IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE VEHICULAR TECHNOLOGY SOCIETY SECTION

Received 29 May 2024, accepted 8 June 2024, date of publication 11 June 2024, date of current version 18 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412933

RESEARCH ARTICLE

Intrusion Detection System for In-Vehicle CAN-FD Bus ID Based on GAN Model

XU WANG[1], YIHU XU[1], YINAN XU[1], ZIYI WANG[2], AND YUJING WU[1], (Member, IEEE)
[1]College of Engineering, Yanbian University, Yanji 133002, China
[2]Yanbian University, Yanji 133002, China

Corresponding author: Yujing Wu (yjwu@ybu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62201492 and Grant 62161049, and in part by Jilin Province Science and Technology Development Plan Project under Grant 20220101141JC and Grant YDZJ202301ZYTS409.

ABSTRACT The growing abundance of electronic control units and peripheral devices loaded and connected to smart connected cars has resulted in a constant stream of cyber-attacks at various levels and dimensions. The CAN-FD bus plays a crucial role in smart connected cars. Currently, the majority of research efforts remain centered around the traditional CAN bus, with fewer studies addressing intrusion detection for the CAN-FD bus in smart connected vehicles. CAN-FD boasts a notable improvement in transmission speed, capable of reaching up to 8 Mbps compared to the 1 Mbps of the standard CAN bus. Utilizing intrusion detection systems designed for the CAN bus in high-speed CAN-FD applications could potentially hinder normal transmission and detection efficiency. Hence, we focus on the attack and intrusion detection of CAN-FD bus ID nodes to prevent unauthorized access and potential malicious attacks. We propose an ID intrusion detection system based on an improved Generative Adversarial Network (GAN) model, which consists of two parts: a data pre-processing module and a detection module. To apply the GAN model to the vehicle bus, we perform pre-processing of the bus data. We introduce the concept of dual discriminator to improve the detection rate and enable the handling of unknown attacks. With the output of dual discriminator, we can determine whether there are any anomalies in the detection data. First, we use a data pre-processing module to convert the ID segments of the automobile CAN-FD into binary image encoding to form ID images. Subsequently, these ID images are fed into an ID image feature extractor in the detection module to extract various auxiliary features. The discriminator receives these auxiliary features and calculates the probability of whether the received image is a normal ID image or not to determine the authenticity of the ID image. The experimental results show that the proposed intrusion detection system is able to detect a message within 0.15 ms, which fully meets the real-time detection requirements while the vehicle is in motion. The average detection rate of the proposed system for different types of attacks is 99.93%, which is an average of 1.2% improvement of the detection rate over the GIDS algorithm. The proposed system not only ensures the normal communication of CAN-FD bus but also realizes real-time and accurate intrusion detection.

INDEX TERMS Binary image coding, CAN-FD, dual discriminator, generative adversarial network (GAN), ID image feature extractor, network security.

I. INTRODUCTION

According to a recent report published in 2022 by Counterpoint Research, a market research organization, connected car sales would increase globally by 12% year-on-year, with a share of more than 50% in total car sales. As such, smart connected cars have become mainstream in the future automotive industry [1]. These are equipped with the functions of information sharing, complex environment perception, intelligent decision-making, and automated collaboration, which can realize an efficient, safe, comfortable, and energy-saving driving environment.

The interior of such cars integrates many electronic control units and devices, including systems for driver assistance,

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

entertainment and information services, as well as vehicle diagnostics. Real-time data sharing and data interaction between the interior and exterior of the vehicle during driving is realized through technologies such as cloud computing, artificial intelligence, and V2X (vehicle to everything) communication [2]. To work together, these systems need to be interconnected through various vehicle bus networks, yielding a complex network ecosystem that must be able to transfer large amounts of data in real time and at high speeds. For example, the driver assistance system utilizes the speed data from the control system to realize intelligent cruise function, to ensure that the vehicle drives stably in the correct lane by capturing the road condition data from the information system with the aid of a high-precision camera. At the same time, the vehicle information display needs to interact with the entertainment system via the bus, displaying data such as volume adjustment and media switching in real time.

With the steady rise in the number of electronic control units and devices integrated in smart connected vehicles, the security risks these vehicles face have also risen. Thus, not only we need to focus on the safety of the internal bus but also consider the various safety hazards that may arise from the various smart devices connected to the outside of the vehicle. This complex and diverse state of connectivity also brings a series of potential security issues. For example, an attacker can use these vulnerabilities to remotely control the vehicle's multimedia system, attack the controller to modify its firmware and gain the right to remotely send commands to the bus, realizing the purpose of remotely controlling the power system and braking system. Hackers can invade the vehicle's internal network through the vehicle bus without the user's knowledge and perform dangerous behaviors such as lowering or increasing the vehicle's speed, shutting down the vehicle's engine, braking suddenly, or disabling the brakes, which may trigger multi-layered security problems [3].

The CAN-FD bus in connected vehicles carries critical control information in areas such as electronic control units and anti-lock braking systems, as well as instrumentation control and anti-theft systems. However, precisely because of its importance, the CAN-FD bus faces challenges in terms of security; it lacks adequate security measures and is therefore vulnerable to malicious attacks. An attacker can modem the commands from the TSP (extended communication module), parse them into the CAN-FD bus protocol and forward them to the bus. Once the attackers have succeeded in getting inside the unsecured in-vehicle bus, they can disguise themselves as legitimate node IDs, send fake packets, and record and replay existing bus packets to execute replay attacks. In addition, they can also mislead the vehicle system by sending false status or data information, leading to wrong decisions, among other issues [2].

The CAN-FD bus, as the core communication medium of automotive networks, requires efficient intrusion detection systems to prevent potential attacks. Traditional intrusion detection methods usually rely on rules and statistical information, but these methods may not be effective in capturing new and sophisticated attack patterns. Failure to capture attacks in a timely manner may lead to unforeseen consequences on a connected vehicle bus with high-speed and real-time data transmission. Therefore, it is necessary to develop an intrusion detection system that can instantly recognize and prevent potential security threats and ensure the stability and security of vehicle systems to protect the vehicle and its passengers from potential attacks. Therefore, it is crucial to ensure the security of automotive systems to prevent potential malicious intrusions and dangerous behaviors. To ensure the widespread use and safe development of smart connected cars, meeting the corresponding security and reliability requirements of the vehicle bus is a key issue that needs to be addressed [4].

Current intrusion detection systems primarily target the CAN bus, leaving the CAN-FD bus inadequately addressed. However, the disparities in data transmission rates and segment lengths between these two buses suggest that algorithms designed for the CAN bus may not seamlessly apply to the CAN-FD bus. This mismatch could potentially disrupt its normal transmission and detection speed, resulting in less effective outcomes. Hence, there is an urgent need for an intrusion detection system precisely tailored to the transmission rate and detection efficiency characteristic of the CAN-FD bus.

In order to meet the requirements of real-time, high-speed and accuracy of intrusion detection in smart connected vehicles, we propose a CAN-FD bus ID intrusion detection system. By designing an optimized GAN model, we can improve the ID intrusion detection rate and detection speed. Chapter 2 presents a review of CAN-FD bus and conventional GAN models as well as the types of attacks and detection methods. Chapters 3 describes our proposed intrusion detection system, and Chapter 4 discusses the experimentation environment and the analysis of results.

II. RELEVANT BACKGROUND

In this section, we first introduce the application and connection methods of CAN-FD bus nodes in vehicular networks, along with details regarding the frame format and ID arbitration mechanism of the bus. Subsequently, we list several common types of ID attacks and present corresponding attack detection algorithms. Finally, we briefly introduce the structure and principles of the basic GAN model.

A. INTRODUCTION TO CAN-FD

Connected vehicles consist of bus networks such as CAN-FD (CAN) / LIN / Flexray / Ethernet [5]. As shown in Fig. 1, the CAN-FD bus (including CAN low-speed and high-speed) is used for vehicle diagnostics, remote monitoring and communication with other vehicles and infrastructures to support the development of Intelligent Transportation Systems (ITS). In addition, it is responsible for functions such as instrumentation control and anti-theft systems. Connected cars enable rich in-vehicle functionality through multi-network convergence technologies but consequently

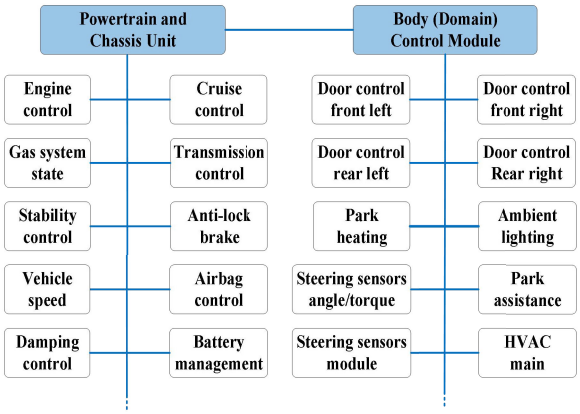

FIGURE 1. CAN-FD network connection diagram.

increase the number of interfaces for accessing the vehicle's internal and external networks [6].

However, these interfaces can also become the entrance target for malicious attacks. For example, when the in-vehicle electronic control system is connected to an external network system such as WiFi, Bluetooth, OBD II network tester, cellular network, etc., the hacker can easily steal the information of the in-vehicle bus network and invade the in-vehicle CAN-FD bus by remote control, thus tampering with the control information of important controllers such as engine, throttle, brake, steering wheel, and others [7].

The CAN-FD carrying important information typically adopts two frame formats: base format and extended format, as illustrated in Fig. 2. In base format the arbitration field consists of the base identifier and the r1 bit. The base identifier is 11 bits long. In extended format the arbitration field consists of the extended identifier, the SRR bit, the IDE bit, and the r1 bit. The extended identifier consists of two sections, the first section is the base identifier (11 bits long), the second section is the identifier extension (18 bits long).

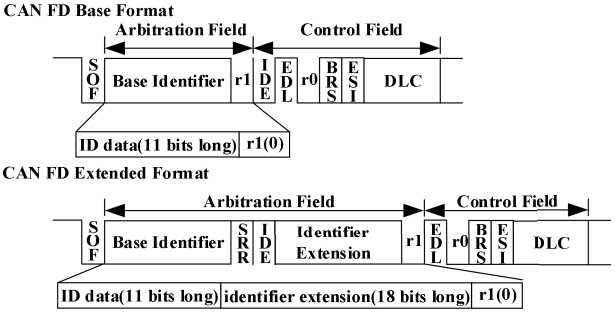

FIGURE 2. CAN-FD base format and extended format frames.

With the advancement of smart connected vehicle technology, the proliferation of Electronic Control Units (ECUs) is also notable, with current smart vehicles boasting up to approximately 120 ECUs. The identification (ID) assigned to each ECU is typically determined by the vehicle manufacturer and system designer [8]. Each ID message serves a specific communication purpose; for instance, as illustrated in Table 1, ECUs such as the Anti-lock Braking System (ABS) (ID 0508H) and Traction Control System (TCS) (ID 0153H) are capable of receiving messages transmitted by the Engine Management System (EMS) (ID 0329H).

TABLE 1. Message transmission and reception by ECU units with ID.

Transmit ECU and ID data	Receive ECU and ID data
EMS (Engine Management System) ID: 0329H	ABS (Anti Brake System) ID: 0580H TCS (Traction Control System) ID: 0153H
TCS (Traction Control System) ID: 0153H	ACC (Advanced Cruise Control System) ID: 0430H
TCU (Transmission Control Unit) ID: 0440H	ESP (Electronic Stability Program) ID: 0580H

Each ID can encompass data signals including vehicle speed, engine speed, frictional torque, accelerator pedal position, etc. Based on the 11-bit identifier frame format, several thousand ID data configurations can be generated [9]. The size of the ID data determines the order of how the data is sent on the bus as well as the location of the data to be sent, the send repetition rate, etc. According to ISO 11898-2, the lower the value of the ID, the higher the priority of the corresponding message [10].

In CAN-FD, the message sender continuously listens to the signals on the bus, and if it detects that a higher-priority message is being transmitted on the bus while sending a message, the sender will stop sending immediately and wait for the current message transmission to complete before attempting to send it again. As shown in Figure 3, when three nodes attempt to transmit messages simultaneously, the node with the highest priority, Node C (ID 404H), transmits the message onto the bus. Meanwhile, Node A (ID 407H) and Node B (ID 405H) must wait for the bus to become idle before they can transmit their messages.

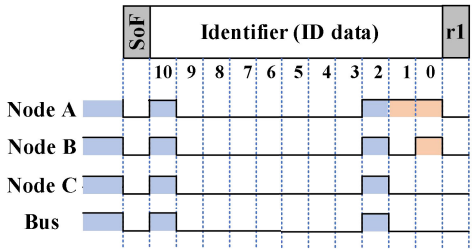

FIGURE 3. ID arbitration condition in CAN-FD protocol.

This mechanism ensures that lower-priority message senders do not interfere with the transmission of higher priority messages, thus maintaining the orderly transmission of messages on the bus [11]. The broadcast mechanism of CAN-FD presents a security risk exploitable by attackers to

transmit high-frequency and high-priority messages, potentially causing bus congestion and system paralysis. Hence, it is crucial to implement an intrusion detection system to monitor bus activity and detect any anomalous behavior, ensuring the secure operation of vehicle systems [12]. Consequently, this paper focuses solely on exploring the ID format within the CAN-FD Base Format.

B. ID ATTACK TYPES AND DETECTION ALGORITHMS

The security implications of external connections were not adequately addressed in the original design of the automotive CAN-FD bus. Consequently, the vehicle bus may lack effective self-protection and defense mechanisms in the event of attacks on external devices or gateways. The data and ID segments of the CAN-FD bus are particularly vulnerable, making them easy targets for attacks. Various types of attacks against the CAN-FD bus can be categorized as follows:

1) FUZZY ATTACK

Fuzzy attacks are a type of injection attack, a form of attack that injects a large amount of random or invalid data into the input data. An attacker uses compromised or unauthorized nodes to send malicious messages to the bus to cause confusion. Moreover, the attacker can easily create and inject diagnostic data, which can lead to loss of control of the on-board unit or even cause serious accidents such as sudden acceleration or stalling of the car [14].

2) DOS ATTACK

The attacker sends a large amount of unknown and known anomalous data over a certain period to reduce the utilization of the bus, thus can prevent it from receiving or processing information from legitimate users, resulting in an abnormal bus load.

3) PRM/GEAR ATTACK

An attacker can maliciously tamper with a vehicle's engine speed (RPM) and transmission (Gear) information through methods such as modifying ECU data, interfering with sensors, or executing a network intrusion. Should the attack induce abnormal engine operation, it could compromise both engine performance and durability. Such attacks have the potential to mislead drivers, prompting inappropriate driving behavior, and simultaneously pose a significant threat to vehicle performance and safety.

Currently, the following intrusion detection methods are available for the aforementioned attacks. Khan et al. proposed a data detection model based on LSTM (Long Short-Term Memory) neural network for detecting intrusion data in the braking system. They created disinformation and attack datasets, applied LSTM for attack detection, and achieved a detection rate of 87% [15]. Jedh et al. presented a data injection attack detection scheme that is independent of the ECU ID and applies thresholds, change point detection, long short-term memory (LSTM), and recurrent neural network (RNN, Recurrent Neural Network) to detect and predict malicious data injected into the CAN bus. When the threshold method is used, the detection rate reaches 97.32% and the detection speed is 2.5 ms [16]. Lee et al. proposed a detection system (OTIDS (Offset Time Interval Detection System)) for CAN bus intrusion attacks by analyzing the time interval between the request message and the corresponding message of a message on the CAN bus to measure the response performance of known nodes [17]. Moreover, the LOF (Local Outlier Factor) intrusion detection method was presented for identifying local outliers or outliers in a dataset [13].

A model known as GIDS (GAN-based Intrusion Detection System) was developed for intrusion detection, achieving an accuracy of 99.5% [18]. Another method, called Wavelet-Based Intrusion Detection System (WINDS), utilizes wavelet analysis of CAN communication transmission patterns to locate behavioral changes in the network, achieving 94% accuracy [19]. Additionally, the Histogram-Based Intrusion Detection and Filtering framework (H-IDFS) assembles CAN packets into windows and computes histograms for intrusion detection, boasting an accuracy of 99.4% [20].

Another hybrid IDS framework combines rule-based and machine-learning-based approaches for efficient and accurate intrusion detection. Notably, binary convolutional neural network (BCNN) based IDS strikes a balance between accuracy and speed [21]. There is also a lightweight neural network based CAN anomaly detection method which reduces the operation time while ensuring the detection accuracy. The method is designed with a redundant neuron screening method and a model compression algorithm with layer-by-layer neuron pruning (LNN) [22]. There are also systems for intrusion detection for LIN and Ethernet, some of which utilize advanced Deep Convolutional Neural Network (DCNN) models. These systems utilize the powerful learning capabilities and efficient feature extraction of DCNN models to effectively detect and identify various network attacks [23].

For CAN-FD networks, the Physical Semantics-Enhanced Anomaly Detection (PSEAD) approach is employed to fully leverage the physical features within the contextual information, aiming to enhance the accuracy and interpretability of anomaly detection [24]. Moreover, a lightweight GRU-based system is utilized, which transforms CAN data frames into feature vectors through data preprocessing and feature extraction. Subsequently, a neural network model with GRU as the primary hidden layer is employed for classification and optimization purposes in intrusion detection [25].

Table 2 illustrates the correlation between existing intrusion detection algorithms and the types of attacks they can detect, along with the network types they can monitor. For example, the LSTM, WINDS, and H-IDFS algorithms are capable of detecting Fuzzy and DoS attacks, while the OTIDS algorithm can detect DoS, PRM, and Gear attacks. On the other hand, the GIDS algorithm can detect four types of attacks.

TABLE 2. Comparison table of attack types and detection algorithms.

	Fuzzy Attack	DoS Attack	PRM Attack	Gear Attack	Network type
LSTM	√	√			CAN
OTIDS		√	√	√	CAN
GIDS	√	√	√	√	CAN
WINDS	√	√			CAN
H-IDFS	√	√			CAN
BCNN	√	√			CAN
DCNN	√	√			LIN, Ethernet
PSEAD	√	√			CAN-FD

C. INTRODUCTION TO THE GAN MODEL

The intrusion detection system proposed in this paper is based on a GAN (Generative Adversarial Network) model, hence a brief introduction is warranted here.

A GAN is a deep learning model that generates high-quality data samples by leveraging two neural networks in a competitive manner: a generator network and a discriminator network (as shown in Fig. 4) [26]. During the training process, these two networks play against each other to continuously improve their respective performance and eventually generate realistic and diverse data.

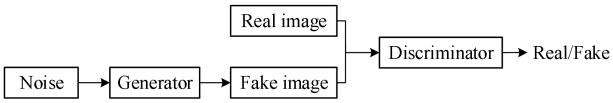

FIGURE 4. Generation of a countermeasure network model.

GANs have a certain generalization ability to detect previously unknown intrusion patterns because they learn the intrinsic distribution of the data. This makes GAN models quite promising for dealing with novel attacks. A generator learns the patterns of a normal data stream and attempts to generate data with similar patterns. The task of the discriminator is to distinguish between generated data and real data. If the generated data is discriminated as anomalous, an intrusion or attack may have occurred.

Currently, one of the areas with the greatest interest to GAN researchers is image and vision [22]. GAN have also been introduced into research in other AI subfields, including speech and language processing, malware detection, and chess game programs. In recent years, some neural network-based intrusion detection methods have also emerged, such as data detection models based on LSTM neural networks. Besides, there are a number of injections that use RNN neural networks to detect and predict malicious messages in CAN buses.

III. CAN-FD BUS IDS DESIGN BASED ON GAN

In pursuit of real-time, fast, and accurate detection algorithms, we propose an intrusion detection system for CAN-FD bus IDs. Through optimization and design of the GAN model, we aim to enhance the accuracy and speed of ID intrusion detection.

Our proposed intrusion detection system consists of data pre-processing and detection modules, as illustrated in Fig. 5. The data pre-processing module is responsible for converting the ID messages from the automotive CAN-FD bus into ID images. Meanwhile, the detection module employs a generative adversarial network algorithm to detect anomalies within these ID images.

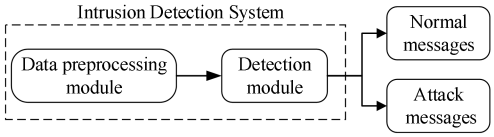

FIGURE 5. Overall system module diagram.

A. DATA PRE-PROCESSING MODULE

Since the messages transmitted by the car are not inherently images, we adopt a binary image encoding approach to transform these messages into image format, aligning with the requirements of our proposed intrusion detection system.

The pre-processing method utilized by the GIDS algorithm involves employing a one-hot vector approach, where each CAN ID is mapped to a 3 by 16 hexadecimal mapping matrix. For instance, as depicted in Figure 6, in the hexadecimal ID data '15A', '1' is mapped to position 1 in the first row of the matrix, '5' is mapped to position 5 in the second row, and 'A' is mapped to position 'a' in the third row. These three positions are shaded in gray, while the remainder remain unshaded.

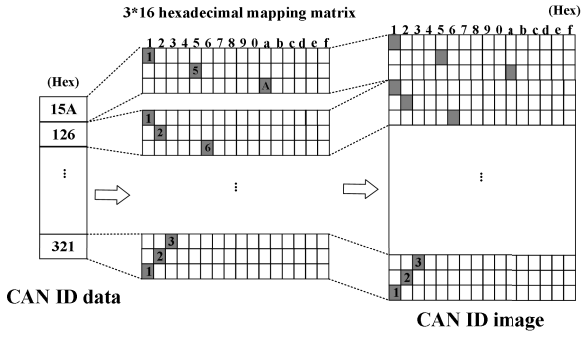

FIGURE 6. One-hot-vector image encoding.

Although the location seeking mode may provide more information in some cases, it also uses a lot of memory and may increase the time required for intrusion detection.

To mitigate memory usage and enhance detection speed, we have proposed a binary image encoding algorithm. Using the ID data "1DC" as an example, the specific process is elaborated in detail with reference to Fig. 7.

First, the hexadecimal ID data '1DC' is converted into the binary number '000111011100'. Then, these binary

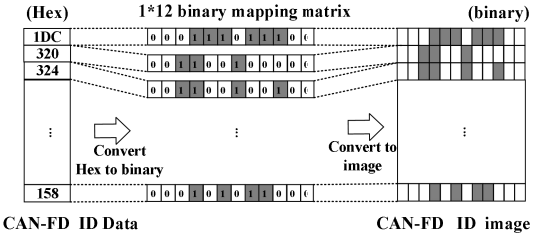

FIGURE 7. Binary image encoding.

values are populated into a 1 by 12 matrix where positions corresponding to '1' are depicted in black and positions corresponding to '0' are depicted in white. If the length of the ID exceeds 12 bits, the matrix length can be extended accordingly. Sequentially, we applied the binary image encoding algorithm to convert several ID data, as depicted on the left side of Fig. 7, obtaining the ID image displayed on the right side of Fig. 7. Consequently, the data pre-processing module is capable of reducing memory consumption and increasing detection speed.

We tested the size of ID images used for training, including 32, 64 and 128 ID data as a group. We evaluated the detection speed and detection efficiency. Through testing, we found that using 64 ID data as a single ID image yields the best detection performance and speed.

B. DETECTION MODULE

Our improved GAN model efficiently extracts image features by enhancing the depth of the convolutional layers, thereby enhancing the accuracy of threat identification. We optimize the generator within the traditional GAN model, naming it the ID image feature extractor. Additionally, we introduce a dual discriminator design, which further improves the accuracy of the detection system.

1) ID IMAGE FEATURE EXTRACTOR

The ID image feature extractor is a convolutional neural network comprising four convolutional layers. Its primary objective is to extract relevant features from the ID images of CAN-FD. Each convolutional layer is followed by an activation layer and a batch normalization layer, which collectively contribute to enhancing the network's performance.

In Fig.8, Conv1 to Conv4 represent the four convolutional layers from left to right. The activation function used in these convolutional layers is ReLU. Additionally, Bn1 to Bn3 represent the three batch normalization layers from left to right, with the number of activated channels being the main parameter. The output of the convolutional neural network is then extracted as auxiliary features, denoted as C. Furthermore, this network has the capability to generate virtual images based on these auxiliary features. The parameters of each network layer are detailed in Table 3, encompassing specifications such as the number of input channels, the number of output channels, the size of the convolutional kernel, and the stride size.

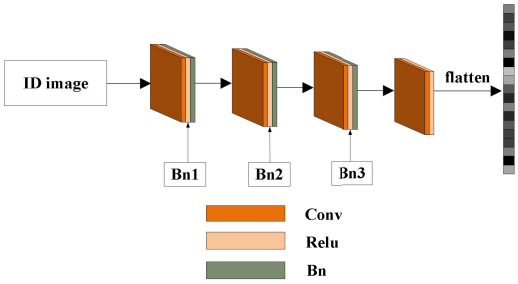

FIGURE 8. Network structure of the ID image feature extractor.

TABLE 3. Network parameters of each network layer.

layers	Convolutional Layer				activation layer	BN layer
	Input channels	Output channels	Convolution kernel size	Convolution step	--------	----
1	1	3	3×3	1	min(max(features, 0), 6)	3
2	3	6	3×3	1	min(max(features, 0), 6)	6
3	6	3	3×3	1	min(max(features, 0), 6)	3
4	3	1	3×3	1	min(max(features, 0), 6)	---

2) MODEL TRAINING PROCESS

To enhance the detection rate and effectively address unknown attacks, we introduce the concept of dual discriminators: the primary discriminator and the secondary discriminator.

a: PRIMARY DISCRIMINATOR

The primary discriminator receives two types of image inputs: normal ID images and attack ID images. Normal ID images are sourced from a dataset comprising typical driving vehicles. On the other hand, attack ID images are generated via simulation using CANoe software, encompassing DoS, fuzzy, and RPM/Gear attacks. Both categories of images undergo initial processing by a feature extractor to produce auxiliary feature C, as illustrated in Figure 9.

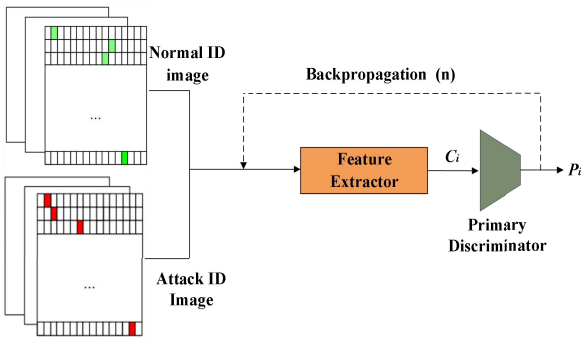

FIGURE 9. The process of primary discriminator training.

The primary discriminator is represented by the cross-entropy loss function L_{d1}, as illustrated in equation (1). In this equation, 'pi' represents the output value of the primary

discriminator during training, similar to a probabilistic prediction. ''yi'' comprises a pair of normal ID image and attack ID image as depicted in Figure 9. The variable 'N' signifies the overall count of samples, utilized to calculate the average cross-entropy.

$$L_{d1} = \frac{1}{N}\sum_{i} C_i = \frac{1}{N}\sum_{i} -[y_i log(p_i) + (1-y_i) log(1-p_i)] \tag{1}$$

The auxiliary feature C_i is inputted into the primary discriminator, which then generates the predicted value P_i. P_i represents the probability that the input is a real sample, with values ranging between 0 and 1, computed using the sigmoid function. The system enhances its learning capability and accuracy through multiple iterations (usually denoted as n times), thereby further improving the overall intrusion detection performance.

b: SECONDARY DISCRIMINATOR

During the training process of the secondary discriminator, we employ the generator to generate unknown ID data for the purpose of detecting unknown attacks. As depicted in Figure 10, the generator utilizes both normal ID images and noise images to produce unknown ID images. These generated images, along with the original ID images, are then inputted into the feature extractor.

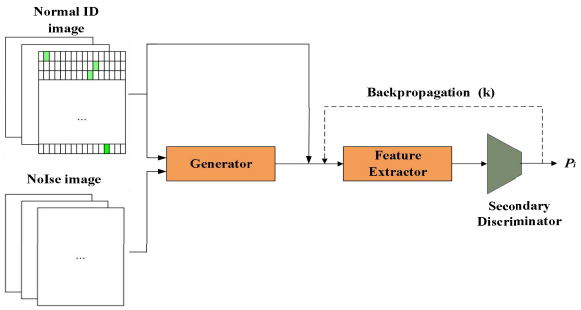

FIGURE 10. The process of secondary discriminator training.

To enhance detection efficiency, we adopted equations (2) and (3) as the loss functions for the secondary discriminator and generator. Unlike using the loss entropy function of the primary discriminator, which could prolong the overall training time and decrease the detection rate, this approach expedites the convergence process of the model. By guiding both the generator and discriminator towards the optimal solution, this loss function effectively addresses common training issues, such as gradient vanishing or explosion, ensuring training stability and reliability. As a result, we can obtain generated results more quickly with high accuracy and stability.

In equation (2), $D_2(x)$ represents the output of the discriminator for real data, while $D_2(G(z))$ represents the output of the discriminator for samples generated by the generator. The parameters 'a' and 'b' are utilized to determine the weights of the two terms in the loss function. By minimizing this loss function, the discriminator learns to effectively distinguish between real and generated data. The values of 'a' and 'b' are chosen appropriately based on experimental needs, with $E_{x\sim Pr}$ denoting the expected values for samples distributed from real data.

In equation (3), $D_2(G(x))$ represents the discriminator's output for the true data sample, and 'c' is the value of the fake data that the generator aims for the discriminator to accept. $E_{z\sim pg}$ denotes the expected value of the unknown data distribution. The number of iterations for the secondary discriminator is denoted as 'k'. By minimizing this loss function, the generator learns to generate samples that are closer to the true data distribution.

$$L_{d_2} = \min_{D_2} \frac{1}{2}E_{x\sim P_r}[D_2(x)-a]^2 + \frac{1}{2}E_{x\sim P_r}[D_2(G(z))-b]^2 \tag{2}$$

$$L_g = \min_{G} \frac{1}{2}E_{z\sim Pg}[D_2(G(x))-c]^2 \tag{3}$$

3) THE NETWORK ARCHITECTURE OF THE DISCRIMINATORS

The network architecture of the primary and secondary discriminators is depicted in Fig. 11. Both discriminators utilize three fully connected layers, receiving feature vectors from the previous layer as input and linearly combining these features with a set of weights. Subsequently, an activation function is applied to non-linearly transform the result of the linear combination. The network parameters of the discriminator are provided in Table 3.

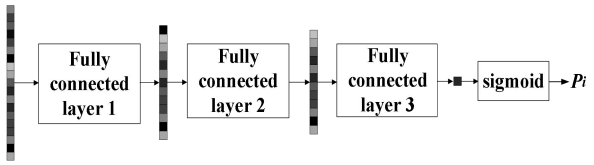

FIGURE 11. Network structure of discriminator.

We implement a sigmoid function to compute the probability (denoted as P_i) of whether the input data is anomalous or not. The range of the P_i value determines the filtering effect of the attacking image, with Pi ranging from 0 to 1. A P_i value closer to 0 indicates a more sensitive filtering. Therefore, it is essential to find a suitable P_i value to achieve the desired filtering effect.

The flowchart of the detection model is presented in Fig. 12. P_1 denotes a prediction value of the primary discriminator, and P_2 denotes a prediction value of the secondary discriminator.

We utilized over three million messages as test data to search for an appropriate Pi value. As depicted in Fig. 13, we achieved an average detection rate of 99.97% when the threshold value was set to 0.53. Setting the threshold above 0.53 may result in normal data being erroneously classified as attack data, while setting it below 0.53 may lead to attack data being incorrectly identified as normal data.

TABLE 4. Network parameters of the discriminator.

full connectivity layer 1		full connectivity layer 2		full connectivity layer3	
Input nodes	Output nodes	Input nodes	Output nodes	Input nodes	Output nodes
56 × 40	32 × 48	32 × 48	16 × 48	16 × 48	1

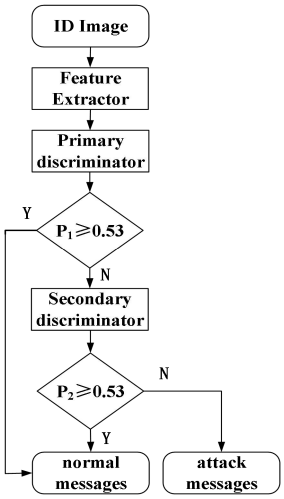

FIGURE 12. Flowchart of the detection system.

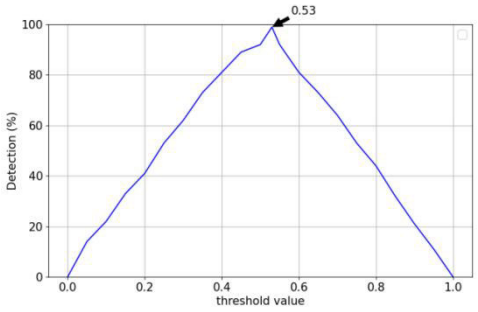

FIGURE 13. Threshold-detection rate relationship curve.

Therefore, we concluded that setting the threshold to 0.53 yields optimal detection efficiency.

IV. EXPERIMENTATION AND ANALYSIS

In this section, we will first describe the construction of the dataset, followed by an analysis of the performance of the training model and the detection results.

A. CONSTRUCTION OF THE DATASETS

The datasets used for the experiments include normal, attack and unknown data.

1) Normal ID data: The normal ID image dataset used in this paper is extracted from Honda and KIA cars in real time. As shown in Fig. 14, we extracted the CAN bus ID data after driving a certain distance through the OBD interface connected to the vehicle's CAN bus.

FIGURE 14. Vehicle bus data extraction via OBD II interface connection.

Among them, there are 39 ID data from Honda and Kia. We input the extracted CAN ID data into the CANoe software and simulated normal ID data by establishing a CAN-FD network topology.

2) Attack ID data: We first set up a real in-vehicle CAN-FD bus network on the CANoe simulation platform (as shown in Fig. 15). By configuring attack nodes, we simulated four different types of attacks.

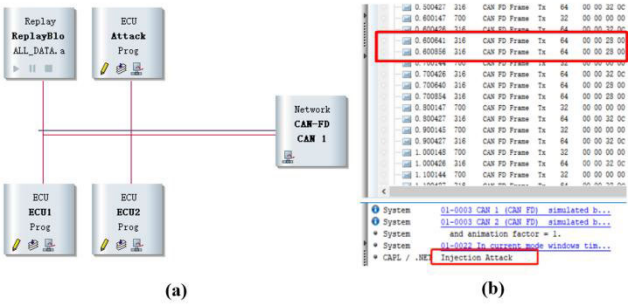

FIGURE 15. (a)CAN-FD network topology diagram; (b) Successful attack notification.

We simulated four types of attacks in CANoe software, namely, Fuzzy ID attack, DoS ID attack, and RPM/Gear ID attack. The definition of each type of attack is as follows:

Fuzzy ID Attack: this type of attack is implemented by injecting messages with forged random ID and data values every 0.5 ms. The CANoe simulation result of fuzzy ID attack is shown in Fig. 16. In Figure (a), the Fuzzy attack model is depicted, while (b) shows a trace screenshot transmitted by CANoe after simulating the Fuzzy attack. 1A0H, 41AH, and 188H represent normal IDs, which attackers can mimic to send malicious messages. Among them, 1A0H contains pedal and steering information, allowing attackers to manipulate

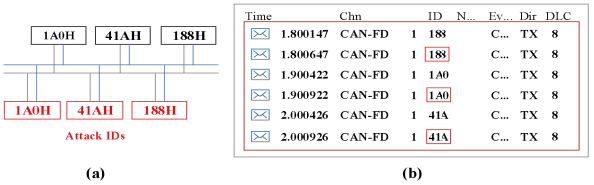

FIGURE 16. (a) Fuzzy attack topology diagram; (b) CANoe simulation result diagram of the Fuzzy attack.

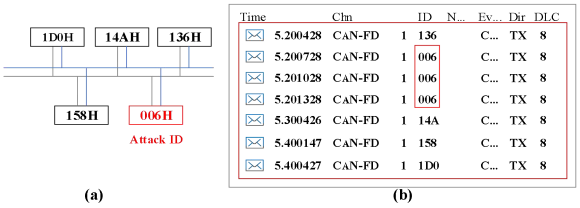

FIGURE 17. (a) DoS attack topology diagram; (b) CANoe simulation result diagram of the DoS attack.

data segment information arbitrarily, for instance, executing maneuvers such as sudden steering or emergency braking.

DoS ID attack: this type of attack injects high-priority messages at very short intervals.We extracted 39 IDs from vehicles of different models. Among them, the highest priority ID is 039H. Therefore, we set 006H as the attack ID for testing DoS attacks. A message with ID 0×006 is sent every 0.3 ms. The CANoe simulation results of the DoS ID attack are shown in Fig. 16. Among them, 1D0H, 14AH, 136H, and 158H are normal IDs, while the attack ID is 006H. According to the simulation results from CANoe, the attacker utilized the highest priority ID, 006H, to transmit messages, causing normal ID messages to remain in a waiting state, thereby impacting the transmission latency of critical data.

RPM/Gear ID Attack: this type of attack is implemented by injecting specific ID messages related to RPM/Gear information every 1 ms. The CANoe simulation results of the RPM/Gear ID attack are shown in Fig. 18. The RPM/Gear ID attack is performed by injecting specific ID messages related to RPM/Gear information every 1 ms. In Figure 18.(a), 039H, 136H, and 13AH represent normal IDs, while 316H and 43FH are attack IDs. Among them, 316H contains RPM information, while 43FH contains Gear information. Attackers exploit these attack IDs to arbitrarily tamper with information, thereby conducting malicious attacks.

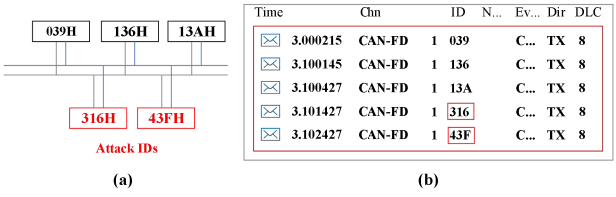

FIGURE 18. (a) RPM/Gear attack topology diagram; (b) CANoe simulation of RPM/Gear attack result diagram.

We generated attack ID datasets for the aforementioned four attacks, and their specific statistics are presented in Table 5. As an example, the DoS attack dataset consists of a total of 3,769,746 frame messages, comprising 3,754,392 normal data messages and 15,354 attack data messages.

B. HARDWARE EXPERIMENTATION AND ANALYSIS

1) HARDWARE AND SOFTWARE TESTING ENVIRONMENT

The intrusion detection system testing environment consists of Pycharm software, FinalShell Service Manager with Raspberry Pi 3B+ development board and other test equipment. We used Pycharm to create the intrusion detection system model and uploaded the model to the Raspberry Pi 3B+ development board through the FinalShell service manager. The experimental environment is depicted in Fig. 19.

TABLE 5. Data type and size.

Attack type	Total frame	Normal data	Attack data
DoS attack	3,769,746	3,754,392	15,354
Fuzzy attack	3,907,532	3,887,530	20,002
RPM attack	4,463,828	4,432,458	31,370
Gear attack	4,264,180	4,235,708	28,472

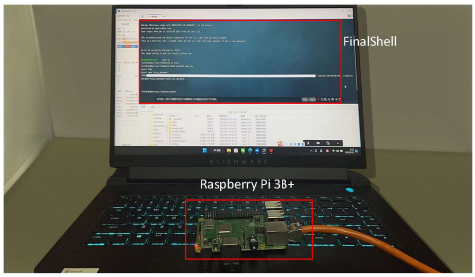

FIGURE 19. Integrated software and hardware experimental environment.

2) USING OPTIMIZER TO ENHANCE OVERALL PERFORMANCE

We conducted tests on three optimization algorithms, AdaGrad, RMSProp, and Adam, to determine the most optimal one for training the proposed intrusion detection model. After 50 training iterations, the results displayed in Figure 20 revealed that the Adam optimizer achieved the highest detection rate. Furthermore, we examined the loss variation of these three optimization algorithms, as depicted in Figure 21. The experimental outcomes indicated that the loss variation

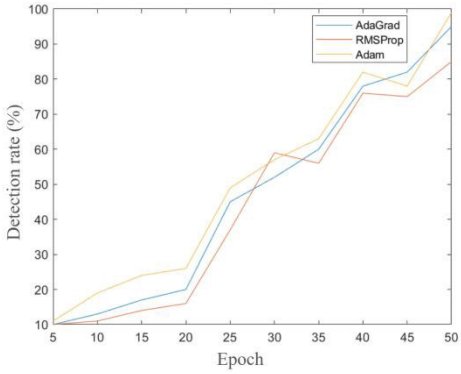

FIGURE 20. Comparison of system detection rates under different optimizers.

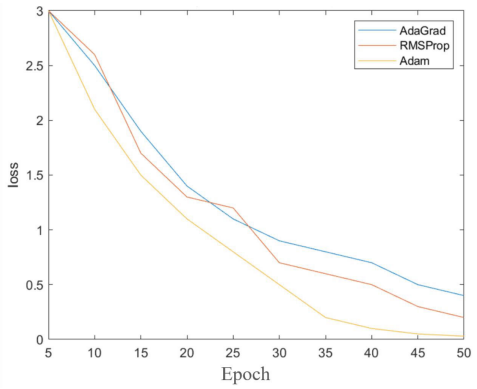

FIGURE 21. Comparison of loss function performance under different optimizers.

curve of the Adam optimizer exhibited smooth progression without significant fluctuations or noise. Initially, the loss function decreased rapidly during the early stages of training and gradually stabilized thereafter. Considering these factors, we selected Adam as the optimizer for the proposed intrusion detection model.

3) DETECTION PERFORMANCE OF PROPOSED INTRUSION DETECTION SYSTEM

Table 6 illustrates the detection efficacy of the proposed intrusion detection system against various attack types. Notably, the completion time for fuzzy attack intrusion detection is 9 minutes and 40 seconds, boasting a detection rate of a remarkable 99.931%, with an average of 6,702 messages being detected each second. Taking into account that the transmission time for each standard message in the automotive CAN-FD system is approximately 0.33 ms, our intrusion detection system can successfully accomplish detection in just 0.15 ms. Rapid detection of potential security threats is critical in preventing potential attacks. This entails swift responses to security threats and the potential to mitigate damage from potential attacks. This demonstrates that our system can conduct real-time intrusion detection on the bus network without impeding the vehicle's normal operation.

TABLE 6. Detection rate of proposed intrusion detection system.

Attack types	Attack ID Images (pcs)	Number of tests (pcs)	Detection rate (%)	Recall rate (%)
Fuzzy attack	20,002	19,988	99.93	99.82
DoS attack	15,354	15,324	99.81	99.74
RPM attack	31,370	31,295	99.76	99.70
Gear attack	28,472	28,435	99.87	99.82

As shown in Table 7, we compare the proposed algorithm with three different intrusion detection algorithms. When compared to the GIDS algorithm of the same type, the accuracy is improved by 0.2%, 0.43% and 0.7% when targeting

TABLE 7. Comparative analysis of intrusion detection rates.

Attack	Fuzzy attack	DoS attack	RPM attack	Gear attack
Proposed system	99.93%	99.81%	99.76%	99.87%
GIDS	99.5%	99.6%	99.0%	96.5%
WINDS	87.8%	94.9%	99.2%	98.8%
H-IDFS	95.1%	97.2%	99.1%	98.9%

DoS attack, Fuzzy attack and PRM attack, respectively. The most significant improvement occurs in the detection of the Gear attack, where the accuracy is improved by 3.5%. The hardware detection of fuzzy attacks using the GIDS algorithm on a Raspberry Pi took 13 minutes and 32 seconds. The experimental results reveal that our algorithm is able to detect intrusion data faster and more accurately, outperforming the GIDS algorithm in terms of hardware execution efficiency.

We also simulated the WINDS and H-IDFS algorithms, which were not GAN models, and compared them with our proposed algorithm. Our algorithm achieved a 12% improvement in detecting fuzzy attacks compared to WINDS. Additionally, for the other three types of attacks, our average detection efficiency increased by 2%.

V. CONCLUSION

With the advancement of connectivity and the increasing complexity of smart connected vehicles, the threat of cyber-attacks and security breaches has become more pronounced. Consequently, the development of a fast and accurate intrusion detection system is paramount to thwarting such attacks.

We propose an in-vehicle CAN-FD bus ID intrusion detection system based on an improved GAN model. By enhancing the depth of the convolutional layer, we can extract image features more efficiently, enabling the recognition of potential threats with greater accuracy. We introduce the concept of a dual discriminator to bolster the detection rate and effectively handle unknown attacks.

Experimental results showcase the efficacy of our intrusion detection system, with message detection occurring within a mere 0.15 milliseconds, meeting the stringent real-time detection requirements during vehicle operation. The average detection rate across various attack types stands at an impressive 99.93%, representing a noteworthy 1.2% enhancement over the GIDS algorithm.

Furthermore, should manufacturers necessitate the adoption of an extended ID format, we stand ready to design corresponding models tailored to different standards and configurations. Additionally, we have delineated the future trajectory of our detection system, implementing a fault-tolerant control module in CANoe to address detected anomalies or intrusion messages. The integration of intrusion detection with a fault-tolerant module holds promise for enhancing vehicle safety and stability.

REFERENCES

[1] X. Li, "Analysis of safety protection in vehicle networking," *Mobile Commun.*, pp. 32–35, Nov. 2015.
[2] V. Maglogiannis, D. Naudts, S. Hadiwardoyo, D. van den Akker, J. Marquez-Barja, and I. Moerman, "Experimental V2X evaluation for C-V2X and ITS-G5 technologies in a real-life highway environment," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1521–1538, Jun. 2022, doi: 10.1109/TNSM.2021.3129348.
[3] Z. Feng, "Research progress on key technologies of automobile information security attack and defense," *J. Inf. Secur.*, pp. 1–14, Feb. 2017.
[4] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: Architecture, protocols, and security," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3701–3709, Oct. 2018, doi: 10.1109/JIOT.2017.2690902.
[5] W. Zeng, M. A. S. Khalid, and S. Chowdhury, "In-vehicle networks outlook: Achievements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1552–1571, 3rd Quart., 2016, doi: 10.1109/COMST.2016.2521642.
[6] S. Jadhav and D. Kshirsagar, "A survey on security in automotive networks," in *Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA)*, Pune, India, Aug. 2018, pp. 1–6, doi: 10.1109/ICCUBEA.2018.8697772.
[7] L. Zhou, S. Du, H. Zhu, C. Chen, K. Ota, and M. Dong, "Location privacy in usage-based automotive insurance: Attacks and countermeasures," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 1, pp. 196–211, Jan. 2019, doi: 10.1109/TIFS.2018.2848227.
[8] R. G. de Oliveira, N. Navet, and A. Henkel, "Multi-objective optimization for safety-related available E/E architectures scoping highly automated driving vehicles," *ACM Trans. Design Autom. Electron. Syst.*, vol. 28, no. 3, pp. 1–37, May 2023, doi: 10.1145/3582004.
[9] *CAN With Flexible Data-Rate*, Version 1.0, BOSCHS, Gerlingen, Germany, Apr. 2012.
[10] K. Zdenek and S. Jirí, "Simulation of CAN bus physical layer using SPICE," in *Proc. Int. Conf. Appl. Electron.*, Pilsen, Czech Republic, Sep. 2013, pp. 1–4.
[11] G. M. Zago and E. P. de Freitas, "A quantitative performance study on CAN and CAN FD vehicular networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4413–4422, May 2018, doi: 10.1109/TIE.2017.2762638.
[12] Y. Xie, G. Zeng, R. Kurachi, F. Xiao, and H. Takada, "Optimizing extensibility of CAN FD for automotive cyber-physical systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7875–7886, Dec. 2021, doi: 10.1109/TITS.2021.3059769.
[13] J. Ning, J. Wang, J. Liu, and N. Kato, "Attacker identification and intrusion detection for in-vehicle networks," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1927–1930, Nov. 2019, doi: 10.1109/LCOMM.2019.2937097.
[14] H. Alnabulsi and R. Islam, "Protecting code injection attacks in intelligent transportation system," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Communications/13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Rotorua, New Zealand, Aug. 2019, pp. 799–806, doi: 10.1109/TrustCom/BIGDATASE.2019.00116.
[15] Z. Khan, M. Chowdhury, M. Islam, C.-Y. Huang, and M. Rahman, "Long short-term memory neural networks for false information attack detection in software-defined in-vehicle network," 2019, *arXiv:1906.10203*.
[16] M. Jedh, L. Ben Othmane, N. Ahmed, and B. Bhargava, "Detection of message injection attacks onto the CAN bus using similarities of successive messages-sequence graphs," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4133–4146, 2021, doi: 10.1109/TIFS.2021.3098162.
[17] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Calgary, AB, Canada, Aug. 2017, pp. 57–5709, doi: 10.1109/PST.2017.00017.
[18] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Belfast, Ireland, Aug. 2018, pp. 1–6, doi: 10.1109/PST.2018.8514157.
[19] M. Bozdal, M. Samie, and I. K. Jennions, "WINDS: A wavelet-based intrusion detection system for controller area network (CAN)," *IEEE Access*, vol. 9, pp. 58621–58633, 2021, doi: 10.1109/ACCESS.2021.3073057.
[20] A. Derhab, M. Belaoued, I. Mohiuddin, F. Kurniawan, and M. K. Khan, "Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 2366–2379, Mar. 2022, doi: 10.1109/TITS.2021.3088998.
[21] L. Zhang, "Intrusion detection systems to secure in-vehicle networks," Tech. Rep., 2023.
[22] D. Ding, Y. Wei, C. Cheng, and J. Long, "Intrusion detection for in-vehicle CAN bus based on lightweight neural network," *J. Circuits, Syst. Comput.*, vol. 32, no. 7, May 2023, Art. no. 2350110.
[23] Y. L. Aung, "VNGuard: Intrusion detection system for in-vehicle networks," in *Proc. Int. Conf. Inf. Secur.* Cham, Switzerland: Springer Nature, 2023, pp. 79–98.
[24] R. Zhao, C. Luo, F. Gao, Z. Gao, L. Li, D. Zhang, and W. Yang, "Application-layer anomaly detection leveraging time-series physical semantics in CAN-FD vehicle networks," *Electronics*, vol. 13, no. 2, p. 377, Jan. 2024.
[25] H. Ma, "A GRU-based lightweight system for CAN intrusion detection in real time," *Secur. Commun. Netw.*, Jun. 2022.
[26] K. Wang, C. Gou, Y. Duan, Y. Lin, X. Zheng, and F.-Y. Wang, "Generative adversarial networks: Introduction and outlook," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 588–598, 2017, doi: 10.1109/JAS.2017.7510583.

XU WANG received the B.S. degree in electronics and telecommunication from Yanbian University, Yanji, China, in 2021, where he is currently pursuing the M.S. degree in electronic and information engineering. His research interests include the implementation of secure intrusion and detection in vehicle bus networks.

YIHU XU received the Ph.D. degree in electronic and information engineering from Chonbuk National University, Jeonju, South Korea, in 2015. Since 2016, he has been with the Department of Electronics and Communication Engineering, Yanbian University, Yanji, China. His research interests include nextgeneration mobile communications, cognitive radio, and in-vehicle networks.

YINAN XU received the Ph.D. degree in electronic and information engineering from Chonbuk National University, Jeonju, South Korea, in 2009. He was with the Department of Electronics and Communication Engineering, Yanbian University, Yanji, China. His research interests include in-vehicle networks and automotive electronic control.

ZIYI WANG is currently pursuing the degree in communication engineering with Yanbian University, Yanji, China. Her research interest includes the implementation of security protocol for in-vehicle networks.

YUJING WU (Member, IEEE) received the M.S. and Ph.D. degrees in electronic and information engineering from Chonbuk National University, Jeonju, South Korea, in 2013 and 2016, respectively. She is currently with the Department of Electronics and Communication Engineering, Yanbian University, Yanji, China. Her research interests include VLSI implementation for digital signal processing and communication systems, which include the design of CAN data reduction and DisplayPort and implementation of security protocol for in-vehicle networks.