Protecting Automotive Controller Area Network: A Review on Intrusion Detection Methods Using Machine Learning Algorithms

Jia Zhou, Weizhe Zhang, Guoqi Xie, Renfa Li, and Keqin Li

1 Introduction

1.1 Background and Motivation

The automotive industry is undergoing rapid changes. The in-depth integration of advanced information technology and automotive technology enables the vehicles equipped with more intelligent functions and more connections with outside. Despite a higher level of comfort, safety, efficiency and personalized experience providing for drivers, the vehicles are also exposed to negative risks brought by the new technologies. The rich connectivity with external environments also means more potential access points which can be exploited by malicious adversaries. The adversaries can further intrude the safety-critical in-vehicle network via compromising the bridge nodes. Considering that vehicle is a man-in-the-loop cyber physical system, the attacker can further gain the ability to control the physical components

J. Zhou
Department of New Networks, Peng Cheng Laboratory, Shenzhen, China
e-mail: zhoujia@hnu.edu.cn

W. Zhang (✉)
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

Department of New Networks, Peng Cheng Laboratory, Shenzhen, China
e-mail: wzzhang@hit.edu.cn

G. Xie · R. Li
Key Laboratory for Embedded and Network Computing of Hunan Province, College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
e-mail: xgqman@hnu.edu.cn; lirenfa@hnu.edu.cn

K. Li
Department of Computer Science, State University of New York, New Paltz, NY, USA
e-mail: lik@newpaltz.edu

V. K. Kukkala, S. Pasricha (eds.), *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*, https://doi.org/10.1007/978-3-031-28016-0_9

of automotive and manipulate its behaviors. It may result in a threat to human life or deeper security issues to the whole society. Security concern has become one of the most challenging issues for in-vehicle network which cannot be ignored.

In-vehicle network is the underlying base for the implementation of automotive functions such as driving safety, autonomous driving, intelligent in-cabin system, and body control. Accordingly, the in-vehicle network is also in the process of innovation to meet future requirements. With the rapid development of intelligence and connectivity of vehicles, the architecture of in-vehicle network is undergoing evolution from distributed model to domain model and zonal model. It is getting more complex and sophisticated, which usually comprises several networks responsible for different functions. In this chapter, we mainly focus on currently the most popular in-vehicle communication protocol Controller Area Network (CAN), which is directly responsible for the safety of vehicles. From our point of view, CAN will still bear an important role in ensuring driving safety in the future in-vehicle network. How to defend automotive CAN bus draws much attention from the public as well as academia.

CAN is capable of providing reliable and real-time communication to ensure the safety of the automotive control systems. But there is no any inherent mechanism at its birth to defend against malicious adversary. Its characteristics such as broadcast nature, plain-text transmission, lack of message authentication, and weak access control make the automotive CAN network vulnerable to cyber attack. Security schemes such as cryptographic measures are introduced in the automotive domain. Message Authentication Code (MAC), which can provide the ability to verify the data integrity as well as identify the sender seems like a good option. It is implemented based on a symmetric cryptographic mechanism, which can favor the deployment on automotive embedded systems by reducing the computational complexity. However, the extremely limited length of the CAN frame cuts the effect of the deployment of message authentication codes. For example, the maximum data payload of a data frame of the standard CAN protocol is only 8 bytes. The longer message authentication code results in a shorter payload which degrades the efficiency of the communication system, while the shorter message authentication code results in an insufficient security level. To mitigate this issue, the longer authentication tag can be transmitted via extra frames. Unfortunately, it can result in a heavier bus load which might affect the real-time performance of the system.

The intrusion detection method can be a simple but efficient solution for protecting in-vehicle network. It can monitor the network traffic and detect anomalies during the runtime of vehicles. Different from the encryption and authentication measures, intrusion detection methods do not occupy the limited bandwidth and payload of the in-vehicle network. It works based on the observation and analysis of network traffic. The intrusion detection system was firstly introduced for in-vehicle network by Hoppe et al. [12]. The authors proposed three ways to utilize features, which are the increase in the frequency of CAN frames, the observation of signal characteristics as well as the abuse of CAN identifiers to detect attacks. More schemes based on intrusion detection methods are designed since then. One way of

designing the intrusion detection system is to build a physical model or pre-defined rules to detect unexpected behaviors. However, the in-depth knowledge about the system is always required for this kind of approaches. Besides, it is difficult to design a closed-loop expression to detect attacks in real cases. Machine learning (ML) is one of the most promising technologies nowadays which can also favor the solution for security concerns of in-vehicle network. ML can extract latent patterns from traffic to provide an effective and flexible solution for intrusion detection on in-vehicle network.

1.2 Contributions and Outline

In this chapter, we survey the studies which take advantage of machine learning technologies to detect intrusion for automotive CAN bus. The structure of our chapter can be seen in Fig. 1. To provide a better understanding about the application scenarios, we firstly introduce the in-vehicle network architecture and how it evolves. Next, we provide a detailed description about the intrusion detection methods exploiting ML algorithms. According to the domain knowledge used for extracting features by ML, we divide these approaches into four categories, which are semantics-based methods, literal-based methods, timing-based methods and signal characteristics-based methods respectively. Our contributions can be concluded as follows:

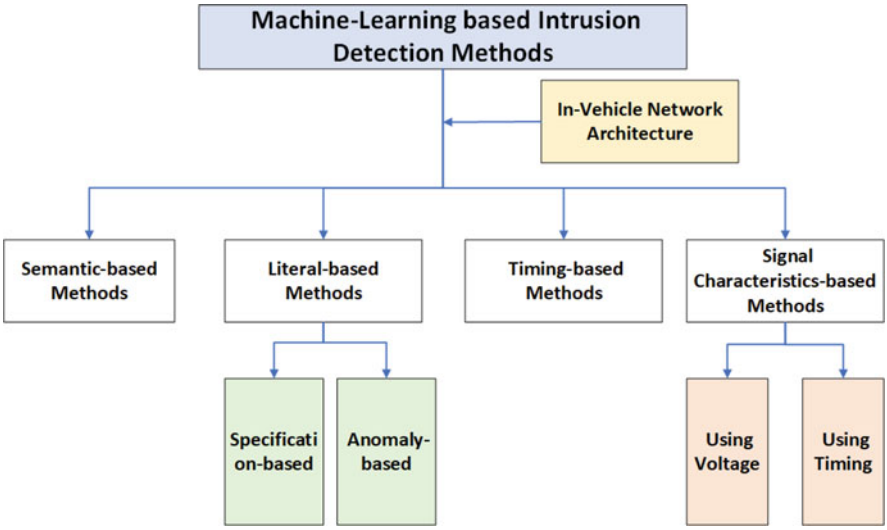

Fig. 1 Structure of the chapter

1. We provide an introduction about current and future in-vehicle network architecture. The evolution trend emphasizes the importance of CAN for driving safety and the necessity to protect it.
2. We classify the machine learning-based intrusion detection methods based on the domain knowledge exploited to extract features. The domain knowledge can be referred to those low-level characteristics in CAN such as timing characteristics or signal shapes, or the high-level characteristics such as the data payload of CAN frames or their semantic values.
3. We provide a detailed description for each category of intrusion detection methods. In each section, we firstly introduce the basic insight of how it works and discuss the disadvantages of the traditional methods. Then, we introduce the existing work based on machine learning algorithms.

The organization of this chapter is as follows: Sect. 2 provides the description about the current and future in-vehicle network architecture. Sections 3 to 6 describes the intrusion detection methods exploiting machine learning algorithms from four aspects, which are semantics-based methods, literal-based methods, timing-based methods and signal characteristics-based methods respectively. Finally, Sect. 7 concludes this chapter.

2 In-Vehicle Network Architecture

In this section, we first provide a description of the in-vehicle network architecture and how it will upgrade in the near future. We also briefly conclude the benefits brought by the architectural evolution. Then, we provide a primer on CAN and illustrate the necessity for research on protecting CAN. From our point of view, CAN will not be abandoned by the future in-vehicle network and will face more security risks. Thus, defending CAN from attacks is important for protecting vehicles no matter for the current or future in-vehicle network.

2.1 Evolution of In-Vehicle Network

The hardware of in-vehicle network mainly consists of two parts, which are the Electronic Control Units (ECUs) and wired cables to connect the ECUs. The ECU is an automotive embedded device equipped with abilities of computing, communication and control. The data and control signals of ECUs can be exchanged over the wired cables. All ECUs inside the vehicle are networked with each other through the internal communication system to form a whole. The whole system can provide the ability from sensing the driving environment to making decisions and implementing high-level automotive driving.

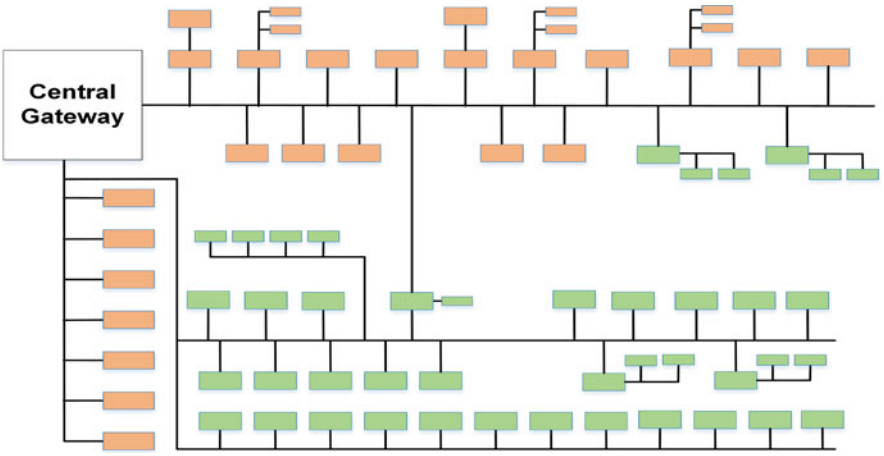

Fig. 2 Distributed in-vehicle network architecture

The traditional in-vehicle network adopts a distributed architecture (as shown in Fig. 2). All ECUs are scattered on the network and work distributively. Generally, it equips low-speed communication protocols such as CAN and LIN (Local Interconnect Network) as the backbone network. The distributed in-vehicle network enables the transition of automotive from mechanization to electronics. However, the increasing number of electronic functions and ECUs has lead to a heavy, large-scale in-vehicle network, making the wiring harness system the third-heaviest automotive component after the engine and chassis [35]. The bulky wiring harness system increases the total weight of the vehicle, resulting in higher energy consumption and cost. Besides, the increasing number of ECUs makes the in-vehicle network more complex. It could lead to a higher cost of software development as well as a higher cost of software verification and validation which might increase the risk of uncertainty.

Furthermore, the demand for automobile intelligence and the rising connections with outside are forcing the innovation of the communication architecture of in-vehicle network. Various advanced communication technologies such as 5G, WIFI, Bluetooth, and Vehicle-to-Everything (V2X) have been deployed on vehicles, which makes vehicles as a complex communication system. To realize the advanced intelligent functions of vehicles, the concept of Software Defined Vehicles (SDV) has gradually become the mainstream for automotive software development. The high integration of automotive technology and information technology increases the complexity of the intelligent connected vehicles continuously, which requires a scalable design of architecture and coordination of ECUs with higher computing power. To meet these requirements, the architecture of in-vehicle network would evolve from the traditional distributed architecture to a new generation of centralized architecture. Specifically, as shown in Fig. 3, it would gradually evolve into a

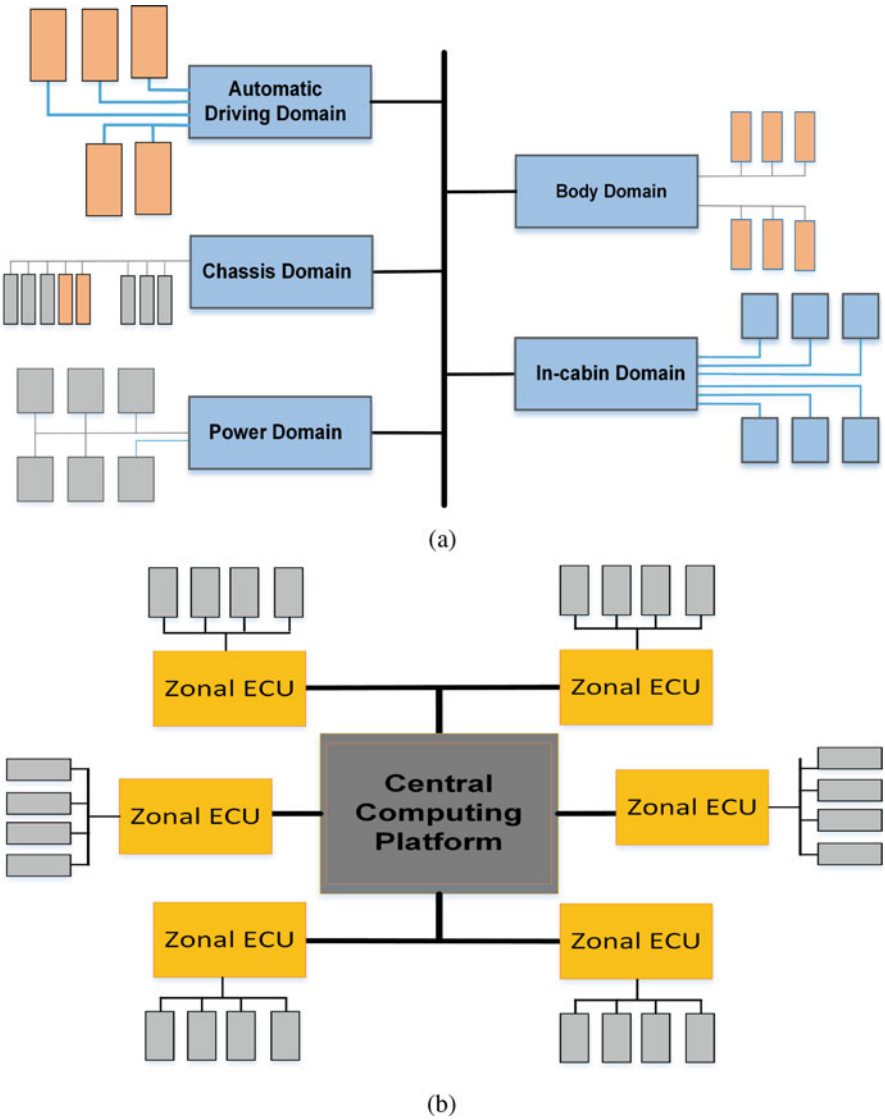

Fig. 3 New in-vehicle network architecture. (**a**) Domain architecture. (**b**) Zonal architecture

domain architecture (Fig. 3a), and further, a more centralized architecture called zonal architecture (Fig. 3b).

One common scheme of domain architecture is to divide the in-vehicle network into five different control domains according to their functions. Each domain is equipped with a Domain Control Unit (DCU) to centralize the functions and computing resources within the domain. The DCU is a higher-performance auto-

motive microcontroller designed to solve the problem of performance bottleneck of distributed in-vehicle network. As shown in the left of Fig. 3, it consists of the power domain responsible for powertrain control and optimization, the chassis domain responsible for driving behavior (braking, steering, transmission, etc.), the body domain responsible for body and comfort control, the in-cabin domain responsible for entertainment, and the automatic driving domain responsible for assisting vehicle driving. The DCU consolidates the functionality within each domain and communicates with other DCUs via high-speed backbone network (such as Ethernet, etc.). In-domain ECUs attached to the DCU are degraded to low-level ECUs or actuators with limited computing and communication resources. Low-speed communication protocols (such as CAN, LIN, etc.) are exploited to connect the DCU with the in-domain nodes.

The zonal architecture further improves the degree of centralization by organizing a three-layer architecture. It consists of the following key components, including (1) computing resources which are a central computing platform, multiple zonal ECUs and many low-level ECUs; (2) communication resources which are high-speed backbone network (such as Ethernet, etc.) to connect the central computing platform with zone ECUs and low-speed local area network (such as CAN, LIN, etc.) to connect the zone ECUs with low-level ECUs. The hardware inside the local area network can be consolidated by the upper level zone ECU, while the hardware of zone ECUs can be further consolidated by the central computing platform. Highly consolidation of hardware resources makes it more available to separate software and hardware to achieve the concept of software-defined vehicles. It can manage the needs of more advanced and intelligent functions for future vehicles.

Currently, most car manufactures are in the stage of transition from distributed architecture to domain architecture. In general, the upgrade of the in-vehicle network architecture can bring advantages in terms of cost reduction and driving intelligence, which are listed as follows:

1. Reduction on hardware cost: Benefiting from architecture evolution, the total number of ECUs can be significantly reduced to optimize the utilization of computing resources. In addition, the layout of the wiring harness system can be optimized, lowering the total weight and hardware cost of vehicles.
2. Reduction on development and verification cost: The highly integration of hardware can favor the application of scalable software-driven framework for decoupling of hardware and software, leading to faster development cycle and lower cost of software development and verification.
3. Support for implementation of OTA: The Over the Air (OTA) technology can achieve the goal to upgrade the automotive software remotely through wireless access points of vehicles. It can provide a convenient, timely, and lower cost of recall management by cutting the necessity to bring the vehicles back. The centralized architecture with fewer ECUs and unified software architecture can reduce the verification complexity of the OTA update process.
4. Support for implementation of advanced intelligent functions: Vehicle intelligence requires the powerful hardware as well as the advanced software devel-

opment model. The application of the scalable software-driven framework, high performance computing platform and heterogeneous communication architecture which are benefited from the new in-vehicle architecture can make it possible to implement advanced functions like intelligent in-cabin system and high-level autonomous driving.

2.2 *The Necessity for Protecting CAN*

CAN is currently the most mature protocol with the highest market share, and has been required to be implemented on production vehicles. It is widely used in automotive network related to safety-critical functions such as automobile transmission and body control. The safety-critical information, e.g., the engine or cruise control is exchanged over the CAN bus. The data in CAN is exchanged via the unit called data frame. Its structure can be divided into five fields, including arbitration field, control field, data field, CRC (Cyclic Redundancy Check) field and ACK (Acknowledgement) field (can be seen in Fig. 4). The arbitration field bears the identifier which can be used for identifying different frames as well as competing the rights of transmitting on the bus.

Safety is always the first priority for vehicles. Despite the proportion of CAN for the in-vehicle network is getting smaller as the architecture evolves, the urgency for research on protecting CAN is even getting stronger. The reasons can be explained as follows. Firstly, CAN will not be abandoned by the future in-vehicle network due to its high efficiency and low cost. Despite many advanced technologies such as high-speed Ethernet and high performance computing devices are introduced, the lowest level network for both the domain model and zone model would still be developed as a signal-oriented communication paradigm. Such design can provide reliable and real-time data exchange to ensure the safety of vehicles. CAN is still going to play critical role in these areas, especially the networks for safety-critical functions. Secondly, the risk of in-vehicle network being attacked increases significantly. The evolution of in-vehicle network architecture is along with the trend that the number of communication technologies used in vehicles increases. That also opens more doors for attacks, resulting in higher security concerns for vehicles. The attackers can intrude on the in-vehicle network by exploiting the flaws in the hardware or software of these access points. Since CAN was originally designed to work in an isolated environment, CAN does not take any security concerns into

Fig. 4 The CAN data frame format

consideration [8], making CAN vulnerable to attacks. It has been demonstrated that the adversary can manipulate the vehicles' behavior after obtaining access to the safety-critical CAN bus [18]. Thus, we claim that defending CAN from attacks is critical for ensuring the safety of vehicles no matter on the current or future in-vehicle network.

3 Semantic-Based Intrusion Detection Methods

3.1 Motivation and Basic Idea

The data transmitted on the in-vehicle network has specific physical meaning for describing the current states and dynamics of the vehicle. An example of physical variables transmitted on in-vehicle CAN is listed in Table 1. For instance, the data can be explained as the speed of the engine, vehicle velocity or the state of the headlights. These data are transmitted and exchanged over the in-vehicle network to control the various functions of vehicle.

For a given dynamic of automotive system, there should be a certain correlation between data read from different sensors since they obey the same physical law. Under normal circumstance, the variable which indicates the inclination angle of the accelerator pedal should change accordingly when the driver presses the pedal. The speed of the engine and vehicle velocity would increase. In the meantime, the automotive gear would also switch in time. The different parts of the vehicle collectively respond to the act of pressing the accelerator pedal in a correlated and consistent manner. Therefore, the physical properties of vehicles can be abstracted by the physical model built from the semantic traffic. The correlation among different sensors can be exploited to detect anomaly. We assume that the attacker cannot compromise all relevant ECUs simultaneously which is plausible in real scenarios. The intrusion detection is to identify any observation which is inconsistent with expected behavior.

To detect unexpected behavior, the first priority is to construct the model for describing the relationship between variables obeying same physical laws. One

Table 1 An example of physical variables on CAN

Physical variables	
Vehicle speed	Position of steer
GPS speed	Torque of wheel
Acceleration pedal	Wheel angle
Brake pedal	Gear
Engine RPM	Coolant temperature
Fuel rate	Ambient temperature
Fuel/Air commanded equivalence	Air intake temperature
Master cylinder pressure	Boost pressure

way is to build the physical model manually based on the physical expression or experience. Cho et al. [5] proposed an anomaly detection method called Brake Anomaly Detection for the brake-by-wire system. Under normal circumstances, the behavior of the vehicle should be consistent with the driver's intent and the surrounding driving environment. The authors chose the Brush tire model [2] as the normal behavior model to characterize the frictional relationship between the tire and the ground. The attack to the brake-by-wire system can be observed by checking the consistency between the driver's input and the actual data captured from the in-vehicle network. The model also takes into account the change in the coefficient of friction of the tires under different weather and road conditions. Similarly, Ref. [10] designed a delicate ring-based architecture to organize multiple correlations by utilizing the physical model and experience. In this study, ten variables and nine nodes in total comprise the well-designed correlation ring to improve the robustness of detection while reducing the overall computation overhead.

However, these methods require in-depth understanding about the target system and expertise, which may not always be available. Researchers resort to machine learning algorithms to construct the model automatically that reflects the physical laws. It is mainly based on the insight that multiple sensors readings are directly proportional to the same physical phenomenon under normal circumstances [1]. Thus, the model can be generated from semantic traffic of in-vehicle network without the requirement for the in-depth knowledge of the control system. The machine learning algorithms to be exploited can be varied including artificial neural network [33], random forest regressor [20], deep autoencoder [11], and CNN model [13].

3.2 Machine Learning-Based Methods

Reference [20] formulated the problem to detect anomalies as a machine learning prediction problem that can be resolved by the regression model. The authors selected a set of correlated sensor data as features of the regression model based on domain knowledge and pairwise correlations firstly. The sensor signals which can be used for calculating vehicles' speed are taken as an example in this study. They included engine speed, acceleration on both longitudinal and lateral orientation, brake pedal ratio, steering angle, gear, and so on. During the training phase, the feature readings are fed into a Random Forest Regressor to train a regression model. While in the testing phase, the output values of the model can be estimated continuously based on the trained regression model. The anomaly can be flagged once the difference between the observed value and the estimated value is larger than a predefined threshold.

A more advanced learning technique for generating the physical model automatically is introduced in an intrusion detection system called context-aware intrusion detection system (CAID) [33]. CAID exploits the Bottleneck Artificial Neural Network (ANN) to develop the reference model of the automotive control system.

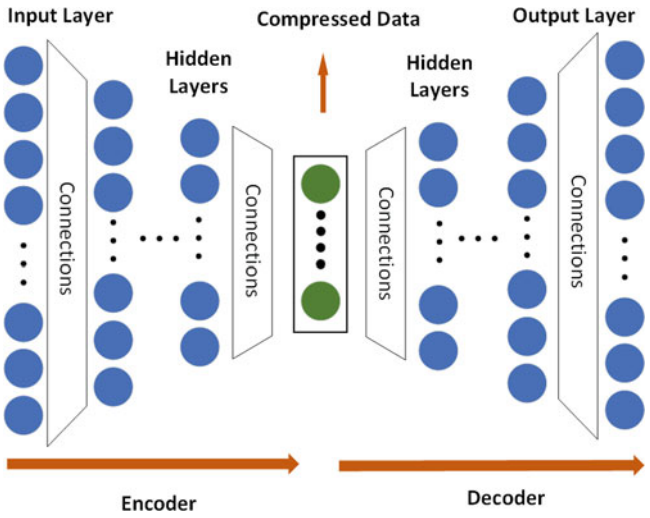

Fig. 5 Architecture of deep autoencoder neural network

The bottleneck ANN is designed as a network model in that the input and output layers are with the same number of neurons while the hidden layer is with a significantly less number of neurons. The sensor signals to describe the state of the engine control unit, such as fuel rate, absolute throttle position, engine RPM, and seven other signals, are collected to validate the performance of the proposed method. The parameters of the model can be generated in the training phase. During the testing phase, the estimated value can be obtained by reconstructing an input via the trained bottleneck ANN. CAID can detect anomalies by checking the similarity of actual readings of the sensor against the estimated values.

Reference [11] devised a deep autoencoder-based intrusion detector to extract the inherent redundancy of related sensors. The autoencoder (as shown in Fig. 5) is composed of two parts which are the encoder and the decoder respectively. Both the encoder and decoder are deep neural networks with multiple hidden layers. The aim of the encoder is to compress the input into low-dimensional features as much as possible, while the decoder aims to restore the compressed features to the original data as much as possible. By cascading the encoder and the decoder together, the autoencoder can extract the pattern of the input data. The overall process of the research [11] is as follows. Firstly, the authors selected a set of correlated data as input. The evaluation is performed on a publicly available dataset. It includes three categories of data, which are data from sensors on CAN bus, data from GPS sensors, and data from IMU sensors. Next, the deep autoencoder is adopted to learn the consistent pattern of these sensor data from the trustful training dataset. The learned consistent pattern can be expressed as the normal behavior of the automotive control system. In [11], the evaluated autoencoder network is designed with a 4-layers encoder and 4-layers decoder. The authors defined three different means to

measure the error of the input against the reconstructed output. The training process of the encoder and decoder can be repeated to update the parameters of the model by minimizing the reconstruction error. Finally, anomalous behaviors can be detected by checking the reconstruction error during running. The reconstruction error shall be ranged within a predefined bound. If the reconstruction error exceeds the bound, an intrusion can be alarmed.

Reference [13] designed a framework that comprises an anomaly detection method based on Convolutional Neural Network (CNN) as well as an ensemble classifier which consists of multiple traditional machine learning algorithms. The ensemble classifier is to evaluate the effectiveness of the proposed CNN-based anomaly detection method. The proposed CNN-based method introduces a multi-stage attention Long Short-Term Memory (LSTM) model to enable the algorithm can focus on the significant parts of the data. The authors provided a comprehensive evaluation of four distinct anomaly types generated by [31] which are instant, constant, gradual drift, and bias to a publicly available dataset, and their combinations.

3.3 Summary

The semantic-based methods exploit the fact that the CAN traffic over the automotive network bears specific physical meanings for representing the dynamics or states of vehicles. Thus, these physical variables can be used to construct the abstract of the physical properties of vehicles. Machine learning algorithms can build the model automatically without requiring in-depth knowledge of the target system. The inconsistency with expected behavior can be regarded as an intrusion. Despite reducing the effort for generating the model compared to the traditional methods, the proprietary nature of CAN makes the obtainment of the specific meanings of the CAN frames a non-trivial work. It hinders the research on semantics-based methods since the specific meanings of the frames are kept confidential from the public.

4 Literal-Based Intrusion Detection Methods

4.1 Motivation and Basic Idea

There are two main limitations of semantic-based intrusion detection methods. First, it is non-trivial to obtain the semantic meaning of data from in-vehicle network. The automotive industry is not willing to disclose the detailed specification of their CAN messages considering the concerns on intellectual property and security. That is, the detailed meaning of automotive CAN messages cannot be obtained publicly. Second, the selection of input data requires domain knowledge or correlation computation. The performance of such methods on irrelevant data beyond the

selected sensor has not been verified. These limitations hinder the application of semantic-based intrusion detection methods.

In this section, we introduce one more intuitive kind of method called literal-based intrusion detection method. It is unnecessary to obtain or derive the semantics of the CAN messages painstakingly. The binary streams (literal value) can be exploited directly as the input for the intrusion detection system. Firstly, the inherent correlations are extracted by analyzing the binary stream of CAN traffic. The extracted correlations can be used to characterize the normal behavior of the system or pattern of the anomalies. After building the required model for the target system, the intrusion can be reported by comparing the expected data with the observed one.

The main insight behind the literal-based intrusion detection methods can be summarized as follows. CAN is highly deterministic and predictable during operations to manage the requirements for strict real-time, and provide stable and reliable services. The stable operational patterns for CAN shall be observed in the absence of cyber attack. It has been pointed out in [9, 21] that the model of normal behavior can be established from the analysis of CAN data streams without understanding the semantics of CAN messages. Information entropy is a measurement to describe the uncertainty of a system. The more orderly and deterministic the system is, the lower the information entropy is. Reference [24] proposed the entropy-based intrusion detection methods for in-vehicle network. The entropy of the data traffic can be computed for representing the state of CAN traffic. When the entropy value deviates from the normal range, it means that there is an attack mounted on the in-vehicle network. However, the estimation of the entropy value can be affected easily by different driving scenes, which results in a high false positive rate.

Machine learning algorithms are better options for processing the binary streams of CAN messages. Generally speaking, the overall process of these methods can be concluded as two phases, which are the training phase and the testing phase as

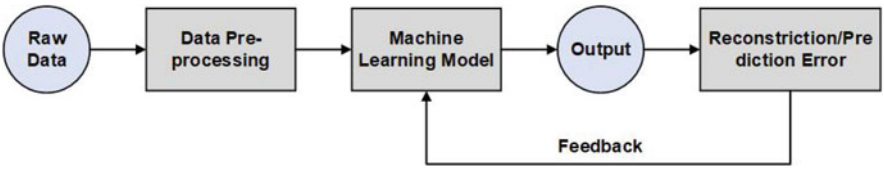

Fig. 6 The training phase

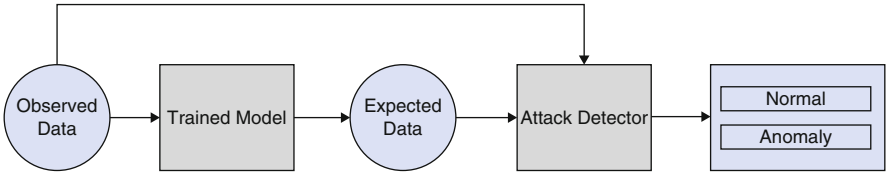

Fig. 7 The testing phase

illustrated in Figs. 6 and 7 respectively. The aim of the training phase is to develop the model for intrusion detection by extracting latent patterns from the traffic of in-vehicle network. The raw data, i.e. the literal binary data of CAN messages after pre-processing can be fed into the machine learning algorithm to train the model. The feedback process is to minimize the reconstruction/prediction error to improve the model performance. The training phase can be performed offline in a controlled environment considering it is a time-consuming task. During the testing phase, the observed data is compared with the output (expected data) by the trained model to detect anomalies. The observed data and the expected data are fed into the attack detector together to identify whether their difference exceeds a well-designed threshold.

The methods in this section can be divided into two categories according to whether the attack sample is required in the training phase, which is specification-based methods using attack samples for the training model and anomaly-based methods using normal samples for generating the model.

4.2 Specification-Based Methods

Methods in this category require labeled attack samples for training the classification model. The model can learn the patterns of the CAN traffic under attack during the training phase. The intrusion can be detected once any similar patterns are observed during the testing phase.

Xie et al. [34] proposed a generative adversarial network (GAN) based intrusion detection method, which can be shown in Fig. 8. Technically, the GAN model consists of two core components: generator (G) and discriminator (D). The basic principle of how GAN works is as follows. The generator utilizes random noise as input and tries to output synthetic data to deceive the discriminator. On the contrary, the discriminator utilizes the ground truth as input and tries to make decisions as accurately as possible that the data from the generator is whether fake or not. The performance of the generator and discriminator can thus be improved during the repeated adversarial process. In [34], the real attacked CAN messages are fed into GAN for training the intrusion detection model.

CANintelliIDS [14] is designed based on a convolutional neural network (CNN) combined with an attention-based gated recurrent unit (GRU) model. Similar to LSTM, the GRU model is suitable for solving the prediction problem of sequential data. Besides, the utilization of GRU can be helpful for improving the efficiency as well as reducing the memory consumption considering its more simplified design and fewer parameters compared to LSTM. The intrusion detection model is trained based on the attack dataset. Different attack scenarios with single or mixed attack types are evaluated in this work.

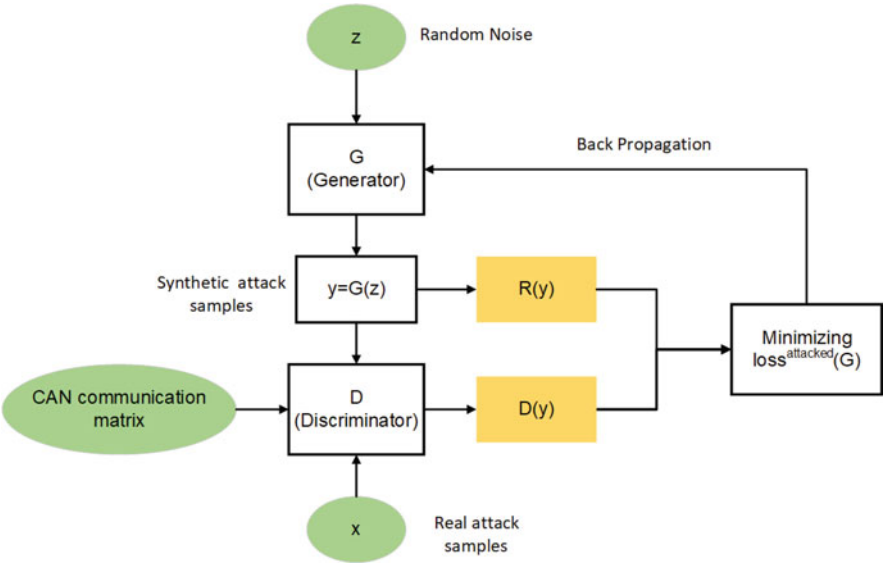

Fig. 8 Training process of GAN generator in Ref. [34]

4.3 Anomaly-Based Methods

The methods belonging to this category do NOT require the attack labeled data during the training phase. The intrusion detection model is generated from the attack-free CAN traffic under normal circumstances. If there is any deviation from the normal model is detected, an intrusion can be alarmed. Compared with specification-based methods, the performance of anomaly-based methods to detect unknown attacks is preferable.

CANnolo [21] implements LSTM as the hidden layer of the auto-encoder. The auto-encoder is used to automatically learn the normal patterns of raw CAN data without semantics. At runtime, CANnolo utilizes the trained model to reconstruct the CAN streams. The Mahalanobis distance between the reconstructed and the observed sequences is computed as an anomaly score to indicate how likely the CAN bus is under attack. Reference [29] designed an LSTM-based RNN model constituted by two non-recurrent hidden layers and two recurrent LSTM layers. To improve the accuracy of the detection model, the features on the time dimension are combined with features on the data dimension as input for LSTM neural network [39]. Besides, the multi-task LSTM framework is utilized to implement parallel computing locally as well as on the mobile edge. The mobile edge can break the limitation of onboard computing capacity.

HDAD [32] introduces the concept of hyper-dimensional computing (HDC) to detect intrusion for in-vehicle network. HDC is a novel computing paradigm that

simulates the working mechanism of neuronal circuits in the human brain. It works using high-level and abstract patterns of neural activity. Firstly, the training data are encoded into hypervectors (HVs) to learn pattern. Only normal patterns are required in the training dataset. The number of dimension for HV can be set as 10,000 or larger. Then, the pattern decoder is subject to reconstruct the HV to the original data. Finally, the reconstruction error is used for determining if there is an intrusion. The authors claimed that the adoption of HDC can benefit from compact model size, reduced computation cost, and one-shot learning in contrast to deep learning-based approaches.

The research on HDC is still at the preliminary stage. To improve the efficiency of the intrusion detection model, CLAM [28] improves the process of data pre-processing to cut the dimensionality of raw CAN traffic which can favor the acceleration of computation. Specifically, READ [22] method designed for reverse engineering of automotive data frames is introduced to assist the data reduction in CLAM. READ method can analyze the traffic and extract signals that vary continuously without supervision. These extracted signals can be explained as physical signals with specific physical meanings such as vehicle speed and engine speed. In the step of data pre-processing, the signal boundaries can be determined by READ methods. Thus, instead of using the whole CAN frame as input, only the bits bounded by data pre-processing are conveyed to the intrusion detection model for improving efficiency. It should be noted that the CLAM model also does NOT need to know the semantics of CAN frames. The CLAM model consists of a 1-D Convolution Network and bi-directional LSTM with an attention mechanism. The attention mechanism can enable the model to focus on the important parts of the data.

4.4 Summary

The literal-based intrusion detection methods can automatically extract intrinsic relationships among variables and develop the intrusion detection model by analyzing the binary stream of CAN frames. The semantics of frames are not required. The intrusion detection model can be trained by either the attack-free samples to generate the normal patterns of CAN frames or the attack-labeled samples to detect well-known intrusion. That is, the literal-based intrusion detection can be directly applied to CAN frames from the data link layer without knowledge of the protocol specifications of the upper layer (application layer). The protocol specifications of the application layer for automotive CAN bus are kept confidential from the public. Different specifications are defined for different car manufacturers and even different car models. From this perspective, compared to semantic-based methods, literal-based intrusion detection methods seem more attractive to both security technicians in the automotive industry as well as researchers from academia.

5 Timing-Based Intrusion Detection Methods

5.1 Motivation and Basic Idea

Considering that the vast majority of CAN frames are triggered periodically, i.e. CAN frames are queued for transmission at a fixed rate, there are some regularities of timing characteristics that can be found from CAN frames traffic. Illegal data due to unauthorized intrusion attacks can disrupt the regularities. Based on this observation, researchers propose that intrusion detection can be implemented by digging into the temporal patterns of CAN data traffic. The inconsistency with expected temporal patterns can be regarded as an anomaly. Similar to literal value-based methods, the timing-based method can also cope with the disadvantage of the proprietary nature of CAN data specifications. The traditional approach [25] builds the mathematical model to describe the timing behavior precisely of CAN frames traffic by utilizing real-time scheduling theory. However, the main downside is that it requires in-depth domain knowledge for building the model and it is hard to build a model adapted to different driving scenes.

5.2 Machine Learning-Based Methods

Tomlinson et al. [30] introduced three straightforward machine learning algorithms (Autoregressive Integrated Moving Average, Z-score, and supervised threshold) combined with time-defined windows to identify abnormal timing changes for CAN traffic. Reference [26] proposed a deep convolutional neural network (DCNN) model-based intrusion detection method. The authors designed a data pre-processing module called frame builder to convert the raw CAN traffic to the data fitted for the CNN model. Subsequently, the DCNN model learns temporal sequential patterns of raw CAN traffic automatically without hand-designed features. The CAN data with labels indicating whether normal or not is required for the training process. The Recurrent neural network (RNN) is naturally designed to cope with time sequence data. Reference [27] designed an RNN model with a 1-layer hidden layer of 100 nodes. From the evaluation results, the proposed RNN model can handle more realistic scenarios in that the period can fluctuate. The period fluctuation can often be observed in CAN traffic collected from real vehicles. It is mainly caused by the process of multiple ECUs to compete with the right of CAN bus usage. The attack samples are needed for computing the final output.

Generative Adversarial Network (GAN) is introduced in [15] to extract temporal features for modeling normal behaviors by attack-free training dataset. The authors improved the original GAN model by introducing a modified evolutionary algorithm to produce multiple generators instead of one single generator. This modification can increase the chance to obtain a better performance generator in the process of adversary game, which can mitigate the issue of instability in GAN. Since no

given attack sample is used for the training model, the data collection process shall be undertaken when driving under different conditions to capture as many normal features as possible. It can be helpful for reducing false positives.

5.3 Summary

Timing-based methods build the intrusion detection model by analyzing the timing characteristics of CAN traffic automatically. As same as the literal-based methods, the semantic values of CAN traffic are NOT required for timing-based methods. The timing-based methods can effectively detect attacks that essentially change the timing behavior of CAN frames, such as denial of service (DoS) attack, suspension attack and injection attack. However, from another perspective, the performance of such methods can be significantly degraded when dealing with more sophisticated attacks which do not influence the timing characteristics. Due to the broadcast nature of CAN, the attacker can eavesdrop and learn the temporal patterns of the target frames silently and stealthy. Next, the attacker can bypass the deployed timing-based intrusion detection system by injecting malicious frames with the same identifier and similar transmission pattern as the victim.

6 Signal Characteristics-Based Intrusion Detection Methods

6.1 Motivation and Basic Idea

Another way to design an intrusion detection system is to exploit the unique hardware characteristics of automotive ECUs to generate a digital fingerprint. Specially, the tiny but measurable differences in specific characteristics (such as voltage or timing) can be obtained from the electrical signal transmitted on the bus medium. The extracted difference can then be utilized as a device fingerprint to enable authentication in CAN. The intrusion can be detected when the actual sending ECU (predicted data) of the newly received CAN frame is inconsistent with its legitimate sending ECU (expected data).

The difference in hardware is mainly due to the imperfect manufacturing processes, which results in the characteristics of unique, stable, and hard to replicate to enable higher security. It was first introduced in [23] which exploits the difference of signal characteristics in the physical layer to identify ECUs for in-vehicle network. This study has demonstrated that the signal characteristics driven by the hardware of ECUs can be unique while remaining stable within a certain range for several months. Inspired by this observation, more researches to protect the in-vehicle network by utilizing low-level signal characteristics of CAN frames are proposed.

An idea of implementing intrusion detection based on signal characteristics is to explicitly define the relationship between the collected data and the hardware characteristics of the sending ECU by establishing a model. Most of such works exploit a linear model to represent the relationship of the accumulation of derived signal characteristics over time or data samples. Viden [4] adopts voltage measurements to build the model to source the sending node. Viden measures the voltage of CAN high and CAN low respectively during the transmission of dominant bits. These measurements are gathered to derive a voltage instance containing six statistics to describe the distribution of measurements. The voltage instance can be expressed as the transient behavior of voltage of sending ECU. At last, Viden constructs a linear model called voltage profile by utilizing the continuously obtained voltage instance. The main reason why Viden can work is that the voltage instances derived from the same ECU shall be nearly equivalent. Thus, the voltage profile can be constructed as a linear model by which the sending ECU can be correctly identified.

Different from Viden, Refs. [3, 19, 38] exploited the skew in clocks of electronic devices to establish the linear model for intrusion detection. The clock skew is defined as the difference in frequency between clocks. The common insight behind these methods is based on the observation that the clock skew is nearly constant for single ECU and unique among different ECUs. Thus, the linear model which represents the timing behavior of clock can be built for detect anomalies. Deviations from the established model can be used to trigger an alarm for intrusion on in-vehicle network. For example, the sudden change of the slope of the linear model can be regarded as an indication that the attack is mounted.

6.2 Machine Learning-Based Methods

Besides the model-based methods, the problem of identifying the sending ECU for newly received CAN frames can also be regarded as a classification problem. The CAN frames from the same ECU are considered to be of the same class. If the actual class of any CAN frames (identified by the intrusion detection system) is inconsistent with its expected class (determined by the frame identifier), it indicates that the adversary performs an attack by injecting frames with falsified ID. The supervised machine learning algorithms can be used to solve such classification problem. Generally speaking, the overall process of methods belonging to this category can be summarized into three phases as shown in Fig. 9.

The first step is to preprocess the electrical CAN signal to derive the characteristics from the physical layer. The signal characteristics exploited in this phase can be varied from voltages measurements to timing characteristics, which is the same as the model-based methods. Subsequently, the statistical features in the time and/or frequency domain are extracted from the measurements. Finally, the supervised learning-based classification algorithms are adopted to generate a classifier to distinguish the attack from the normal CAN traffic.

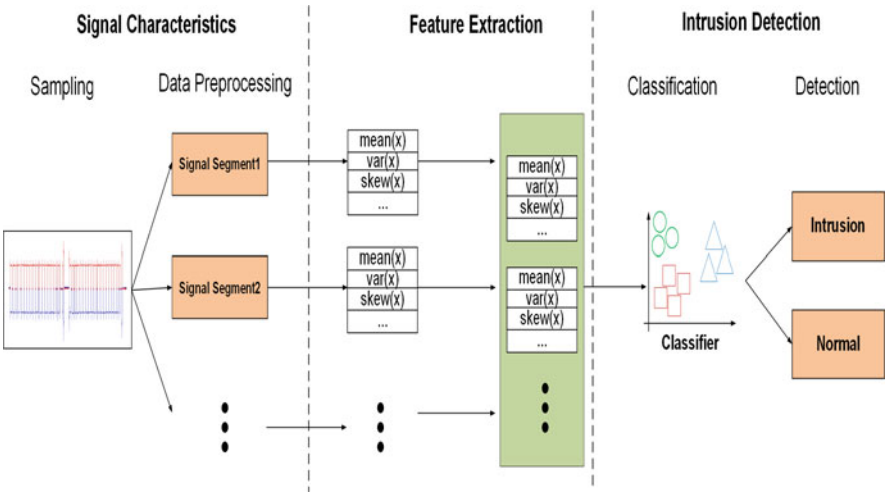

Fig. 9 Workflow of the supervised learning algorithms-based methods

6.2.1 Signal Characteristics Derivation

6.2.1.1 Using Voltage as Signal Characteristics

Choi et al. [6] proposed an approach to source the transmitting ECU by measuring the voltage of an array of the same consecutive bits. Specifically, it requires that an identical predefined bit sequence is embedded in all CAN frames transmitted on the bus. To achieve this, all data frames on the bus are set as the extended frame format with a 29-bit identifier. A predefined bit sequence which is 18-bit long is placed at the extended identifier field. Subsequently, the voltages of the pre-assigned bit sequence for every newly received CAN frame are sampled and measured. Obviously, reprogramming for all active ECUs on CAN bus is required to add the predefined bit sequence to each CAN frame. Besides, it can NOT be applied to the natural extended frames (the extended part of the identifier is already occupied). These limitations hinder its deployment on real production vehicles.

References [7, 16, 17] improved the process for extracting signal characteristics based on voltage. More specifically, SCISSION [16] and EASI [17] divide the string of consecutive dominant bits into three parts, which are the rising edge, the falling edge, and the holding edge of the dominant state part (as shown in Fig. 10). The approach adopted by VoltageIDS [7] is similar except that it only considers the 1-bit length holding edge of the dominant state part. Next, the voltage is measured and gathered separately for each part. The significant features of voltage on the rising edge and falling edge could be suppressed without such actions considering that their length is too short (resulting in much fewer samples) compared to the holding edge. By doing so, the combined features including the voltage measurements as well as the signal shape can be extracted to better represent

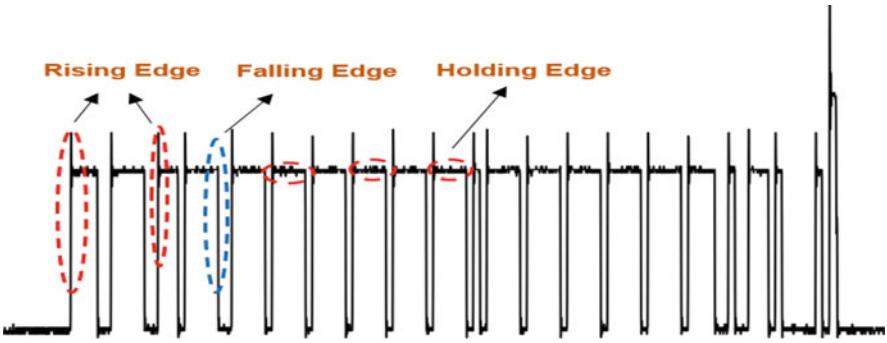

Fig. 10 An example of CAN electrical signal

the signal characteristics of the sending ECU. In addition, EASI [17] designs a low-cost solution to improve the efficiency of preprocessing the electrical signal. The authors optimized the interval for signal sampling and introduce the Random Interleaved Sampling technique, which greatly reduces the sampling rate and system resource requirements. It can favor the development on in-vehicle network.

6.2.1.2 Using Timing as Signal Characteristics

Apart from the voltage characteristics, the timing characteristics of CAN electrical signal can also be utilized to construct the intrusion detection system. Most existing works [3, 19, 36] using timing characteristics estimate the clock skew based on the periodic CAN traffic. Considering that most CAN frames are transmitted nearly periodically, the skew in the clock of the transmitter can be estimated by the difference between the expected and the actual arrival time of periodic traffic. From the observations of CAN traffic from real vehicles, the actual period of many frames can fluctuate a little wild and some frames might stop transmission for a while in real cases [19]. To mitigate these challenges, CANvas [19] improves the estimation process by introducing the concept of hyper-period. However, the dependency on periodic traffic still remains which makes it unavailable to aperiodic frames or sporadic frames.

BTMonitor [37] employs the timing characteristics of a single CAN frame to build the intrusion detection model, by which the dependency on periodic traffic can be cut. The insight behind BTMonitor is that the electrical signal length which is driven by the hardware of the transmitter can reflect the timing characteristics of sending ECU. Thus, the clock skew can be derived by measuring the signal length from a single frame, making the signal preprocessing process independent of the periodic traffic. To capture the signal which can accurately reveal the hardware characteristics of sending ECU, the signal segment in the identifier field shall be excluded from the measurement process. The reason is that multiple ECUs on the bus might initiate the transmission simultaneously and compete for the right of bus

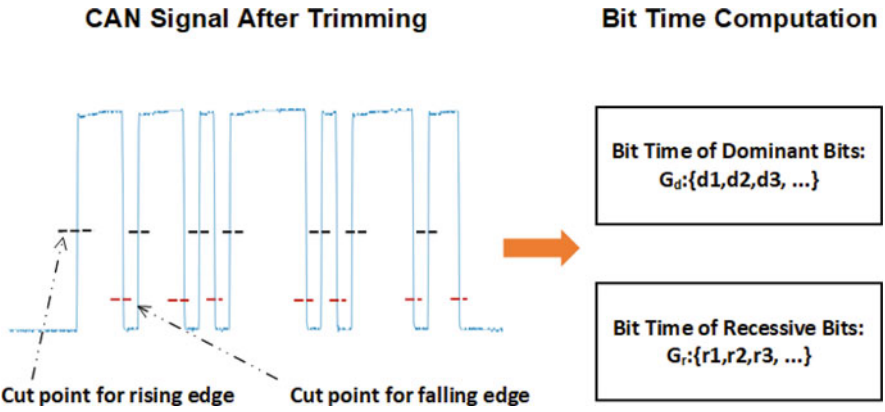

Fig. 11 Signal characteristics derivation in BTMonitor [37]

usage during the arbitration process. Thus, the signal segment in the identifier field might reflect the signal characteristics of more ECUs beyond the transmitter. For the same reason, the duration of the signal during the acknowledge field is also excluded to remove the effect on signal length by other nodes.

After the trimming, BTMonitor divides the remaining signal into different segments along the consecutive edges. These signal segments can be referred to as two categories, which are dominant bits and recessive bits. To reduce the requirement for a high sampling rate for measuring device, BTMonitor takes the rising edge as well as the falling edge into consideration. The point to divide the signal on the rising edge is different from the point on the falling edge. Finally, BTMonitor measures and computes a corresponding bit time for each signal segment. The calculated bit time of each category is gathered up to form data samples that represent the timing characteristics of sending ECU. The process is shown in Fig. 11.

6.2.2 Feature Extraction and Intrusion Detection

Once the signal characteristics are obtained, the preprocessed data is fed into the next phase to extract statistical features in the time and/or frequency domain. The extracted features can be used as device fingerprints to identify different ECUs. As an example, BTMonitor adopts eight statistical features in the time domain for each of the categories of dominant bits and recessive bits, i.e. 16 statistical features in total to represent one received data sample. The selected features are shown in Table 2. Then, the generated fingerprint is input into the classifier for intrusion detection.

During the training phase, supervised learning algorithms along with labeled samples (training datasets) are used to train the classifier. During the runtime phase, the newly derived device fingerprints are fed into the trained classifier to predict its

Table 2 Selected features in time domain by BTMonitor [37]. x represents bit time. N is the number of data

Feature	Description
Mean	$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x(i)$
Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x(i)-\overline{x})^2}$
Variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x(i)-\overline{x})^2$
Skewness	$\gamma = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\overline{x}}{\sigma})^3$
Kurtosis	$\beta = \frac{1}{N}\sum_{i=1}^{N}(\frac{x(i)-\overline{x}}{\sigma})^4 - 3$
RMS (Root mean square)	$A = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x(i)^2}$
Highest value	$H = \max(x(i))$
Energy	$en = \frac{1}{N}\sum_{i=1}^{N} x(i)^2$

actual sending ECU. If the predicted sending ECU is inconsistent with the legitimate sending ECU, an intrusion is alarmed. Varied machine learning algorithms, such as Linear Support Vector Machines [6, 7, 17], Packed Decision Trees (BDT) [6, 7], Logistic Regression [16, 17, 37], Naive Bayes Classifiers [17], Neural Networks [6], etc. are used to generate classification models.

6.3 Summary

The difference in the signal can be utilized to generate the fingerprint for the ECU. The derived fingerprint can then provide the ability to authenticate the sending ECU and detect intrusion. We summarize the overall process of machine learning-based methods in three steps, which are signal characteristics derivation, feature extraction, and intrusion detection respectively. These methods can be divided into two categories based on the exploited signal characteristics, which are the methods using signal voltage and the methods using signal timing. The general process of feature extraction and intrusion detection in both categories is similar. The statistical features in the time and/or frequency domain are extracted from the extracted signal characteristic and combined as the device fingerprint. Finally, popular machine learning algorithms are utilized as the classification model to detect intrusion. The signal characteristics-based methods can provide high security for automotive CAN bus considering that the fingerprint is derived from the inherent physical characteristics and is hard to be duplicated. However, how to obtain an effective but stable fingerprint from the mutable and sensitive signal is the major challenge to be solved.

7 Conclusion

CAN is the most important communication protocol for the current in-vehicle network and aged for over 35 years. With the rapid development of connectivity

and intelligence for today's vehicles, the underlying internal communication system is updated accordingly to manage the future's needs. In this chapter, we firstly take a discussion about the traditional and tomorrow in-vehicle network architecture as well as the advantages brought by the new architecture, aiming to provide a whole picture of how in-vehicle network evolves. The necessity of protecting CAN for ensuring the safety of vehicles is emphasized to motivate the research on defending techniques. Subsequently, we introduce different approaches to detect intrusion by categories based on the domain knowledge used in machine learning algorithms.

The variables with specific physical meanings in CAN can respond to a physical phenomenon in a correlated way. These observations can be exploited to detect intrusion which is detailed in semantic-based intrusion detection methods. Further studies reveal that the latent relationship can be extracted without requiring semantics of CAN frames. Literal-based intrusion detection methods provide a detailed description of how it works from two aspects according to whether the attack sample is required for training the model. Timing-based intrusion detection methods exploit the fact that most CAN traffic is triggered periodically thus the timing of CAN traffic can exhibit specific patterns. However, the main drawback is that it cannot deal with attack scenarios in which the timing characteristics are not affected. At last, signal characteristics-based intrusion detection provides a novel way of fingerprinting the ECUs by measuring the low-level characteristics of CAN electrical signals. Considering it is derived from the unique and inherent hardware characteristics, it can provide high security for in-vehicle CAN bus.

In conclusion, we survey the machine learning-based intrusion detection methods for automotive CAN bus and provide the introduction from the perspective of the exploited domain knowledge. We hope this chapter can help the interested reader to understand and grasp the status and research of machine learning-based intrusion detection methods comprehensively.

References

1. Akowuah, F., Kong, F.: Physical invariant based attack detection for autonomous vehicles: Survey, vision, and challenges. In: 2021 Fourth international conference on connected and autonomous driving (MetroCAD), pp. 31–40. IEEE, Piscataway (2021)
2. Bakker, E., Nyborg, L., Pacejka, H.B.: Tyre modelling for use in vehicle dynamics studies. SAE Trans. **96**, 190–204 (1987)
3. Cho, K.T., Shin, K.G.: Fingerprinting electronic control units for vehicle intrusion detection. In: 25th USENIX conference on security symposium (USENIX Security), pp. 911–927. USENIX Association, Berkeley (2016)
4. Cho, K., Shin, K.G.: Viden: Attacker identification on in-vehicle networks. In: 2017 ACM conference on computer and communications security (CCS), pp. 1109–1123. ACM, New York (2017)
5. Cho, K.T., Shin, K.G., Park, T.: CPS approach to checking norm operation of a brake-by-wire system. In: ACM/IEEE sixth international conference on cyber-physical systems (ICCPS), pp. 41–50. ACM, New York (2015)

6. Choi, W., Jo, H.J., Woo, S., Chun, J.Y., Park, J., Lee, D.H.: Identifying ecus using inimitable characteristics of signals in controller area networks. IEEE Trans. Veh. Technol. **67**(6), 4757–4770 (2018)
7. Choi, W., Joo, K., Jo, H.J., Park, M.C., Lee, D.H.: Voltageids: low-level communication characteristics for automotive intrusion detection system. IEEE Trans. Inf. Forens. Secur. **13**(8), 2114–2129 (2018)
8. Di Natale, M., Zeng, H., Giusto, P., Ghosal, A.: Understanding and using the controller area network communication protocol: theory and practice. Springer Science & Business Media (2012)
9. Groza, B., Murvay, P.S.: Efficient intrusion detection with bloom filtering in controller area networks. IEEE Trans. Inf. Forens. Secur. **14**(4), 1037–1051 (2018)
10. Guo, F., Wang, Z., Du, S., Li, H., Zhu, H., Pei, Q., Cao, Z., Zhao, J.: Detecting vehicle anomaly in the edge via sensor consistency and frequency characteristic. IEEE Trans. Veh. Technol. **68**(6), 5618–5628 (2019)
11. He, T., Zhang, L., Kong, F., Salekin, A.: Exploring inherent sensor redundancy for automotive anomaly detection. In: 2020 57th ACM/IEEE design automation conference (DAC), pp. 1–6. IEEE, Piscataway (2020)
12. Hoppe, T., Kiltz, S., Dittmann, J.: Security threats to automotive can networks–practical examples and selected short-term countermeasures. In: International conference on computer safety, reliability, and security (SAFECOMP), pp. 235–248. Springer, Berlin, Heidelberg (2008)
13. Javed, A.R., Usman, M., Rehman, S.U., Khan, M.U., Haghighi, M.S.: Anomaly detection in automated vehicles using multistage attention-based convolutional neural network. IEEE Trans. Intell. Transp. Syst. **22**(7), 4291–4300 (2020)
14. Javed, A.R., Ur Rehman, S., Khan, M.U., Alazab, M., Reddy, T.: CANintelliiDS: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. IEEE Trans. Netw. Sci. Eng. **8**(2), 1456–1466 (2021)
15. Kavousi-Fard, A., Dabbaghjamanesh, M., Jin, T., Su, W., Roustaei, M.: An evolutionary deep learning-based anomaly detection model for securing vehicles. IEEE Trans. Intell. Transp. Syst. **22**(7), 4478–4486 (2020)
16. Kneib, M., Huth, C.: Scission: signal characteristic-based sender identification and intrusion detection in automotive networks. In: ACM SIGSAC conference on computer and communications security (CCS), pp. 787–800. ACM, New York (2018)
17. Kneib, M., Schell, O., Huth, C.: EASI: Edge-based sender identification on resource-constrained platforms for automotive networks. In: The 2020 network and distributed system security symposium (NDSS), pp. 1–16. ISOC, San Diego (2020)
18. Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S.: Experimental security analysis of a modern automobile. In: IEEE symposium on security and privacy (S&P), pp. 447–462. IEEE, Piscataway (2010)
19. Kulandaivel, S., Goyal, T., Agrawal, A.K., Sekar, V.: Canvas: fast and inexpensive automotive network mapping. In: The 28th USENIX conference on security symposium (USENIX Security), pp. 389–405. USENIX Association, Berkeley
20. Li, H., Zhao, L., Juliato, M., Ahmed, S., Sastry, M.R., Yang, L.L.: Poster: intrusion detection system for in-vehicle networks using sensor correlation and integration. In: The 2017 ACM SIGSAC conference on computer and communications security (CCS), pp. 2531–2533 (2017)
21. Longari, S., Valcarcel, D.H.N., Zago, M., Carminati, M., Zanero, S.: CANnolo: an anomaly detection system based on LSTM autoencoders for controller area network. IEEE Trans. Netw. Serv. Manag. **18**(2), 1913–1924 (2020)
22. Marchetti, M., Stabili, D.: Read: reverse engineering of automotive data frames. IEEE Trans. Inf. Forens. Secur. **14**(4), 1083–1097 (2018)
23. Murvay, P.S., Groza, B.: Source identification using signal characteristics in controller area networks. IEEE Signal Process. Lett. **21**(4), 395–399 (2014)

24. Müter, M., Asaj, N.: Entropy-based anomaly detection for in-vehicle networks. In: IEEE intelligent vehicles symposium, pp. 1110–1115. IEEE, Piscataway (2011)
25. Olufowobi, H., Young, C., Zambreno, J., Bloom, G.: Saiducant: specification-based automotive intrusion detection using controller area network (CAN) timing. IEEE Trans. Veh. Technol. **69**(2), 1484–1494 (2019)
26. Song, H.M., Woo, J., Kim, H.K.: In-vehicle network intrusion detection using deep convolutional neural network. Veh. Commun. **21**, 100198 (2020)
27. Suda, H., Natsui, M., Hanyu, T.: Systematic intrusion detection technique for an in-vehicle network based on time-series feature extraction. In: 2018 IEEE 48th international symposium on multiple-valued logic (ISMVL), pp. 56–61. IEEE (2018)
28. Sun, H., Chen, M., Weng, J., Liu, Z., Geng, G.: Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism. IEEE Trans. Veh. Technol. **70**(10), 10880–10893 (2021)
29. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: 2016 IEEE international conference on data science and advanced analytics (DSAA), pp. 130–139. IEEE (2016)
30. Tomlinson, A., Bryans, J., Shaikh, S.A., Kalutarage, H.K.: Detection of automotive can cyber-attacks by identifying packet timing anomalies in time windows. In: 2018 48th Annual IEEE/IFIP international conference on dependable systems and networks workshops (DSN-W), pp. 231–238. IEEE (2018)
31. van Wyk, F., Wang, Y., Khojandi, A., Masoud, N.: Real-time sensor anomaly detection and identification in automated vehicles. IEEE Trans. Intell. Transp. Syst. **21**(3), 1264–1276 (2020). https://doi.org/10.1109/TITS.2019.2906038
32. Wang, R., Kong, F., Sudler, H., Jiao, X.: Brief industry paper: Hdad: hyperdimensional computing-based anomaly detection for automotive sensor attacks. In: 2021 IEEE 27th real-time and embedded technology and applications symposium (RTAS), pp. 461–464. IEEE (2021)
33. Wasicek, A., Pesé, M.D., Weimerskirch, A., Burakova, Y., Singh, K.: Context-aware intrusion detection in automotive control systems. In: 5th ESCAR USA conference, pp. 21–22 (2017)
34. Xie, G., Yang, L.T., Yang, Y., Luo, H., Li, R., Alazab, M.: Threat analysis for automotive can networks: a GAN model-based intrusion detection technique. IEEE Trans. Intell. Transp. Syst. **22**(7), 4467–4477 (2021)
35. Zeng, W., Khalid, M.A., Chowdhury, S.: In-vehicle networks outlook: achievements and challenges. IEEE Commun. Surv. Tutorials **18**(3), 1552–1571 (2016)
36. Zhao, Y., Xun, Y., Liu, J.: Clockids: A real-time vehicle intrusion detection system based on clock skew. IEEE Internet Things J. **9**, 15593 (2022)
37. Zhou, J., Joshi, P., Zeng, H., Li, R.: Btmonitor: bit-time-based intrusion detection and attacker identification in controller area network. ACM Trans. Embed. Comput. Syst. **18**(6), 1 (2020)
38. Zhou, J., Xie, G., Zeng, H., Zhang, W., Yang, L.T., Alazab, M., Li, R.: A model-based method for enabling source mapping and intrusion detection on proprietary can bus. IEEE Trans. Intell. Transp. Syst. (2022)
39. Zhu, K., Chen, Z., Peng, Y., Zhang, L.: Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM. IEEE Trans. Veh. Technol. **68**(5), 4275–4284 (2019)