The frame structure of the CAN bus message.

III. CAN BUS AND ATTACK MODEL

In this section, we detail the characteristics of the in-vehicle CAN bus, analyze its security vulnerabilities, and propose four common attack scenarios.

A. CAN BUS

CAN network protocol transmission units called messages or frames are divided into the following four types: data frames, remote frames, error frames, and overload frames [22]. These four types of frames carry the tasks of data transmission, command requests, error warnings, and overload notifications, respectively. Nodes on the CAN bus send frames bit by bit, where data frames are the dominant ones used in the transmission. Depending on the length of the identifier (ID), CAN bus data frames are divided into base frames with 11-bit identifiers (CAN 2.0A) and extended frames with 29-bit identifiers (CAN 2.0B). The structure of a data frame [23] is shown in Fig. 1 The data frame starts with a start of frame(SOF) and ends with an end of frame(EOF). The ID in the arbitration field indicates the identity of the frame, which determines receiving nodes and the transmission priority. The data field, up to 8 bytes in length, contains specific information, such as vehicle speed, water temperature, and engine speed. The Data Length Code(DLC) in the control field presents the size of the data field in bytes.

Due to numerous safety-sensitive applications in automobiles, the CAN bus uses a Multi-Master communication system, in which messages are broadcasted. The CAN bus resources are allocated using Carrier Sense Multiple Access with Collision Detection (CSMA/CD) bit-by-bit arbitration mechanism. The node listens to the bus in real-time and sends messages only when the bus is free. If multiple nodes send frames at the same time, the IDs of all frames are compared bit by bit and the frames with higher IDs are dropped. In other words, frames with lower IDs have higher priority. Nodes connected to a bus can listen to all messages of the bus and filter messages according to the frame identifier. In addition, the frame does not contain the addresses of its sending and receiving nodes. The node filters out the desired frames based on the IDs when listening to the bus. In fact, each ECU contains two sets of a limited number of frame identifiers.

Fig. 2 shows a CAN bus communication system with three ECUs. For the node ECU_i,we define the set of producer identifiers as $SetPID_i = \{P_{i1}, \cdots, P_{in}\}$ and the set of consumer identifiers as $SetCID_i = \{C_{i1}, \cdots, C_{in}\}$. When ECU_i sends a frame with$ID_f = P_{i1}$, $ECU_j(i \neq j)$will receives

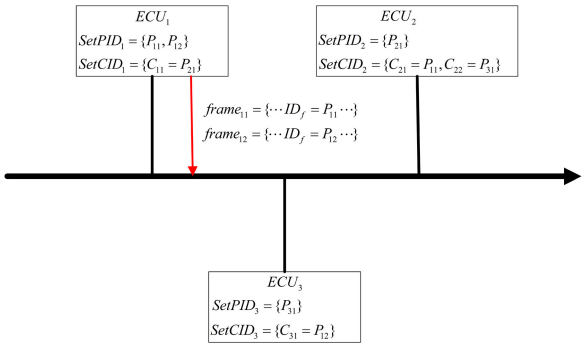

FIGURE 2. The CAN communication model.

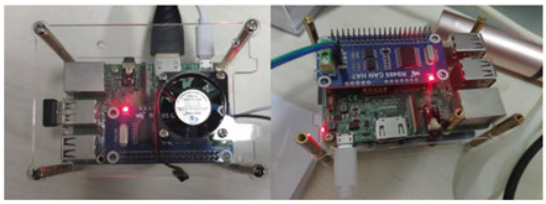

FIGURE 3. The CAN bus transceiver built with RS465 CAN HAT and raspberry pie microcomputer.

the frame if $ID_f \in SetCID_j$. A CAN bus communication system with 3 ECUs is shown in Figure. ECU_1 sends $frame_{11}$ with $ID_f = P_{11}$ and $frame_{12}$ with $ID_f = P_{12}$to the bus. For $P_{11} = C_{21} \in SetCID_2$ and $P_{11} \notin SetCID_2$, $frame_{11}$ is received by ECU_2 instead of ECU_3. Similarly, $frame_{12}$ is received by ECU_3 instead of ECU_2.

B. PRACTICAL EXPERIMENTS AND CAN BUS SECURITY VULNERABILITY ANALYSIS

In order to fully study the running state of the real vehicle CAN bus and the actual effect of the attack, we built two CAN bus transceiver devices as shown in Fig. 3 to obtain and send bus data through OBD-II. The transceiver is built with RS465 CAN HAT and Raspberry Pie. First, the control interface reserved by RS465 CAN HAT is connected to the GPIO interface of Raspberry Pie. Secondly, the H and L interfaces of RS465 CAN HAT are connected to the H and L lines of OBD-II interface of the CAN bus. First, the control interface reserved by RS465 CAN HAT is connected to the GPIO interface of Raspberry Pie. Secondly, the H and L interfaces of RS465 CAN HAT are connected to the H and L lines of OBD-II interface. Furthermore, we write a Python-based program, through which we can send and receive bus data in real time.

Considering the potential threat that could be posed by sending attack signals to a running vehicle, we have built a CAN bus attack platform as shown in Fig. 4. The attack platform mainly consists of a real vehicle instrumentation system and a gateway. Further, we implement the attack on the platform with the built transceiver to verify the security vulnerability of the CAN bus and analyze the specific impact of different attacks on the bus.

Through in-depth analysis of the CAN protocol and attack tests based on attack platforms, there are some inherent security vulnerabilities in the design of the CAN bus. First of all, multi-master broadcast communication causes that as soon as one node is compromised by an attacker, the entire data leakage of the bus becomes possible. Additionally, it is vulnerable to a distributed denial of service(DDoS) attack which can block network traffic entirely. The lack of an effective encryption mechanism makes it easier for attackers to obtain private information or inject attacks. To make matters worse, CAN bus frames contain no sender and receiver information, nor timestamps. As a result, it is difficult for the receiver to verify the legitimacy of the sender and the authenticity of the frame. It greatly reduces the difficulty for the attacker to construct a frame considered legitimate.

FIGURE 4. The attack platform for the CAN bus.

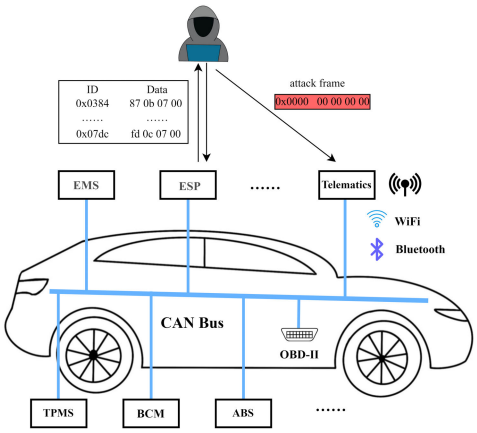

FIGURE 5. A general flow of an attack on the in-vehicle CAN bus.

A general flow of an attack on the in-vehicle CAN bus is shown in Fig. 5. In the first step, the hacker looks for the potential attack surface of the target vehicle and then uses wired or wireless attack surfaces to access the bus. In the second step, the hacker monitors the bus through a sniffer to obtain a large number of legitimate frames. In the third step, the hacker constructs malicious frames and injects them into the CAN bus of the target vehicle to carry out a specific attack.

C. ATTACK MODEL

In this section, we assume that the attacker has successfully hacked into the vehicle by directly connecting into the vehicle such as OBD-II or remotely connecting into the vehicle such as WiFi and Bluetooth. According to the

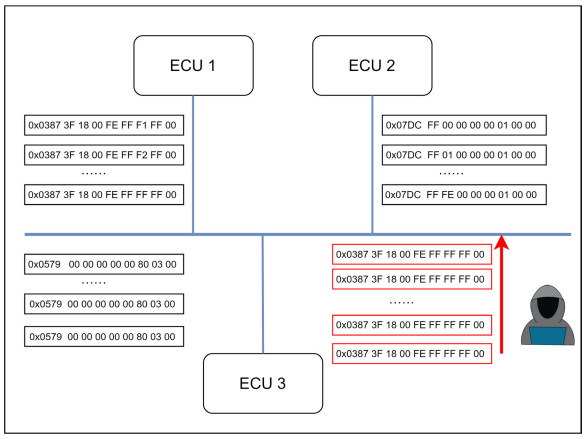

FIGURE 6. DDoS attack scenario.

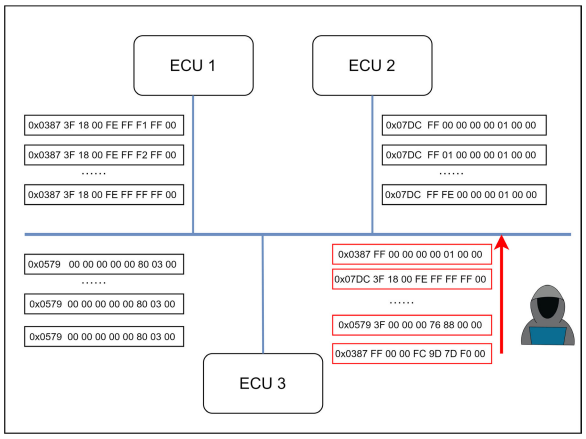

FIGURE 7. Fuzzy attack scenario.

attacker's knowledge of the attacked bus, attack intention, and attack technique, we classify the attack scenarios into DDoS attack, Fuzzy attack, Replay attack, and Masquerade attack. We implemented each of these attacks on the constructed attack platform to analyze their specific impact on the bus.

DDoS Attack: As described before, in the multi-master broadcast transmission mechanism of the can bus, the priority of frame sending depends on the value of the frame ID. Fig. 6 shows the DDoS attack scenario against the CAN bus. When multiple frames are sent to the bus at the same time, the one with the smallest ID value will get permission to send. DDoS attacks make the bus blocked by sending a large number of messages with low ID and high priority to the bus in a short period of time, so that valid messages sent by the legitimate ECU cannot obtain sending privileges, and eventually, the vehicle is paralyzed. Due to the lack of identification mechanism of message ID legitimacy in the vehicle, as long as the ID value is small enough, such as 0×00, this message can get higher sending privileges than all legitimate frames, even if this ID has not been used by any legitimate frame at all. Therefore, the attacker does not need to have a deep knowledge of the vehicle CAN bus operation to launch such a serious attack with very little prior knowledge.
Fuzzy Attack: The principle of a fuzzy attack is similar to a fuzzy test, in which a randomly generated message is sent to the bus to test the response of the bus in order to find vulnerabilities. Fig. 7 shows the fuzzy attack scenario against the CAN bus. In this type of attack, the attacker sends randomly generated mes-sages to the bus, and depending on how the bus reacts to different messages, the attacker is able to obtain more in-depth information, including the specific function represented by the ID and the specific meaning of the change in the value of the data field. Unlike DDoS attacks, the ID of the message in a fuzzy attack is randomly selected from legitimate IDs, and the value of each bit of the data field is randomly generated between 0 and 255. fuzzy attacks do not necessarily send a large number of messages in a short period of time, and the attacker may send random frames with fixed IDs at a small rate in order to try to find out the specific function of the ID.

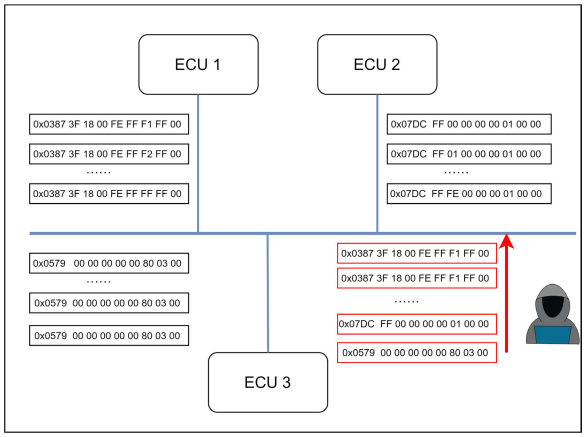

FIGURE 8. Replay attack scenario.

1) REPLAY ATTACKS

Fig. 8 shows the Replay attack scenario against the CAN bus. The attacker sends legitimate information to the bus that was previously captured from the bus at a particular time. A typical example of a replay attack is that hackers eavesdrop on the signals sent to the car by the owner's remote key, and unlock the car or even start the engine by replaying these messages. These attack frames are essentially legitimate frames that have been intercepted by the attacker, so it is difficult to detect by the frame format and content.

2) MASQUERADE ATTACKS

Masquerade attacks require the attacker to have in-depth knowledge of the vehicle CAN bus operation. Fig. 9 shows the masquerade attack scenario against the CAN bus. The attacker replaces specific normal frame data fields with target control information to cause certain ECUs to perform pre-determined actions, such as turning on the wipers, brakes, or steering. Further, the IDs and data fields of such attack frames are within the legal range and thus are difficult to detect.

IV. PROPOSED METHOD FOR INTRUSION DETECTION

According to the relevant protocols of the vehicle CAN bus such as ISO 11898 and our data collected from real vehicles,

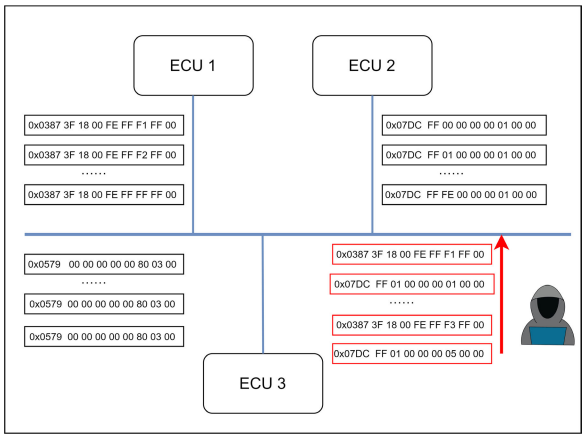

FIGURE 9. Masquerade attack scenario.

most of the messages of the vehicle CAN bus are periodic, i.e., frames are sent to the bus periodically at specified time intervals.

The specific IDs of the periodic messages have different values and functions depending on the make and model of the vehicle, and the format and specific meaning of their data fields vary. However, it is certain that the number of legal IDs used for each type of vehicle is much smaller than the number of IDs provided in the protocol(0x0000-0xFFFF), and the data fields for the different types of IDs are related to them. Considering the characteristics of the periodic sending of messages and the influence of specific environments on the periodicity, as well as the data fields associated with IDs, we propose an anomaly detection scheme based on a multi-observation Hidden Markov Model.

We define the anomaly detection problem as a multi-observation HMM model that determines the anomalous state of a frame by calculating the probability of frame occurrence at a specific moment based on the frame's timing, ID, and data domain. We define the bus traffic in time T as:

$$F = \{f_1, f_2, \ldots, f_T\}. \tag{1}$$

Then, we divide it into the ID sequence S and the data sequence O. S represents HMM state sequence in time T such that

$$S = \{ID_1, ID_2, \ldots, ID_T\}. \tag{2}$$

O represents the set of multiple HMM observation sequences in time T such that

$$O = \{O_1, O_2, \ldots, O_L\}, \tag{3}$$

where L is the number of bytes in the data field, and $O_l = \{Data_{l1}, Data_{l2}, \ldots, Data_{lT}\}$is the sequence of the ith byte of the data field in time T. And we can get the set of all possible states:

$$Q = \{q_1, q_2, \ldots, q_N\}, \tag{4}$$

where N is the number of all possible states. V is the set of all sets of possible observations:

$$V = \{V_1, V_2, \ldots, V_L\}, \tag{5}$$

where $V_l = \{v_{l1}, v_{l2}, \ldots, v_{lM_l}\}$ is the set of all possible observations of the observation sequence O_l and M_l is the number of all possible observations of V_l. The state transition probability matrix is defined as:

$$A = [a_{ij}]_{N\times N}, \tag{6}$$

where $a_{ij} = P(ID_{t+1} = q_i | ID_t = q_j)$. The observation probability matrix of the observation sequence O_l is defined as:

$$B_l = [b_j(k)]_{N\times M_l}, \tag{7}$$

where $b_j(k) = P(Data_{lt} = v_{ik} | ID_t = q_j)$. The initial probability distribution vector is defined as $\pi = (\pi_i)$, where $\pi_i = P(ID_1 = q_i)$.

Hence, a multi-observation HMM set can be defined as:

$$\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}, \tag{8}$$

where $\lambda_l = \{\pi, A, B_l\}$.

For λ_i, we define the forward probability at moment t:

$$\alpha_{lt}(j) = P\left(Data_{l1}, Data_{l2}, \ldots, Data_{lt}, ID_t = q_j | \lambda_l\right). \tag{9}$$

Similarly, we define the backward probability at moment t:

$$\beta_{lt}(j) = P\left(Data_{l(t+1)}, Data_{l(t+2)}, \ldots, Data_{lT} | ID_t = q_j, \lambda_l\right). \tag{10}$$

Given the model λ_l and the observation O_l, the probability of being in state q_j at moment t is shown below:

$$\gamma_{lt}(j) = P\left(ID_t = q_j | \lambda_l, O_l\right) = \frac{\alpha_{lt}(j)\,\beta_{lt}(j)}{\sum_{k=1}^{N} \alpha_{lt}(k)\,\beta_{lt}(k)}. \tag{11}$$

Specifically, our overall solution comprises three parts: the data preparation, the training module, and the detection module. In the data preparation process, we first obtain a long-term flow of data frames from the vehicle, keeping the ID, data field, and timestamp of each information frame. Further, the information is converted to decimal form. As shown in Fig. 10, the training module aims to establish HMMs and measurement thresholds depending on pre-processed data and an initial time window that rely on the captured vehicle's normal flow. In the training module, the preprocessed normal flow F is fed into the system to compute a multi-observation HMMs. Subsequently, we compute $\gamma_i = \{\gamma_{1i}, \gamma_{2i}, \ldots, \gamma_{Li}\}$ for each frame f_i in F using the obtained $threshold = \{threshold_1, threshold_2, \ldots, threshold_L\}$, which are used as the baseline for the detection module. Algorithm 1 shows the exact process of acquisition of HMMs and thresholds.

In the detection phase, we calculate γ_W of detected *frame*, where $\gamma_W = \{\gamma_{1W}, \gamma_{2W}, \ldots, \gamma_{LW}\}$. Algorithm 2 details the process of anomaly detection by comparing the threshold and γ_W. If any γ_{lW} is not within the range of the corresponding $threshold_l$, *frame* will be classified as an abnormal frame.The training and detection phases are repeated in multiple time windows W until the optimal time window W is determined.

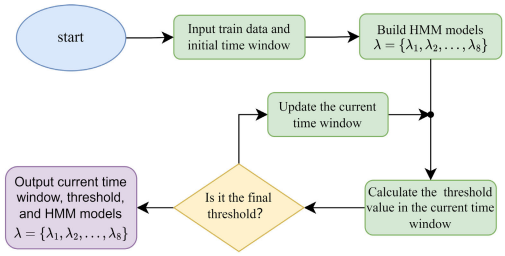

FIGURE 10. Training process.

Algorithm 1 Establish HMMs and Initial Thresholds

Input: *Frame* (list of CAN frame), W (Initial time window)

Initialization:*threshold*
$N \leftarrow$ CAN ID types
$L \leftarrow$ Maximum number of bytes in the CAN data field
$M = \{M_1, M_2, \ldots, M_L\} \leftarrow$ Set of data field types per byte
for $1 \le i \le N$ **do**
  $\pi_i \leftarrow$ Frequency of ID_i
end for
for $1 \le i \le N$ **do**
  for $1 \le j \le N$ **do**
    $A_{ij} \leftarrow$ Number of frequencies transferred from ID_i to ID_j
    $a_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}$
  end for
end for
for $1 \le l \le L$ **do**
  for $1 \le j \le N$ **do**
    for $1 \le k \le M_l$ **do**
      $B_{ljk} \leftarrow$ Number of frequencies ID_j and $o_l = k$
      $b_{ljk} = \frac{B_{jk}}{\sum_{k=1}^{M_l} B_{jk}}$
    end for
  end for
  $\lambda_l = \{\pi, A, B_l\}$
end for
for $1 \le i \le lenthofFrame$ **do**
  for $1 \le l \le L$ **do**
    $threshold\,[i]\,[l] \leftarrow \gamma_{lW}\,(j)$ of $frame_i$
  end for
end for

Output: HMMs $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$, *threshold*

Algorithm 2 Anomaly Detection Module

Input: *Frame* (list of CAN frame to be detected), W (Initial time window), HMMs $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_L\}$, *threshold*

for $1 \le i \le$length of *Frame* **do**
  calculate γ_W of $frame_i$
  for $1 \le l \le L$ **do**
    if $threshold\,[:]\,[l]\,MIN \le \gamma_{lW} \le threshold\,[:]\,[l]\,MAX$ **then**
      $frame_i$ is noraml
    else
      $frame_i$ is abnormal
    end if
  end for
end for

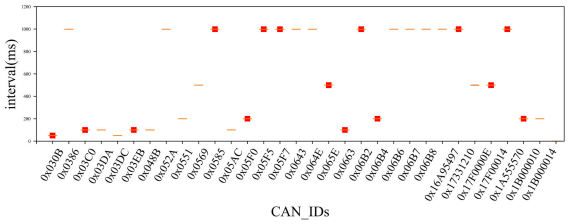

FIGURE 11. Time interval appeared on the bus of the 31 CAN IDs.

V. EXPERIMENT AND RESULTS

A. EXPERIMENTAL SETTINGS AND PERFORMANCE METRICS

Normal data in the experiment were collected from a car driving normally for half an hour during non-peak hours in an urban area. The attack scenarios in the experiment are generated based on the built attack platform. Considering the richness and completeness of the test data, we performed various maneuvers on the vehicle during driving, including acceleration and deceleration, steering, raising and lowering windows, etc. Fig. 11 depicts the time interval that appeared on the bus of the 31 CAN IDs collected from the experimental vehicle, from which a significant periodicity can be observed. As can be seen from the graph, some IDs, such as 0x0386, fluctuate at intervals of even less than 1ms; some IDs, such as 0x030B, appear to have larger intervals but do not exceed 10ms. Considering that CAN bus anomaly detection is a typical binary classification problem, in the following experimental evaluation we classify the detected data into true positive (TP), false positive (FP), true negative (TN), and false negative (FN).

Based on the above metrics, we further evaluate the performance of the proposed algorithm using Accuracy, Precision, Recall, and F1-Score.

Recall, also called True Positive Rate, refers to the ratio of correctly detected attack frames to the actual attack frames, denoted by

$$Recall = \frac{TP}{TP + FN}. \tag{12}$$

Accuracy refers to the ratio of correct detection to all detection, denoted by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{13}$$

Precision refers to the ratio of correctly detected attack frames to the total detected attack frames, denoted by

$$Precision = \frac{TP}{TP + FP}. \tag{14}$$

F1 score is the weighted average of Precision and Recall, denoted by

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}. \quad (15)$$

In order to demonstrate the applicability and generalisability of the proposed scenario, four attack scenarios were defined in our tests. In the construction of the attack scenarios, two assumptions are considered. On the one hand, we assume that the attacker has obtained bus access either wired or wirelessly, which allows them to freely receive and send data frames. On the other hand, we assume that the attacker has acquired the set of legitimate IDs so that the attack frame consists of the legitimate IDs and the modified data fields.

B. RESULTS

In order to fully estimate the performance of the proposed algorithm, we evaluated it on the basis of recall, accuracy, precision, and f1 for each of the four attack scenarios presented in Section III. During the experiments, we found that the data domain of some IDs remained constant but others varied. Therefore, the replay attack was split into reply_invariant and reply_variant with respect to these two types of frames.

In this experiment, the time window is considered as the size of a set of CAN messages for calculating γ_W. To investigate whether the size of the time window affects the detection performance, we evaluated the performance metrics of the proposed algorithm under five attack scenarios with different time windows. Time windows were defined as 3-36 and increasing on 3.

Fig. 12 shows the recall of the proposed algorithm for the five attack scenarios with different time windows. In summary, the proposed algorithm achieves the best recall when the time window is 18. For DDoS attacks, fuzzy attacks and reply_variant attacks, the recall rate can exceed 0.99 in all time windows. These three types of attack frames have large deviations from normal frames in ID domain or data domain, so they are easy to detect. For masquerade attacks and reply_invariant, the recall rate increases first and then decreases with the increase of the time window, and achieves the best performance when the time window is 18. This is due to the high similarity between these two types of attacks and normal attacks. When the window size is too small, many abnormal frames are mistaken as normal frames. When the window size is too large, some normal frames are wrongly detected as abnormal frames. In general, the proposed algorithm can effectively detect the existing attack frames and achieve a recall rate of more than 92%.

Fig. 13 shows the precision of the proposed algorithm for the five attack scenarios with different time windows. It can be seen that the change trend of precision over time window in different attack scenarios is similar to the recall. It can be seen that the change trend of precision over time window in different attack scenarios is similar to the recall value. For DDoS attacks, fuzzy attacks and reply_variant attacks, the

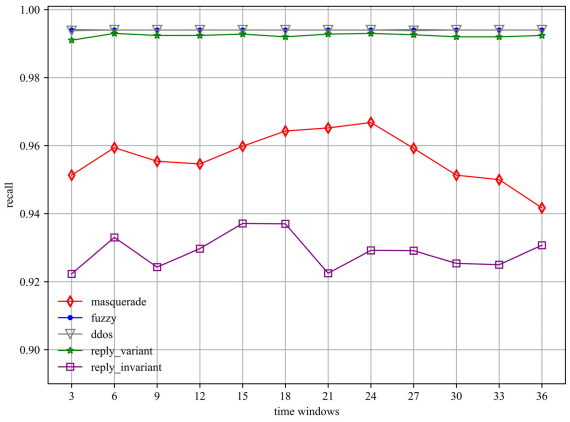

FIGURE 12. The recall of proposed algorithm with different window sizes.

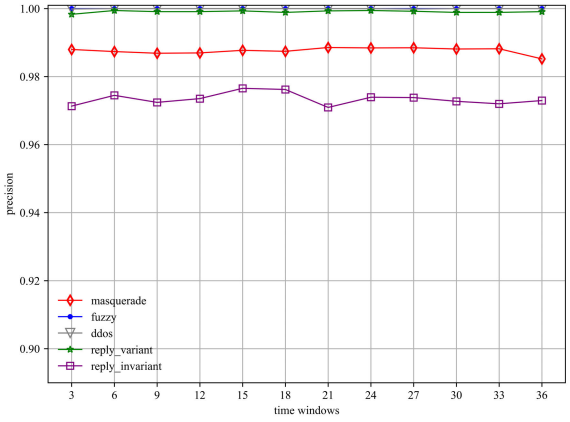

FIGURE 13. The precision of proposed algorithm with different window sizes.

precision can exceed 0.99 in all time windows. For masquerade attacks and reply_invariant attacks have a precision of more than 97% higher than recall, which means our algorithm has few false positives. Fig. 14 shows the accuracy of the proposed algorithm for the five attack scenarios with different time windows. Fig. 15 shows the f1 of the proposed algorithm for the five attack scenarios with different time windows. Similarly, the trend of f1 and accuracy over time windows in different attack scenarios is similar to the recall value. This means that the detection time required for the proposed algorithm is not very high. For DDoS attacks, fuzzy attacks and reply_variant attacks, both F1 and accuracy can approach 100%, which means that such attacks can be detected almost completely accurately. In terms of masquerade attacks and reply_invariant attacks, although the detection is very difficult, the proposed algorithm still shows very good performance. When the time window size is 18, both performance metrics exceed 95%.

It is clear that the proposed algorithm achieves recall, accuracy, precision and f1 close to 1 for the DDoS, fuzzy and reply_variant scenarios. And in reply_variant and masquerade scenarios, the performance metrics also exceed 93. It can be said that the proposed algorithm shows excellent detection in both simple and complex attack scenarios. Furthermore,

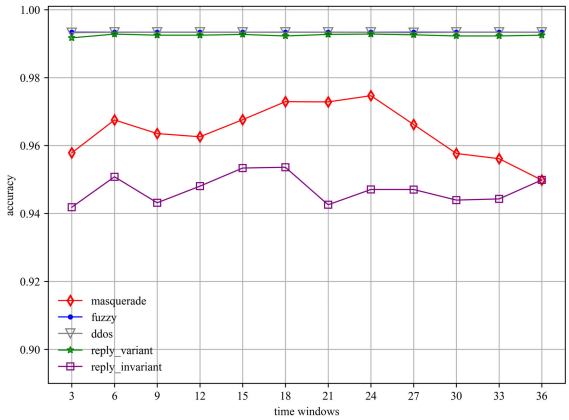

FIGURE 14. The accuracy of proposed algorithm with different window sizes.

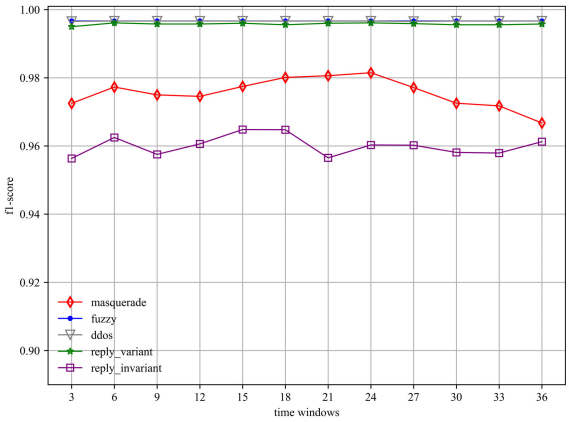

FIGURE 15. The f1 of proposed algorithm with different window sizes.

TABLE 1. Fuzzy attack.

Method	Recall	Precision	Accuracy	F1-score
KNN	0.88416667	0.99717444	0.992111	0.936926
DT	0.8758888	0.99961374	0.9595	0.933473
SVM(rbf)	0.998	0.9827033	0.994222	0.991234
CLAM	0.996964721	0.993947	0.99620584	0.995403
Our Proposed	0.99337014	0.999	0.994	0.996674

on balance, the most suitable time window size is between 18 and 24.

In this subsection, we compare the performance of the proposed approach with existing intrusion detection methods, such as k-nearest neighbor(KNN), decision-tree(DT), support vector machine(SVM) and CLAM [22]. In order to obtain convincing comparison results, we used the same training and test sets. Table 1 shows the test results of the above algorithms in the fuzzy attack scenario. It can be seen that SVM, CLAM and the proposed algorithm perform similarly under fuzzy attacks, but KNN and DT have lower recall. We speculate that frequent fuzzy attacks lead to deviations from normal values for the period of normal frames, resulting in high FN for KNN and DT.

Table 2 shows the test results of the above algorithms under the DDoS attack scenario. In fact, DDoS attack frames are less difficult to detect because they are usually constructed

TABLE 2. DDoS attack.

Method	Recall	Precision	Accuracy	F1-score
KNN	0.99337	0.999	0.994	0.99667404
DT	0.9993	0.998004	0.999333333	0.999000999
SVM(rbf)	0.998	0.982703	0.99422222	0.99123358
CLAM	1	0.99976	0.999531	0.99959881
Our Proposed	1	0.9981	0.999333333	0.999000999

TABLE 3. Reply_variant attack.

Method	Recall	Precision	Accuracy	F1-score
KNN	0.975359	0.931373	0.968666667	0.952858576
DT	0.875889	0.999614	0.9595	0.93347309
SVM(rbf)	0.985626	0.930233	0.971333333	0.957128614
CLAM	0.975155	0.9677419	0.99236641	0.979899497
Our Proposed	0.99337	0.999	0.994	0.99667404

TABLE 4. Masquerade and reply_invariant attack.

Method	Recall	Precision	Accuracy	F1-score
KNN	0.778455	0.813163	0.868666667	0.795430945
DT	0.672065	0.715517	0.804	0.693110647
SVM(rbf)	0.927984	0.742998	0.872666667	0.825251601
CLAM	0.90929183	0.96028014	0.94335937	0.9340407
Our Proposed	0.957837	0.987985	0.9513	0.97250461

by the ID with the highest priority. Clearly, all of the above methods have excellent detection results for DDoS attacks.

Table 3 shows the test results of the above algorithms in the reply_variant attack scenario. The proposed algorithm still performs well for all four metrics but the other algorithms are not able to achieve a good balance between recall and precision.

Table 4 shows the test results of the above algorithms under the reply_invariant attack scenario and the masquerade attack. These two attack scenarios have different objectives but similar attack forms. The detection performance of our proposed algorithm under these two attack scenarios is degraded due to the fact that the attack frames of reply_invariant and masquerade are very similar to the normal frames. Then, despite the degradation, our algorithm still manages to exceed 95% on all metrics, significantly outperforming the other algorithms. This is attributed to the fact that the proposed algorithm considers not only the dependencies between IDs, but also the correlation between IDs and data domains.

VI. CONCLUSION

In this paper, we proposed a multiple observation HMM-based, frame-by-frame CAN bus intrusion detection algorithm that does not depend on vehicle type and specific attack information in order to deal with security incidents caused by vehicle intelligence and networking. First, we establish multiple HMMs based on the ID domain and data domain of normal CAN bus traffic, and comprehensively calculate the existence probability of the normal frame as the detection threshold. When the presence probability of the frames to be detected does not fall within the normal threshold, they are determined to be anomalous. Then, in order to comprehensively evaluate the proposed algorithm, we establish four attack models based on the collected real vehicle data.

In the experimental simulation, we used four indicators: accuracy, precision, recall, and f1 to calculate the performance of the proposed algorithm in each attack scenario. Experimental results show that the proposed algorithm is superior to other classical frame-by-frame detection algorithms. In future work, we will strive to improve the performance of the intrusion detection model against higher-order attacks carried out by experienced hackers to make it close to the actual security protection of the CAN bus.

REFERENCES

[1] C. Chen, G. Yao, C. Wang, S. Goudos, and S. Wan, "Enhancing the robustness of object detection via 6G vehicular edge computing," *Digit. Commun. Netw.*, vol. 8, no. 6, pp. 923–931, 2022.
[2] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "VoltageIDS: Low-level communication characteristics for automotive intrusion detection system," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 2114–2129, Aug. 2018.
[3] S. Liu, J. Yu, X. Deng, and S. Wan, "FedCPF: An efficient-communication federated learning approach for vehicular edge computing in 6G communication networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1616–1629, Feb. 2022.
[4] C. Chen, L. Liu, S. Wan, X. Hui, and Q. Pei, "Data dissemination for industry 4.0 applications in Internet of Vehicles based on short-term traffic prediction," *ACM Trans. Internet Technol.*, vol. 22, no. 1, pp. 1–18, Feb. 2022.
[5] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, no. 91, pp. 1–91, 2015.
[6] Z. Cai, A. Wang, W. Zhang, M. Gruffke, and H. Schweppe, "0-days & mitigations: Roadways to exploit and secure connected BMW cars," *Black Hat USA*, vol. 2019, p. 39, Aug. 2019.
[7] Upstream. (May 10, 1991). *Upstream's 2022 Global Automotive Cybersecurity Report*. [Online]. Available: https://upstream.auto/2022report/
[8] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An efficient authentication scheme for intra-vehicular controller area network," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3107–3122, 2020.
[9] K. Agrawal, T. Alladi, A. Agrawal, V. Chamola, and A. Benslimane, "NovelADS: A novel anomaly detection system for intra-vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22596–22606, Nov. 2022.
[10] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 447–462.
[11] C.-W. Lin and A. Sangiovanni-Vincentelli, "Cyber-security for the controller area network (CAN) communication protocol," in *Proc. Int. Conf. Cyber Secur.*, Dec. 2012, pp. 1–7.
[12] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle CAN," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 993–1006, Sep. 2015.
[13] H. J. Jo and W. Choi, "A survey of attacks on controller area networks and corresponding countermeasures," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6123–6141, Jul. 2022.
[14] 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.
[15] U. Ezeobi, H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "Reverse engineering controller area network messages using unsupervised machine learning," *IEEE Consum. Electron. Mag.*, vol. 11, no. 1, pp. 50–56, Jan. 2022.
[16] M. L. Han, B. I. Kwak, and H. K. Kim, "Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2941–2956, 2021.
[17] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2016, pp. 130–139.
[18] 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.
[19] 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.
[20] G. Xie, L. T. Yang, Y. Yang, H. Luo, R. Li, and M. Alazab, "Threat analysis for automotive can networks: A GAN model-based intrusion detection technique," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4467–4477, Jul. 2021.
[21] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25469–25478, Dec. 2022.
[22] H. Sun, M. Chen, J. Weng, Z. Liu, and G. Geng, "Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10880–10893, Oct. 2021.
[23] H. Sun, M. Sun, J. Weng, and Z. Liu, "Analysis of ID sequences similarity using DTW in intrusion detection for CAN bus," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10426–10441, Oct. 2022.

CHEN DONG was born in Hebei, China, in 1995. She received the B.S. degree in electronic and information engineering from Beijing Information Science and Technology University, Beijing, China, in 2017, and the master's degree from the School of Electronic and Information Engineering, Beijing Jiaotong University, in 2019, where she is currently pursuing the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety. Her current research interests include in-vehicle networks, the Internet of Vehicles, the Internet of Things (IoT), privacy, and security.

HAO WU (Member, IEEE) received the Ph.D. degree in information and communication engineering from the Harbin Institute of Technology, in 2000. She is currently a Full Professor with the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University (BJTU), China. She has published more than 100 papers in international journals and conferences. Her research interests include intelligent transportation systems (ITSs), security and QoS issues in wireless networks (VANETs, MANETs, and WSNs), and the Internet of Things (IoT). She is a reviewer of IEEE major conferences and journals in wireless networks and security.

QINGYUAN LI was born in Hebei, China, in 1989. He received the B.S. degree from the School of Optoelectronic Information and Computer Engineering, University of Shanghai for Science and Technology, in 2012, and the master's degree from the School of Electronic and Information Engineering, Beijing Jiaotong University, in 2015, where he is currently pursuing the Ph.D. degree with the State Key Laboratory of Rail Traffic Control and Safety. His current research interests include vehicular networks, the Internet of Things (IoT), privacy, and security.