Al-based intrusion detection systems for in-vehicle networks: a survey.

RAJAPAKSHA, S., KALUTARAGE, H., AL-KADRI, M.O., PETROVSKI, A., MADZUDZO, G. and CHEAH, M.

2023

AI-Based Intrusion Detection Systems for In-Vehicle Networks: A Survey

SAMPATH RAJAPAKSHA and HARSHA KALUTARAGE, Robert Gordon University
M. OMAR AL-KADRI, Birmingham City University
ANDREI PETROVSKI, Robert Gordon University
GARIKAYI MADZUDZO and MADELINE CHEAH, Horiba Mira Ltd.

The Controller Area Network (CAN) is the most widely used in-vehicle communication protocol, which still lacks the implementation of suitable security mechanisms such as message authentication and encryption. This makes the CAN bus vulnerable to numerous cyber attacks. Various Intrusion Detection Systems (IDSs) have been developed to detect these attacks. However, the high generalization capabilities of Artificial Intelligence (AI) make AI-based IDS an excellent countermeasure against automotive cyber attacks. This article surveys AI-based in-vehicle IDS from 2016 to 2022 (August) with a novel taxonomy. It reviews the detection techniques, attack types, features, and benchmark datasets. Furthermore, the article discusses the security of AI models, necessary steps to develop AI-based IDSs in the CAN bus, identifies the limitations of existing proposals, and gives recommendations for future research directions.

CCS Concepts: • **Security and privacy** → **Network security**; **Intrusion detection systems;**

Additional Key Words and Phrases: Intrusion Detection System (IDS), in-vehicle network, Controller Area Network (CAN), machine learning, automotive cybersecurity

ACM Reference format:
Sampath Rajapaksha, Harsha Kalutarage, M. Omar Al-Kadri, Andrei Petrovski, Garikayi Madzudzo, and Madeline Cheah. 2023. AI-Based Intrusion Detection Systems for In-Vehicle Networks: A Survey. *ACM Comput. Surv.* 55, 11, Article 237 (February 2023), 40 pages.
https://doi.org/10.1145/3570954

1 INTRODUCTION

Modern automobiles are becoming intelligent, complex, and highly connected. In 1980, a vehicle had just 1% of electronic equipment in comparison to its mechanical counterparts. However, today, electronic components have increased up to 50% [148]. This will continue to increase with the advent of autonomous cars, which will rely on powerful computer systems, a range of sensors, networking, and satellite navigation, all of which require electronics. Furthermore, modern vehicles embody software that exceeds 100 million lines of code, and it is expected to grow beyond

Authors' addresses: S. Rajapaksha, H. Kalutarage, and A. Petrovski, Robert Gordon University, Garthdee Road, Aberdeen, AB10 7QB, UK; emails: {s.rajapaksha, h.kalutarage, a.petrovski}@rgu.ac.uk; M. O. Al-Kadri, Birmingham City University, Millennium Point, Curzon Street, Birmingham, B4 7XG, UK; email: omar.alkadri@bcu.ac.uk; G. Madzudzo and M. Cheah, Horiba Mira Ltd, Watling Street (A5), Nuneaton, Warwickshire, CV10 0TU, UK; emails: {garikayi.madzudzo, madeline.cheah}@horiba-mira.com.

0360-0300/2023/02-ART237 $15.00
https://doi.org/10.1145/3570954

300 million lines of code in the near future [24]. Software on modern automobiles run on 70 to 100 microprocessor-based **Electronic Control Units (ECUs)** that are networked throughout the vehicle [24]. More than 125 million cars with embedded connectivity will be shipped worldwide between 2018 and 2022 [125]. This connectivity to the outside world makes modern automobiles part of the **Internet of Things (IoT)** [148]. In addition to a large number of ECUs, modern vehicles are equipped with multiple sensors, actuators, cameras, radars, and communication devices, among others [2, 22]. These systems are intended to improve performance, efficiency, intelligent services, and safety for automobile users by collecting and interpreting different data [22]. However, at the same time, these systems make modern automobiles significantly more complex.

Vehicle networks facilitate communication between these systems, which can be classified into two categories: external and internal networks. The internal network is also known as the in-vehicle or intra-vehicle network. Based on the communication type, the external network can be categorized as V2V (Vehicle to Vehicle) and V2I (Vehicle to Infrastructure), in which both are also called **Vehicular Ad Hoc Networks (VANETs)**. V2X (Vehicle to Everything) is sometimes used to represent both V2V and V2I [148]. In addition to their designated functions, ECUs communicate with other ECUs. This communication occurs through various standard automobile in-vehicle communication protocols, such as **Controller Area Network (CAN)**, FlexRay, LIN (Local Interconnect Network), and **Media Oriented System Transport (MOST)** [85]. Among these protocols, CAN is considered to be the de facto protocol for in-vehicle communication [2]. In contrast, VANET is an ad hoc network that facilitates direct communication between vehicles without having a fixed infrastructure [22].

Increased connectivity and complexity of modern automobiles have created more attack surfaces to security threats. Examples of such attack surfaces are V2X communication, telematics service, Bluetooth connection, and the **On-Board Diagnostic (OBD)** port [148]. Although vehicle networks provide various benefits such as efficiency, low cost, and safety, most of these networks are vulnerable to cyber attacks, especially the CAN bus, which is used by the majority of vehicles for in-vehicle communication. A few CAN bus vulnerabilities are absence of authentication, broadcast transmission, lack of encryption, an ID-based priority scheme, and available interfaces [98]. Researchers performed various experimental attacks to exploit vehicular network vulnerabilities. Hoppe et al. [59] performed frame sniffing and replay attacks in a simulated environment to gain control over window lift, warning light, and airbag systems. Further, Koscher et al. [82] performed different attacks on a real car, and they were able to control different modules like body control module, radio, and engine.

Due to various vulnerabilities and potential cyber attacks, considerable efforts had been made to protect vehicles from security threats. Both detection and prevention mechanisms can be used to identify or prevent cybercrimes. However, detection strategies are more realistic in terms of the operational and economical realities [120]. Hence, as a reactive security mechanism, current literature has focused on developing **Intrusion Detection Systems (IDSs)** for **In-Vehicle Networks (IVNs)**. Based on the detection strategy, IDS can be categorized as signature-based detection and anomaly-based detection [115]. Because of certain limitations in signature-based IDS, such as the inability to detect novel attacks, and the requirement for frequently updating the known-attack database, anomaly-based detection approaches captured the attention of previous researchers due to advantages like the capability of detecting novel attacks [116]. Current literature has classified IDS into more sub-sections, such as fingerprint based, parameter monitoring based, information theory based, and **Machine Learning (ML)** based [175]. Moreover, the majority of IDSs are designed to detect anomalies in the CAN bus as opposed to other vehicle network protocols.

This work focuses on the application of **Artificial Intelligence (AI)** techniques on IDSs of IVNs. The term *AI* represents both traditional ML and **Deep Learning (DL)** techniques in this work. This

survey article reviews state-of-the-art works published between 2016 and 2022 (August), focusing on detection methods, evaluation/benchmark datasets, attack types, and performance evaluation. The contributions of this article can be summarized as follows:

(1) We review and classify state-of-the-art AI-based in-vehicle IDSs considering their detection algorithms, features, datasets, targeted attack types, and performance evaluation along with comprehensive summary tables (see Tables 3–7).
(2) A novel AI-based IDS taxonomy is introduced that focuses on attack types, CAN bus data frame features, and AI algorithms for IVNs (CAN bus) to categorize reviewed works based on their detection features and algorithms.
(3) A review of benchmark datasets available to train and evaluate algorithms and the steps to follow in developing an AI-based attack detection in the CAN bus are provided.
(4) Based on reviewed works, we identify and discuss limitations of current approaches for securing IVNs (CAN bus).
(5) A set of possible future research directions is discussed.

The rest of the article is organized as follows. In Section 2, the existing surveys on the subject are discussed. Section 3 explains the methodology used for this study. In Section 4, an overview of common attacks on IVNs is provided. Section 5 introduces AI-based IDS taxonomy for IVNs and reviews existing works based on detection features and algorithms. Section 6 discusses the security of AI models. Section 7 presents the findings, AI-based attack detection steps, and potential future research directions. Finally, Section 8 concludes the article.

2 RELATED WORK

Surveys that focus on AI-based IDSs in mainstream IT networks are not relevant to our work, because the behavior of the CAN network is quite different from traditional IT networks and the message specifications are confidential for car manufacturers. Therefore, solutions proposed for those networks knowing all specifications might not be suitable for IVNs. Several surveys of IDSs for vehicle networks are available in the literature. Rajbahadur et al. [135] produced a taxonomy with three categories, nine sub-categories, and 38 dimensions to review anomaly detection techniques of connected vehicles. Even though this study provided a comprehensive categorization of IDSs, individual paper summaries and implementation techniques are not discussed. Al-Jarrah et al. [2] mainly focused on in-vehicle IDSs by referring to 44 prior works. In this work, IDSs are categorized into three categories as flow based, payload based, and hybrid. Finally, they discussed some of the research challenges and gaps in in-vehicle IDSs. Loukas et al. [104] provided a comprehensive taxonomy focusing on different types of vehicles (aircraft, land vehicles, and watercraft). Statistical, ML, and rule-based IDSs were discussed under the audit technique. This paper reviewed only 13 ML-based IDSs for CAN bus that were published between 2011 and 2017. Lokman et al. [101] discussed the vulnerabilities of the CAN network and potential attacks. IDSs found in the literature were discussed based on detection approaches, deployment strategy, attacking techniques, and technical challenges. However, this work cannot be considered as a comprehensive survey compared to ours, as only five works were discussed under the ML-based approach. Dupont et al. [37] categorized a few existing CAN IDSs based on three dimensions. These include the number of frames, data used for the detection, and detection mechanism. A recent survey [175] classified 20 in-vehicle IDSs into four areas such as fingerprint based, parameter monitoring based, information theory based, and ML based. The authors briefly discussed datasets used in previous works. However, this study discussed only 9 ML-based IDSs published between 2012 and 2018. In both [160, 179], the authors categorized intrusion detection in CAN network into signature-based and anomaly-based methods. Current cryptographic and

Table 1. Comparison of Surveys for In-Vehicle IDSs

Survey	Time Range	AI-Based Works	AI-Based Taxonomy	Review of Benchmark Datasets	Feature Selection	Evaluation Data	Evaluation Results
[179]	2016	2			X		X
[101]	2016	5			X	X	X
[104]	2011–2017	13			X	X	X
[135]	2015–2017	9				X	X
[175]	2012–2018	9		X		X	X
[2]	2015–2018	23			X	X	X
[37]	2016–2018	7			X	X	
[160]	2016–2018	12			X	X	X
[7]	2014–2020	14			X	X	X
[74]	2020–2022	33		X			
This survey	2016–2022	102	X	X	X	X	X

IDS approaches were discussed and compared in the work of Aliwa et al. [7]. Similar to Young et al. [179], the authors classified in-vehicle IDSs into signature based and anomaly based. IDSs based on anomaly detection were further classified as statistical, ML, rule-based, and physical fingerprinting methods. This work briefly discussed 14 ML-based IDSs published between 2014 and 2020. However, most of these works belong to 2018 or earlier (12 out of 14), and the latest works were not included. Karopoulos et al. [74] provided a unified taxonomy for IVN IDS. They identified 33 ML-based IDSs designed for IVNs. Individual paper summaries were not included in this work.

Several limitations can be identified in existing reviews and surveys for IVNs. These include a limited focus on the adopted AI techniques and a lack of discussion of recent state-of-the-art works. Lack of in-depth analysis of AI-based detection techniques, review of benchmark datasets, result evaluation, feature importance to detect different attacks, and threats to AI-based models also can be identified as significant limitations of existing literature. To the best of the authors' knowledge, there are no surveys available that focus on AI-based IDS for IVNs. This survey is the first to review AI-based IDSs for IVNs (particularly the CAN bus) with a novel AI-based IDS taxonomy. The aforementioned shortcomings are addressed in this survey, and therefore this work is unique. Table 1 provides a comparison between this survey and other available surveys for in-vehicle IDSs, highlighting the contributions of this work.

3 METHODOLOGY

This section discusses the scope and survey method used in this work. To ensure scope focus, this survey does not discuss other related areas such as VANETs, IoT networks, Mobile Ad hoc Networks (MANETs), and cryptography solutions (CAN frame authentication and encryption).

3.1 Survey Method

3.1.1 Protocol. The papers reviewed in this study were selected using PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses) [93] protocol. Figure 1 illustrates the PRISMA selection process.

3.1.2 Eligibility Criteria.

- Paperspublished between 2016 and 2022 (August) were selected based on the scope of this survey. Papers should make use of AI algorithms to detect attacks/anomalies in IVNs.
- Google Scholar was used for the keyword search. The keywords used were "in-vehicle intrusion detection machine learning,", "in-vehicle attack detection machine learning," "in-vehicle intrusion detection," "in-vehicle machine learning attack," "in-vehicle cybersecurity survey," "controller area network IDS," "controller area network attack detection," "controller area

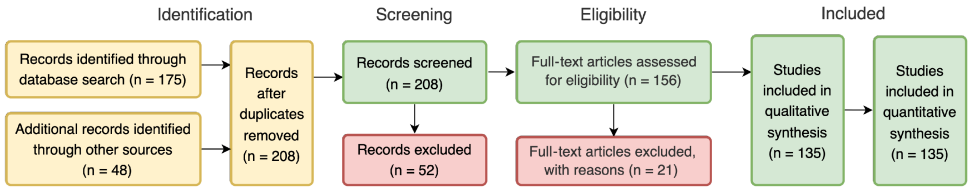

Fig. 1. PRISMA protocol used for sampling.

network machine learning," and "in-vehicle network anomaly detection." These keywords were selected considering the focus of this paper.

- Backward and forward snowballing [172] and recommendations given by Mendeley Reference Manager were also used to collect all relevant references.
- Papers were included or excluded by reading the abstract and introduction considering the scope of this review. The final set of papers was selected so that each category listed in the taxonomy had at least one and preferably a few representative papers.

3.1.3 Risk of Bias. Google Scholar is considered a good starting point, as it helps avoid bias for any specific publisher [172]. This study selected Google Scholar as the search engine. Although this is a comprehensive review, there will still be good papers not selected, as they are out of defined eligibility criteria. Only the papers written in the English language were considered. Due to these limitations, this work may have overlooked some important works.

4 BACKGROUND

This section provides a brief introduction to IVN protocols along with their vulnerabilities to cyber attacks. Common attack types and characteristics of CAN bus data frames are also discussed in this section.

4.1 In-Vehicle Networks

IVNs facilitate communication within the vehicle. Among different network protocols, CAN is the most common network protocol used for in-vehicle communication due to several benefits, such as low cost, speed, light weight, robustness [14], and simplified installation [7]. Different ECUs communicate with each other through the CAN network, and it is considered as a messaged-based protocol [101]. High-speed CAN bus and low-speed CAN bus are defined based on data rates. The bit rate of high-speed CAN bus ranges from 125 Kbps to 1 Mbps, whereas the low-speed CAN bus ranges from 5 to 125 Kbps. CAN bus supports a payload up to 8 bytes. Time-critical modules such as engine control and transmission control are connected to a high-speed CAN bus, whereas less time-critical modules such as door control and light control are connected to the low-speed CAN bus. These two buses are connected through a gateway [98]. CAN Flexible Data (CAN-FD) supports bit rate up to 8 Mbps with a maximum payload of 64 bytes [39]. Based on the functions and required communication speed, the network in the vehicle can be divided into four domains [63, 122]: the power train domain includes time-critical applications such as the engine controller and transmission; the chassis domain includes steering control, brake control, and suspension, which are also time-critical applications; the body domain, which includes functions such as light control, windows, and seats; and the infotainment domain controls communication and multimedia functions such as audio/video, navigation, and display.

Despite the benefits offered by the CAN bus, it is vulnerable to cyber attacks due to various vulnerabilities [98], such as the following:

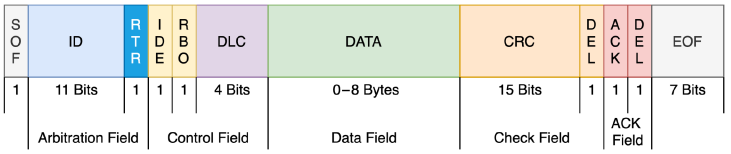

Fig. 2. CAN bus data frame.

- *No authentication*: Since the CAN bus has no authentication, any ECU could transmit a frame with a CAN ID that belongs to another ECU.
- *Broadcast domain*: The CAN bus is a broadcast domain. All nodes receive CAN frames transmitted through the network. A compromised node can listen to all messages broadcast in the CAN network.
- *No encryption*: CAN messages are not encrypted considering the time constraints. Cyber attackers can collect and analyze these messages easily (sniffing attack).
- *ID-based priority*: The CAN network uses an ID-based priority to handle multiple concurrent messages. The lower the ID, the higher the priority. Malicious nodes can continuously transmit frames with lower IDs, thus creating a **Denial-of-Service (DoS)** attack.

FlexRay is a time-triggered in-vehicle communication protocol introduced in 2000 by the FlexRay consortium. It has higher bandwidth and more fault tolerance capabilities with a maximum baud rate of 10 Mbps and a payload length of 254 bytes. FlexRay is more expensive compared to the CAN network. However, it is more vulnerable to DoS and spoofing attacks [79].

MOST is an IVN that transmits multimedia data. MOST uses bandwidths of 25, 50, and 150 Mbps. The use of higher bandwidth ranges makes it more suitable for multimedia. LIN is an inexpensive IVN protocol used in less critical applications such as seat belts, door locks, mirrors, batteries, and temperature monitoring.

4.2 CAN Bus Data Frame

A CAN frame has a specific message structure defined in a database-like file known as the **DataBase CAN (DBC)** file. This is confidential proprietary of the vehicle manufacture and contains all necessary information of a specific vehicle related to ECUs, CAN messages, signals, message IDs, message frequency, and payload of the CAN frame [13]. Further, the DBC file specifies whether the CAN ID is periodic or event driven [107]. There are four CAN frame types that can be identified: data frame, remote frame, overload frame, and error frame [14]. This article focuses on the CAN data frame, as all works discussed here used only the data frame in their IDSs to derive features. The CAN data frame consists of seven fields that support data transmission from the transmitter to the receiver (ECUs). Figure 2 illustrates the fields of a CAN data frame with respective sizes. Seven fields of the CAN data frame are described next:

- *Start of frame (SOF)*: Start of frame specifies the beginning of a CAN frame. It uses the dominant bit (logical 0) to inform the beginning of CAN frame transmission to other nodes.
- *Arbitration field (CAN-ID)*: Arbitration field (arbitration ID or simply ID) is used to prioritize the message when multiple ECUs concurrently transfer messages. For instance, two nodes with CAN IDs 0x0D0 (000011010000 in binary) and 0x2E1 (001011100001 in binary) try to transmit messages simultaneously. Node with ID 0x0D0 will gain the bus access to transit the frame due to the lowest value (higher priority). Usually, CAN ID is 11 bits and the extended format has 29 bits. Remote Transmission Request (RTR) distinguishes the data frame and remote frame. Generally, each node (ECU) is assigned one or more IDs. However, the same ID cannot be used by two nodes (ID is unique for one node).

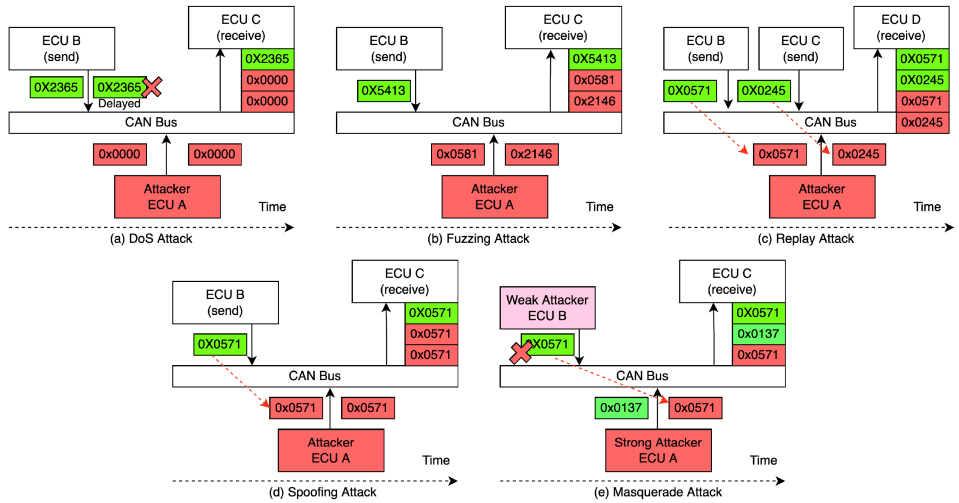

Fig. 3. Common attacks on IVNs.

- *Control field (DLC)*: Control field is a 6-bit field including data length code (4 bits) that is used to identify the length of the payload and two additional bits reserved for future use.
- *Data field*: Data field contains the actual information that needs to transmit on the CAN bus. This is also known as the payload of the CAN frame. This ranges from 0 to 8 bytes. Payload values contain sensor data, category data, constant data, or cyclical counter data [108].
- *CRC field*: CRC (cyclic redundancy code) is also known as the safety field. This is a 15-bit field followed by 1-bit CRC delimiter. This is used to check the frame validity.
- *Acknowledge field (ACK)*: This is known as the confirmation field consisting of 1-bit acknowledge and 1-bit delimiter fields. The ACK field is used to ensure that the receiver nodes receive the CAN frames.
- *End of frame (EOF)*: This specifies the end of the CAN frame.

4.3 Attacks on IVNs

IVNs are vulnerable to different cyber attack types. Attackers can access IVNs through physical access points (OBD-II port, USB, CD player, etc.), short-range wireless technologies (Bluetooth, RFID, etc.), and long-range wireless technologies (Wi-Fi, LTE, etc.). Some of the common attack types include the following:

- *DoS attack*: DoS attacks try to make communication services unavailable by sending a large number of frames. In the CAN bus, attackers can continuously send frames with low CAN IDs (highest-priority IDs) that disable communication between nodes. Koscher et al. [82] disabled the communication of individual components of the CAN bus using a DoS attack. Figure 3(a) shows the DoS attack in the CAN bus. Due to the high-priority CAN ID 0x0000, CAN ID 0x2365, which transmits by ECU B, will be delayed. This attack might increase or decrease the message frequency of the CAN bus.
- *Fuzzing attack*: In a fuzzing attack, the malicious node sends a large number of messages into the network using randomly generated ID, DLC, and CAN payloads that act as legitimate messages [62]. Chockalingam et al. [30] used hex-swapping and added Gaussian noise to the UNIX timestamps to create a fuzzy attack in CAN data. Fuzzing attack in the CAN bus is illustrated in Figure 3(b) (e.g., randomly generated CAN IDs 0x0581 and 0x2146 transmit

Table 2. Experimental Attacks on IVNs

Reference	Attack Type	Attack Vector	Violated Security Property	Affected Asset	Consequence
[111] 2015	Spoofing, CAN message injection	ECU	Authentication	Safety	Engine control, break control, steering wheel control
[112] 2016	Spoofing, CAN message injection	OBD	Authentication	Safety	Steering activation
[81] 2016	Information disclosure	OBD	Confidentiality	Safety and reliability	Driver distraction, turn on/off engine
[123] 2017	CAN message injection	ECU	Integrity	Safety	Damage integrated circuit and gateways
[38] 2018	Spoofing, brute force	ECU and OBD	Authentication	Safety	Airbag detonation
[23] 2019	CAN message injection	OBD and USB	Authentication	Safety	Control the vehicle remotely

by the attacker ECU A). Attackers can use the prior knowledge of CAN frames (CAN bus sniffing is used to read and analyse the data) or use this attack without having prior knowledge of CAN frames (as a black-box attack). A fuzzing attack will also increase the message frequency or change the ID sequences of the CAN bus.

- *Replay attack*: In replay attacks, attackers store and send valid messages at different times. For example, previous valid RPM values can be transmitted at a later stage. Even though this is an easy attack to launch, it can create serious safety risks for vehicles and passengers. Koscher et al. [82] used replay attacks to control the radio and number of body control module functions in the CAN bus. Figure 3(c) depicts the replay attack in which attacker ECU A transmits CAN IDs of ECU B and ECU C. Replay attack might cause to change the ID or payload sequences (or both).
- *Spoofing attack*: In a spoofing attack, the attacker targets specific CAN IDs to inject malicious messages. In some works [38, 64, 111, 112, 130], the authors used spoofing attacks in their experimental attacks on vehicle networks. Figure 3(d) illustrates the spoofing attack where attacker ECU A targets CAN ID 0x0571 of ECU B. This attack might change the frequency of targeted ID and ID sequences.
- *Masquerade attack*: This is also known as an impersonation attack whereby a compromised node impersonates another node. For example, the attacker can monitor and learn about message IDs and their frequencies of weak attacker node B (ID 0x0571). The attacker can then stop node B message transmission, paving the way for node A to transmit a fabricated message that represents node B [28]. In this case, the frequency of node B messages remains the same as before. However, node A will be the transmitter as shown in Figure 3(e). Woo et al. [173] performed an experimental masquerade attack using an Android smartphone on a mid-size car. This attack will not change the message frequency. However, the context of CAN IDs (sequence) or payload might change as a result of this attack.

Figure 3 only illustrates the change of CAN IDs. However, these attacks (fuzzing, reply, spoofing, and masquerade attacks) might change the CAN ID, CAN payload, or both at the same time. These aspects will be considered in the proposed taxonomy of this survey. Table 2 presents some of the experimental attacks that were carried out on IVNs. These threats not only pose information security or privacy issues but also directly affect the safety of drivers, passengers, and the surrounding environment. The rest of the article will discuss AI-based proposals to enhance the security of IVNs (particularly CAN bus).

5 AI-BASED IDSS FOR THE CAN BUS

IDSs can be categorized into two categories as signature-based detection and anomaly-based detection based on the detection technique. Signature-based detection has a low false-positive rate,

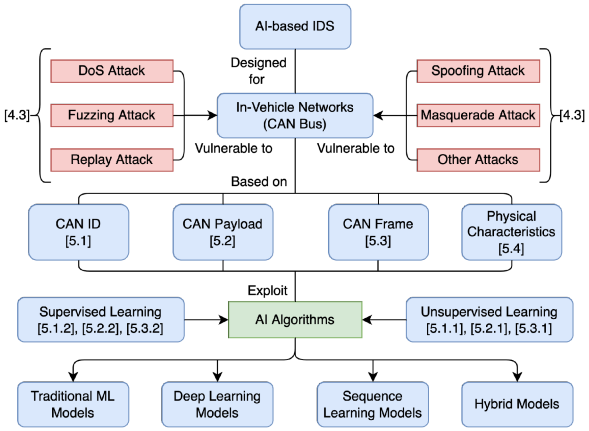

Fig. 4. AI-based IDS taxonomy for the CAN bus. The numbers indicate the sections that cover each topic of the taxonomy.

as it can be identified previously known attacks accurately. However, signature-based techniques fail to identify novel or previously unseen attacks. Anomaly-based detection techniques are capable of identifying novel attacks. AI-based techniques have been successfully used by researchers to identify cyber attacks in automobiles [71, 107, 168].

The classification of the papers reviewed in this survey is shown in Figure 4. CAN bus is vulnerable to cyber attacks such as DoS, fuzzing, reply, spoofing, and masquerade attacks. In the literature, authors experimented with other attack scenarios, such as USB firmware update, Over-the-Air (OTA) malicious update, chip tuning, anomalous speed, and RPM changes. Based on the objectives of the attacker, these attacks target CAN packet frequency or CAN payload or both fields. IDSs in the literature are designed to capture these changes in the CAN bus traffic. Hence, these properties were considered in the proposed taxonomy to classify the existing works. These IDSs developed based on features including CAN ID (ID), CAN Payload (Payload), CAN frame, and Physical characteristics. CAN frame represents feature combinations of ID, Payload, DLC, and time. Physical characteristics represent physical layer features such as voltage. IDSs focus on this work, and exploit AI algorithms such as traditional ML models, DL models, sequence learning models, and hybrid models. Various ML algorithms such as **Decision Tree (DT)**, **Random Forest (RF)**, **Support Vector Machine (SVM)**, Logistic Regression, **Naive Bayes (NB)**, and clustering have been studied for several decades and are known as shallow models or traditional ML models [97]. If an **Artificial Neural Network (ANN)** model is associated with one or two hidden layers, then it is considered as a shallow learning method [46]. DL-based models are highly effective for identifying complex patterns. Recently, automotive cyber security researchers have used DL-based models such as **Deep Neural Network (DNN)**, **Recurrent Neural Network (RNN)** including **Long Short-Term Memory (LSTM)** and **Gated Recurrent Unit (GRU)**, **Convolutional Neural Network (CNN)**, Deep Belief Network, Autoencoders, and **Generative Adversarial Nets (GAN)** to identify intrusions in vehicle networks. Sequence learning is a technique that is highly used in Natural Language Processing applications. For IVNs, CAN data can also be considered as sequential data or multivariate time series data. Most of the CAN IDs are transmitted based on defined time intervals or as a sequence of events. This property can be used to identify the anomalies in such sequences. N-gram and **Hidden Markov Model (HMM)**-based techniques have been used in the recent literature to identify anomalies in the CAN bus. The fourth category, hybrid models, which used AI-based and rule-based (specification-based) approaches, have their own strengths

Table 3. Summary of ID-Based Attack Detection in a CAN Bus Using Unsupervised and Supervised Learning

Reference	Model	Algorithm	Dataset	Attack	Strengths	Weaknesses
[134] 2022	Hybrid	GRU	Public real data (HCRL CH, HCRL SA, ROAD)	16 attacks including injection and masquerade attacks	High detection rate for a wide variety of attacks, near real-time detection	Limited capability to detect attacks on high-frequent aperiodic IDs
[58] 2022	Deep learning	GAN, Autoencoder	Public real data (HCRL CH)	DoS, fuzzy, RPM and gear spoofing	Near real-time detection	Limited to message injection attacks
[136] 2021	Traditional ML (Supervised)	SVM, KNN	Public real data (HCRL CH)	DoS, fuzzy, RPM spoofing	Feature extraction using benign data	Low detection rate for spoofing attack
[32] 2022	Deep learning (Supervised)	CNN	Public real data (HCRL CH)	DoS, fuzzy, RPM spoofing	High detection rate for injection attacks	High detection latency

The dataset description can be found in Section 5.5.1. The complete table is available in the supplementary material associated with this article.

and limitations. Generally, rule-based detection techniques have a low false-positive rate and high efficiency. AI-based techniques can identify unknown attacks better than rule-based techniques, even though they require more computing resources.

Both supervised and unsupervised learning can be used to train these algorithms. In supervised training, the algorithm learns based on the labeled data, whereas unsupervised training learns by understanding the behavior, structure, and distribution of the data [69]. In IDSs, unsupervised learning uses only benign data (also referred as one class) to train the algorithm and defined a threshold to detect anomalies. For instance, the LSTM algorithm could be trained using only benign data without using the labels as outputs [55]. Hence, in this work, algorithms that used only benign data (one-class) during the training phase are categorized under unsupervised learning. The following sections review the literature that belongs to each category discussed previously. Comprehensive summary tables for each section are tabulated in Tables 3 through 8.

5.1 ID-Based Detection

Attacks such as DoS and spoofing are changing some properties of message ID sequences. These attacks can be launched by inserting or deleting frames that change the frame frequency compared to normal situations. Even if the attack (masquerade attack) does not change the frequency of IDs, the context of the IDs might be changed due to the time synchronization mismatch with a legitimate ECU [27]. These properties can be utilized to detect attacks on the CAN bus. In reviewed literature, the authors used IDs as a feature of AI-based algorithms to develop IDSs. Timestamp or time differences between consecutive IDs were used to calculate feature values related to IDs. This section discusses such IDSs.

5.1.1 Unsupervised Learning. Seo et al. [140] proposed a novel IDS for IVNs based on GAN. The **GAN-Based Intrusion Detection System (GIDS)** can learn to detect unknown attacks using only benign data. Two models—a generative model to capture the data distribution and a discriminate model to estimate the probability that a sample comes from the training data—were used in GIDS. Two discriminators were combined to detect both known and unknown attacks. Hyundai's YF Sonata was used as a testing vehicle to generate the **Hacking and Countermeasure Research Labs Car Hacking (HCRL CH)** dataset [51] and launched DoS, fuzzy, RPM, and gear spoofing attacks. The first discriminator achieved 100% average accuracy, whereas the second discriminator achieved 98% average accuracy. As per the authors, GIDS is difficult to manipulate by an attacker due to the pre-trained DL method. Further, it can detect intrusions in real time. The same data pre-processing technique used in the work of Seo et al. [140] was used for the GAN-based IDS proposed by Chen et al. [25]. They replaced the GAN's true false classifier with additional double

Table 4. Summary of Payload-Based Attack Detection in a CAN Bus Using Unsupervised Learning

Reference	Model	Algorithm	Dataset	Attack	Strengths	Weaknesses
[15] 2021	Deep learning	LSTM	Public real data (HCRL CH)	Flood, replay, drop, spoof, fuzzy	Near real-time detection	Only considered continuous signal values, ignore ID correlations
[151] 2021	Deep learning	CNN-LSTM	Public real data (HCRL SET)	RPM and gear spoofing	Used only benign data for training	Low detection rate for gear attacks
[158] 2022	Deep learning	CNN	Simulation data (SynCAN)	Plateau, continuous change, playback, suppress	Used only benign data for training, time and memory efficient	Ignored the signal dependencies
[171] 2022	Deep learning	Autoencoder	Public real data (HCRL OTIDS)	Payload value change	Near real-time detection	Only tested for simple dataset and attack

The dataset description can be found in Section 5.5.1. The complete table is available in the supplementary material associated with this article.

classifiers. This model outperformed the model proposed by Seo et al. [140] for all attacks. The HCRL CH dataset was used to evaluate the GAN and convolutional adversarial autoencoder-based model [58]. This model was trained with unlabeled data to learn the normal patterns. Experimental results showed that the proposed model outperforms baseline models for both detection rate and latency. However, this work was only limited to message injection attack detection. Kavousi-Fard et al. [75] also used a GAN-based model for IVN anomaly detection. Frequency for each CAN ID was used as the feature, which limits the detection of attacks such as masquerade attacks.

Avatefipour et al. [13] proposed an **Anomaly Detection Model (ADM)** based on a modified **One-Class Support Vector Machine (OCSVM)**. The authors used a modified bat algorithm as the parameter optimization algorithm. For the model evaluation, CAN bus data from an unmodified vehicle and two other public CAN bus datasets [88] were used. **Isolation Forest (IF)** and classical OCSVM were selected to benchmark the proposed model. The proposed model outperformed the baseline models for DoS attacks. Furthermore, the computational time requirement was sufficient to deploy the proposed solution in a real-world environment. Similarly, the OCSVM-based ADM proposed by Al-Saud et al. [3] used ID frequencies as features. They used the social spider optimization algorithm to find the best support vector regression parameters. A real vehicle dataset with a DoS attack was used to evaluate the proposed model. Despite the promising results of both models, the lack of testing against various real-life attack scenarios can be identified as a common limitation. IDS, which used an adapted streaming data IF algorithm [143], showed that CAN traffic demonstrates insignificant concept drift. Therefore, model retraining based on a sliding window did not improve the model performance.

Kalutarage et al. [71] developed a context-aware anomaly detector for monitoring cyber attacks on CAN bus using sequence modeling. N-gram distributions were used to build the sequence model. The authors have estimated maximum likelihood estimators (MLEs) for each N-gram and developed an algorithm to calculate the anomaly certainty ratio using pre-build N-gram models, a predefined threshold, and observation windows. The anomaly certainty score and predefined threshold were used to classify the message as anomaly or benign. Experimental results that utilized the HCRL CH dataset showed that the proposed model could identify RPM and gear spoofing attacks with higher accuracy. This was tested against only two types of spoofing attacks, and the computational efficiency of the algorithm was not discussed. An ensemble model based on GRU and time-based models was proposed by Rajapaksha et al. [134] to overcome the computational inefficiency of N-gram-based models and to detect a wide variety of attacks. The GRU-based model predicts the next CAN ID, whereas the time-based model monitors ID inter-arrival times. Three public datasets and 16 attacks were used to evaluate the proposed model and achieved a greater than 99% F1-score for 13 attacks. This work showed ensemble models' effectiveness in detecting a wide variety of CAN bus attacks. Shi et al. [144] proposed a temporal convolutional network based IDS

using word embedding of CAN IDs. Experimental results on the HCRL CH dataset showed good detection for fuzzy and DoS attacks. Treating CAN ID sequences as word sentences, Nam et al. [118] introduced a **Generative Pretrained Transformer (GPT)** model to learn the pattern of a normal CAN ID sequence. Deviations from normal patterns were identified as attacks. They combined two GPT networks in a bi-directional manner. This outperformed the single unidirectional GPT model. The proposed model was designed only to detect injection attacks. A bag-of-words approach was used in the work of Baldini [16] to detect intrusions in IVNs. They generated frequency counts of the presence of words for each sliding window and used them as a feature for ML models. Marchetti and Stabili [107] proposed an anomaly detection algorithm considering the recurring pattern of CAN IDs. They created a transition matrix with all possible transitions between consecutive CAN IDs. This is equivalent to 2-grams in the N-gram based model used in the work of Kalatarage et al. [71]. Instead of probabilities, the authors used true and false status to generate the transition matrix. In the attack detection stage, a validated transition matrix was used to check the availability of consecutive ID sequences and identified new IDs as normal or anomalies. A real dataset with replay, bad injection, and mixed injection attacks was used for model evaluation. Experimental results showed a low detection rate of around 20% to 40% for replay attack. Compared to Kalatarage et al. [71], this might produce a high false-positive rate, as this approach assigned labels for each message, whereas Kalatarage et al. [71] classified messages based on a window.

Desta et al. [32] implemented an IDS using an LSTM model. Two approaches were used for the performance evaluation. The first approach compared the predicted ID with the actual ID. This only achieved 60% accuracy. The second approach used log loss and a predefined threshold to identify anomalies and achieved reasonable accuracy. A real car dataset was used with attacks such as insertion, drop, and illegal IDs. In another work [33], Desta et al. improved the previous IDS [32] by training separate LSTM models for each ID and combining them to create a single anomaly signal. This achieved 100% detection for all attacks. Song and Kim [147] proposed a self-supervised method for in-vehicle anomaly detection using noised pseudo-normal data. The proposed model included two models: a generator and a predictor. The generator used an LSTM model similar to Desta et al. [32] to predict the next CAN ID and the predictor used the Reduced Inception-ResNet model proposed in the work of Song et al. [149] to detect anomalies. They used the noised pseudo-normal data generated by the generator model to train the ADM. The HCRL CH dataset was used for the performance comparison, and the proposed model outperformed the other algorithms such as SVM, OCSVM, and CNN. Sharmin and Mansor [142] proposed IDS based on IF to detect message injection attacks using CAN ID timing as features. The HCRL CH dataset was used to evaluate the algorithm with gear and RPM spoofing attacks. This was trained using a one-class approach. Linear time complexity and low resource requirement were the advantages of the proposed solution.

Kuwahara et al. [86] used two types of features including total counting and ID counting in a CAN sequence window to detect malicious messages. Both supervised and unsupervised methods were used as the classification approach. Principal Component Analysis and k-d tree were used to optimize the nearest neighbor discovery. Experimental results that used a real vehicle dataset with simulated attacks showed that the supervised method outperformed the unsupervised method. However, supervised methods require attack data during the training and fail to identify unknown attacks. Han et al. [54] proposed an anomaly detection and attack identification method for IVNs. They calculated statistical features for event-triggered intervals for each CAN ID. Calculated feature values were used to train ML models such as DT, RF, and XGBoost to identify attacks. Experimental results that used two real datasets with realistic attacks showed high anomaly and attack detection capability.

5.1.2 Supervised Learning. The sequential behavior of CAN data can be used to detect anomalous behavior. Using this property, Song et al. [149] proposed a **Deep Convolutional Neural Network (DCNN)**-based IDS to protect the CAN bus from cyber attacks. During the injection attack period, a sequential pattern of the ID changes due to frequent frame injection. The authors capitalized on this change to detect message injection attacks. Inception-ResNet was used as the DCNN model with $29 \times 29 \times 1$ input and binary output. The HCRL CH dataset was used to evaluate the proposed solution. The DCNN model outperformed the baseline models for all attack types. Further, the proposed model required 5 ms to process one sample, which included 29 CAN messages under GPU acceleration. Desta et al. [34] also used a CNN-based IDS trained on recurrence plots. Deployment on NVIDIA's Jetson TX2 showed that higher detection latency of 117 ms. A lightweight multi-attack quantized ML model deployed using Xilinx's DL processing unit IP on a Zynq Ultrascale+ (XCZU3EG) FPGA was proposed in the work of Shankar [141]. This system used CNN as the detection algorithm and outperformed the baseline models. A blockchain-based federated forest Software-Defined Networking (SDN)-enabled IDS was proposed by Aliyu et al. [8]. This system created a RF model to detect attacks. Fourier transformation was used to create features from CAN IDs. The HCRL OTIDS [48] dataset was used for the performance evaluation. The usage of blockchain reduces the risk of adversary poisoning, which potentially improves the security of the AI model. This model helps vehicle owners and manufacturers keep the underlying data confidential.

Jedh et al. [67] used an LSTM model to detect malicious message injections in the CAN bus. They used message sequence graphs of CAN IDs at successive time windows to calculate Pearson and cosine similarities, which were then used as the features for the LSTM model. A real vehicle dataset with fabricated RPM and speed messages was used to evaluate the model performance. Experimental results showed that the detection capability of the algorithm depends on the selected window size. Refat et al. [136] also used a similar graph-based model as a CAN IDS. They extracted seven graph properties as features and used them to train SVM and KNN ML models. The proposed model achieved a greater than 95% F1-score for DoS, fuzzy, and spoofing attacks in the HCRL CH dataset.

One of the major drawbacks of ID-based IDSs is their limited ability to detect attacks that manipulate the message payload without changing ID sequences or frequencies. However, even for an attack that manipulates only the payload, CAN ID sequences might change due to event-triggered messages in the CAN bus [122].

5.2 Payload-Based Detection

Attacks such as replay and spoofing not only change the CAN IDs but might also change the CAN payload as well. This depends on the characteristics of the particular attack. Generally, there are two ways to change the payload: either replay (previous payloads) or modify the payload values. These changes will cause to change in the pattern of payload sequences. This section discusses the IDSs that utilize this property to detect attacks.

5.2.1 Unsupervised Learning. Chockalingam et al. [30] tested LSTM and OCSVM to detect anomalies in CAN frames. The authors used a real dataset and created fuzzing and misplaced non-anomalous packets. Experimental results have shown that the OCSVM model produced a 7% false-positive rate using the linear kernel. The non-linear kernel took much time to optimize. The LSTM model outperformed OCSVM. However, this was tested with two attack types, and complete evaluation results for individual attack types are unavailable. Tomlinson et al. [161] used a one-class compound classifier to detect attacks in the IVN. Payload values of three separate CAN IDs were considered for the analysis. The authors used fuzzing attacks to test the classifier. However,

evaluation results were not promising and produced many false positives. They proposed ensemble detection methods for CAN IDs to overcome problems that arise with one classifier. Tomlinson et al. [162] used OCSVM, the compound classifier, and the **Local Outlier Factor (LOF)** to identify attacks in CAN data values. Experiment results showed that OCSVM and LOF outperformed the compound classifier. However, the results were not acceptable to use in real-world situations.

Narayanan et al. [120] built an anomaly detection system called *OBD_SecureAlert*. This concept is based on formulating a sequence of CAN bus messages as a time series ML problem. The authors considered the vehicle movement as a sequence of states dependent on its previous state. All observations of a sliding window were used to determine the posterior probability of that sequence. The anomalous status was identified by considering the probability of such sequence and a defined threshold. Evaluation results showed that OBD_SecureAlert works well with single and multiple observations. However, this was tested against limited anomalous states, and identifying specific sensor data in CAN messages is challenging. Levi et al. [90] proposed a HMM-based hybrid anomaly detection algorithm using a new temporal detection technique. The authors used a rule-based engine to monitor different interfaces and generated events using raw data. These events from different interfaces were used to train a HMM model as a normal behavior. Experimental results showed that the proposed model achieves high AUC and F1 measures with low false-positive and false-negative rates. They proposed a hybrid deployment approach that uses a rule-based client to send data to the back-end and run the detection algorithm on the cloud. This cloud-based platform facilitates monitoring car fleets in addition to individual cars. Despite these advantages, the proposed method is based on events and relevant attributes. It is challenging to obtain the complete list of events and attributes.

Taylor et al. [157] introduced an LSTM model for CAN bus anomaly detection. The underlying concept of this approach is that the ML model can be trained to predict the next packet data value, and its deviation from the actual value can be used to identify the anomalies. Nineteen CAN IDs were selected, and they trained different LSTM models for each ID. The authors selected a real dataset and created five types of attacks considering three basic cases: new packets are added, expected packets are missing, and the payload of packets is unusual. Experimental results have shown that different IDs achieve various ROC curve values ranging from 0.17 to 1 for five attack types. For the practical use of the proposed model, the threshold needs to be selected with more experiments. Further, since they considered separate models for IDs, it cannot utilize the inter-dependencies between IDs for anomaly detection. Tanksale [154] proposed an LSTM-based IDS using CAN measurements such as longitudinal acceleration, RPM, and brake position. This model can predict the future values of selected CAN signals based on previous signal values. The deviation between the predicted and actual values was used to identify the anomalous signals. A dataset collected from 10 cars was used to train the algorithm, whereas a subset of the dataset with injected anomalous frames was used for the performance evaluation. The proposed model showed greater than 98% accuracy with a 1% to 2% false-positive rate. A similar LSTM-based model with an embedding layer was proposed in the work of Balaji and Ghaderi [15] for CAN payload values. The usage of payload values of other IDs as the context ensured the capturing of inter-ID correlation. Experimental results on gear attacks of the HCRL CH dataset showed a relatively lower detection rate. Hanselmann et al. [55] introduced CANet, an LSTM-based IDS to identify attacks in the CAN bus. They used separate LSTM models for each CAN ID and concatenated the outputs to a single latent vector. This was trained in an unsupervised manner, and the difference between initial and reconstructed signal values was used to define the normal status. The authors used a real vehicle dataset with 13 IDs and a synthetic dataset (SynCAN) with 10 IDs for the performance evaluation. Six attacks and two baseline models were selected to evaluate the model performance. Experimental results showed that the proposed model achieved a higher detection rate and low

false-positive/negative rates for all attacks in both datasets. It outperformed the baseline models for almost all attack types. However, their anomaly score is only feasible with a limited number of signals and hence not suitable for real-world applications.

Novikova et al. [124] developed an autoencoder to detect anomalous payload in the CAN bus. A large dataset from nine different vehicles was used to train the algorithm. They identified 81 signals common to all vehicles and grouped them into 32 subgroups of three signals based on the signal relationships. Experimental results on modified payload values showed a higher detection rate. The SynCAN dataset was used to evaluate the reproducibility of the proposed model. Detection rates of plateau, continuous change, and playback attacks were 99%, 94%, and 95%, respectively. Subgroups require a separate autoencoder for each group, and deploying a large number of DL models under a resource-constrained IVN is a challenging task. Kukkala et al. [83] proposed a GRU-based recurrent autoencoder to detect anomalies in the CAN bus. The SynCAN dataset was selected as the evaluation dataset, and separate autoencoder models were trained for each ID. Signal-level intrusion scores between predicted and true signal values were used to identify the anomalous signal values. The authors used accuracy as the evaluation metric without considering the highly imbalanced status of the dataset. Longari et al. [103] also proposed a similar LSTM autoencoder model. A real-world dataset from an Alfa Romeo Giulia Veloce with injected anomalous frames was used for model training and evaluation. Kukkala et al. [84] improved the GRU-based recurrent autoencoder [83] by replacing the GRU layer with an LSTM layer and introducing the self-attention mechanism. Instead of identifying the threshold value as the intrusion score, the proposed model used OCSVM as the attack detector. Thiruloga et al. [158] introduced a novel anomaly detection framework using a temporal CNN. The proposed model used a DT-based classifier as the attack detector. This model achieved an improvement of 32.7% in false-negative rate compared to the best-performing baseline model. All of these models [83, 84, 103, 158] processed ID-wise data independently and trained separate models for each ID. This limits the capability of detecting signal correlations to detect anomalies such as collective anomalies. The deep contractive autoencoder-based ADM proposed by Lokman et al. [102] achieved 91% to 100% detection rates for three attacks. Gherbi et al. [45] proposed a multivariate time series representation to represent CAN payload data and used autoencoder-based DL models such as the fully connected network, CNN, LSTM, and temporal convolution network to detect intrusions in the CAN bus. The proposed models achieved a higher F1-score for all attacks in the SynCAN dataset. However, the proposed feature matrix might be inefficient for a real vehicle due to the many ECUs.

Narasimhan et al. [119] trained an autoencoder-based model and used a Gaussian mixture model to identify intrusions in the CAN bus. All other autoencoder-based models discussed earlier used the reconstructed signal for the anomaly detection, whereas this model used the latent space as the input to the Gaussian mixture model. A real dataset of Mercedes ML350 with DoS and fuzzy attacks was used for the performance evaluation. However, this dataset included only four CAN IDs; therefore, results obtained through this model might hinder the practical performance of the model. A multi-layer denoising autoencoder model was used by Wei et al. [171]. He et al. [57] proposed the Hybrid Similar Neighborhood Robust Factorization Machine Model (HSNRFM). They used data fields of similar neighbors to enhance the feature representation. The factorization machine model used the second-order interaction features to predict the final probability of anomalous outcomes. Both of these models [57, 171] used only two CAN IDs to train and evaluate the proposed models. Tanaka et al. [153] employed a density ratio estimation method using a **Neural Network (NN)** model. This approach is based on the change detection method to detect packet frequency changes. However, this model also used only three CAN IDs for the model evaluation. CNN-LSTM with attention mechanism based IDS was proposed by Sun et al. [151]. This model used one-dimensional convolution to extract the abstract features, whereas bi-directional

LSTM was used to extract time dependence. They only considered the continuous physical values extracted from the 64-bit payload. Bit flip rate was used to identify continuous fields. The CAN Signal Extraction and Translation Dataset (HCRL-SET) provided by HCRL was used for the model evaluation with simulated payload attacks. The proposed model outperformed the baseline models. They evaluated the model detection time under attack in a real vehicle. This showed that the attacks could be detected within 5.7 ms in a real vehicle. However, the proposed model has a few limitations, including selecting a subset of signal values and ignoring payload correlation between different IDs.

5.2.2 Supervised Learning. Kang and Kang [73] proposed a DNN-based IDS for IVNs. CAN payload was selected to generate the features, whereas mode and value information were used as the dimensionality reduction technique. Initial weights for the DNN model were obtained using a separate Deep Belief Network. Finally, the authors used a template-matching technique to compare the training sample and a new CAN packet to identify the attack scenarios. The authors used a simulation dataset with packet injection attacks for the evaluation. Experimental results showed that the DNN model outperformed baseline models. Zhou et al. [185] proposed a DNN and a triplet loss network for CAN bus anomaly detection. The proposed model used the distance between anchor samples and the positive and negative examples to identify anomalies. Experimental results showed the real-time detection capability of the algorithm. However, both Kang and Kang [73] and Zhou et al. [185] relied on mode and value information of CAN data, and identifying these information is quite challenging without having the DBC file. Zhang et al. [181] proposed DNN-based IDS for the CAN bus. They have used Gradient Descent with Momentum and Gradient Descent with Momentum and Adaptive Gain to improve efficiency and accuracy. Performance evaluation was done using a real dataset collected from a car. Experimental results revealed that the proposed model could detect replay attacks with a high detection rate. Further, it was noticed that the Gradient Descent with Momentum and Adaptive Gain algorithm achieved faster convergence compared to the ADM algorithm. The authors had access to the sensor values and used those as separate features. However, these values cannot be distinguished without having the DBC file or knowledge about the CAN payload.

Fenzl et al. [40] introduced a continuous field classification algorithm to identify the payload value alignments. Then, a DL-based approach was used to identify the anomalous fields. Datasets from Renault Zoe electric car and manipulated signals were used to evaluate the model performance. Interestingly, their continuous field classification approach showed slightly better detection capability than the field classification obtained by the DBC file. However, these attacks are not realistic, as they were created during the post-processing. This approach does not reflect the inter-dependencies among variables. Martinelli et al. [109] used four ***k*-Nearest Neighbor (KNN)** classifiers to identify four types of attacks that target the CAN bus. These algorithms include two types of fuzzy-roughKNNs the discernibility classifier and a fuzzy unordered rule induction algorithm. Fenzl et al. [41] used DTs modeled through genetic programming to detect intrusions in the CAN bus. Features and feature boundaries selection were based on the CAN DBC files. Three datasets, including the HCRL CH dataset, were used to evaluate the performance of the algorithm. Experimental results showed that the proposed algorithm achieved similar detection capability as ANN algorithms with much-improved detection time.

Wang et al. [168] proposed an LSTM model with optimized parameters. They implemented an LSTM model to evaluate the vulnerability of an ADM with a black-box attack. A dataset of the velocity of a vehicle was collected from the CAN bus for evaluation purposes. The threshold to identify anomalies can be defined considering the maximum MSE. Evaluation results for the ADM under attack scenarios are not available, as the authors only focused on the evaluation results of

Table 5. Summary of Payload-Based Attack Detection in a CAN Bus Using Supervised Learning

Reference	Model	Algorithm	Dataset	Attack	Strengths	Weaknesses
[181] 2019	Deep learning	DNN	Simulation data	Packet injection	Real-time detection	Limited generalization capability without a CAN DBC file
[40] 2020	Deep learning	DNN	Collected real data (Renault Zoe electric car data)	Payload value manipulation	High detection rate, explainability of the results	Only tested for simple simulated attacks
[168] 2020	Deep learning	LSTM	Collected real data (Toyota hybrid car data of 120-second drive)	DoS, fuzzing, spoofing	Both point and contextual anomalies detection	Only tested for limited attacks
[41] 2020	Traditional ML	DT, genetic programming	Public real data (HCRL CH, Tesla Model X data, Renault Zoe electric car data)	RPM and gear spoofing	Near real-time detection, memory efficient	Limited generalization capability without a CAN DBC file

A dataset description can be found in Section 5.5.1. The complete table is available in the supplementary material associated with this article.

the victim model for attacks. Similarly, Li et al. [92] used an LSTM model to test an adversarial attack defending system. To this end, they used the LSTM model proposed by Khan et al. [78]. This model achieved a 98% accuracy for the used simple attack scenario.

5.3 CAN Frame-Based Detection

In addition to using only ID or payload as the feature, IDSs in the literature used a combination of features to capture the pattern changes in CAN data sequences. This has the advantage of detecting both ID changes and payload manipulation attacks. Other features combined with ID and payload are DLC and time (time gap).

5.3.1 Unsupervised Learning. Berger et al. [20] tested NN, LSTM, SVM, and OCSVM algorithms for IVN attack detection. Experimental results that used the HCRL CH dataset showed that NN outperformed other models. A mobile edge-assisted LSTM-based anomaly detection approach was proposed by Zhu et al. [187] to overcome the computational limitations of IVNs. A real-time performance of 0.61 ms was observed in the proposed model with around 90% accuracy. Gao et al. [43] introduced a new in-vehicle IDS based on DL and SOEKS (set of experience knowledge structures). Experimental results that used a real vehicle dataset showed that usage of SOEKS and information entropy improved attack detection. Barletta et al. [17] proposed an unsupervised Kohonen SOM (self-organizing map)-based anomaly detector for the CAN bus. They integrated Kohonen SOM with a *k*-means clustering algorithm using a distance-based approach. This model was tested with DoS, fuzzy, gear, and RPM spoofing attacks. They compared this with the traditional approach where the *k*-means algorithm processed a neuron's codebook vectors. Experimental results have shown that the proposed technique outperforms the traditional approach for all attack datasets. Leslie [89] proposed an ensemble hierarchical agglomerative clustering-based model to detect malicious traffic in heavy-duty ground vehicles. The author used a dataset related to the SAE J1939 protocol, which is based on the CAN bus. This was evaluated using spoofed engine speed messages and showed a higher detection rate.

Lin et al. [96] proposed a deep denoising autoencoder-based model to detect injection attacks on IVNs. They used an evolutionary-based optimization algorithm to overcome premature convergence and find the optimum network structure. Experimental results that used the HCRL OTIDS and two real datasets showed that the proposed model outperformed selected baseline models. Nakamura et al. [117] proposed a hybrid model of a LightGBM-based supervised model and an autoencoder-based unsupervised model. Time differences of consecutive CAN IDs, CAN

Table 6. Summary of CAN Frame-Based Attack Detection in a CAN Bus Using Unsupervised Learning

Reference	Model	Algorithm	Dataset	Attack	Strengths	Weaknesses
[96] 2020	Deep learning	Autoencoder	Collected real data and public real data (HCRL OTIDS)	Flooding, fuzzy, and malfunction	Found the optimum network learning structure for a higher detection rate	Risk of finding a complex model structure
[133] 2021	Deep learning	LSTM	Collected real data	Random CAN payload values	Used only benign data for training	Limited generalization capability to other vehicles
[117] 2021	Hybrid	LightGBM and Autoencoder	Public real data (HCRL SA)	Flooding, fuzzy, and malfunction	Used only benign data for autoencoder model training	Limited performance evaluation
[76] 2021	Deep learning	LSTM	Public real data (HCRL CH)	DoS, fuzzy, RPM, and gear spoofing	Near real-time detection	Demanded a large number of observations to obtain high detection accuracy

A dataset description can be found in Section 5.5.1. The complete table is available in the supplementary material associated with this article.

ID, and payload values were used as the features. Experimental results that used the **HCRL Survival Analysis (HCRL SA)** dataset showed that the hybrid model outperformed the pre-trained LightGBM model. However, a comparison between the pre-trained and autoencoder models is not available to make a fair comparison of the hybrid model performance. Qin et al. [133] proposed an LSTM-based anomaly detection algorithm to detect the abnormal behavior of the CAN bus. Experimental results have shown that the proposed model can detect anomalous data with greater than 90% accuracy. Further, the authors tested this with two more vehicles, and the performance was not good enough to generalize the model to other vehicles. An LSTM model with an improved feature processing technique was used in the work of Khan et al. [76] for IVN malicious activity detection. The HCRL CH dataset based experimental evaluation outperformed the baseline models for both detection rate and latency. The LSTM autoencoder-based model proposed by Ashraf et al. [12] also used the HCRL CH dataset. Packet count and bandwidth of the outbound traffic of a fixed window were used as the features. These features are only suitable for detecting injection attacks. Zhou et al. [186] proposed an autoencoder model with dedicated models for each CAN ID. An improved IF method with data mass was used to detect tempering attacks in the work of Duan et al. [36]. This was evaluated using a simulation environment and outperformed the OCSVM and LOF algorithms.

5.3.2 Supervised Learning. Tian et al. [159] proposed an IDS based on the Gradient Boosting Decision Tree for the CAN bus. Nine features were used for the classification, including the payload of CAN message and entropy-based feature. They changed the payload values of a real dataset to create abnormal messages. Experimental results showed that the true-positive rate was 97.67% and the false-positive rate was 1.2%. However, this was tested with a very basic attack scenario of CAN payload values changing, and real-world attack detection will be much more complex. Wasicek et al. [169] implemented a CAID (context-aware IDS) framework using ANN to identify manipulations in IVNs. CAID is equipped with three modules: the monitor module reads and aggregates information, the detectors module identifies anomalies, and the reporter module connects with the user. Features used for ANN model include vehicle speed, engine RPM, fuel rate, and calculated load. This model was evaluated using a real vehicle for chip tuning and power boxing manipulations. Experimental results have shown that it could accurately recognize the manipulated attacks. However, this experiment was done in a constrained environment, whereas the real-world environment might be quite different. The ANN-based lightweight model proposed by Basavaraj and Tayeb [19]. This model marginally outperformed the baseline models. Alshammari et al. [9] proposed KNN and SVM algorithms to cluster and classify DoS and fuzzy attacks in the CAN bus. As

per the experimental results, KNN outperformed the SVM algorithm for both attacks of the HCRL CH dataset. However, the DoS detection rate was comparatively low compared to the fuzzy attack.

Zhang et al. [184] proposed an IDS for the CAN bus considering the balance between the efficiency of the rule-based approach and the high detection rate of the DNN-based approach. The first stage, which is the rule-based approach, enables efficient anomaly detection. CAN frames, which pass the rule-based detection model, send to the DNN-based detection model to further identify undetected anomalies. Evaluation against five types of attacks using three real datasets showed high detection rates and low false-positive rates for all datasets. However, evaluation results with regard to five attack types are not included in this work. Similarly, Zhang and Ma [183] introduced a hybrid approach for in-vehicle intrusion detection. Datasets related to four real vehicles were used for the performance evaluation. This approach was only applicable to periodic messages. Weber et al. [170] proposed a hybrid IDS that is capable of identifying both point and contextual anomalies. The authors used eight classes of sensor data defined by Müter et al. [116]. They used LODA (a lightweight online detector of anomalies) [131] as the classification algorithm. Synthetic CAN data with an altered sequence was used to evaluate the proposed model. Despite the promising results, this was tested with limited simplified anomaly scenarios. Rule-based and RF-based hybrid IDS was proposed by Kang et al. [72]. Time interval, data field differences, and ID lag values were used as the features. The RF model showed a poor detection capability than the rule-based approach.

Kalkan and Sahingoz [70] used six different ML models—RF, bagging, ADA boosting, NB, Logistic Regression, and NN—to compare their attack detection capability of a large CAN dataset. The authors could achieve a promising detection rate using simple ML algorithms with default parameters. However, they did not discuss the dataset creation or features used to train the algorithms. Similarly, Alfardus and Rawat [6] also used ML algorithms such as KNN, RF, SVM, and **Multilayer Perceptron (MLP)** to detect CAN bus attacks. Moulahi et al. [114] used RF, DT, SVM, and MLP to compare the detection capability. Features related to time, ID, DLC, and payload values were used. Performance evaluation using the HCRL OTIDS dataset showed very low detection capability for fuzzy attacks. Amato et al. [10] used NN and MLP-based models to detect attacks on the HCRL CH dataset. Dong et al. [35] did a comparative study on supervised versus semi-supervised ML for IVNs anomaly detection. Minawi et al. [113] used Random Tree, RF, Stochastic Gradient Descent with hinge loss, and NB to detect gear and RPM spoofing, DoS, and fuzzy attacks in the HCRL CH dataset. CAN ID and payload values were used as the features. Except for the fuzzy attack, all attacks were detected with a 100% F1-score. Anjum et al. [11] also used the HCRL CH dataset to evaluate the XGBoost-based CAN IDS. Park and Choi [129] used multi-labeled hierarchical classification as the intrusion detection model. Experimental results that used the HCRL SA dataset showed that the proposed model outperformed the selected baseline models. The same dataset was used in the NN-based IDS proposed by Francia and El-Sheikh [42]. The main objective of the proposed approach was to identify vehicle models and anomalies. All of these works [6, 10, 11, 35, 70, 113, 114, 129] can be considered as basic ML and DL model comparisons for CAN attacks. None of these models has the capability to detect unknown attacks. The XGBoost classifier outperformed the VGG16 model for gear and RPM spoofing attacks in the work of Lin et al. [94]. Aksu and Aydin [1] proposed a meta-heuristic algorithm called the *modified genetic algorithm* for the CAN feature selection. This can be considered as a dimensionality reduction approach. They used ML models such as SVM and DT to evaluate the effectiveness of the feature selection.

Suda et al. [150] proposed LSTM-based IDS, which utilized the time series features of the CAN frame. These features include frame interval (derived from the time), ID, and payload values. Data was collected from a real vehicle and used modified ID, data field, and flooding as attacks to evaluate the system. Khan et al. [77] proposed an LSTM-based attack detection model for IVNs. They used

two attack-free CAN bus datasets—HCRL CH and the AEGIS repository [68]—to create replay and amplitude-shift attacks. Experiment results for replay and amplitude-shift attacks showed that the LSTM model achieved the best accuracy for both datasets. Even though LSTM recorded the best results comparatively, these figures are not promising, as accuracy and precision values were around 80% to 90% and recall values were around 30% to 40%. Further, the DBC file and processed data with features are hard to find. Xiao et al. [177] introduced a novel RNN-based IDS by optimizing LSTM and GRU architectures and using a simplified attention model to make the model lightweight. The RF algorithm was used as the classification algorithm using the features generated by the RNN model. In contrast, CAN ID, DLC, and payload fields were used as the input features for the RNN model. They validated their approach using the HCRL OTIDS dataset and compared the performance with eight variants of the proposed model. However, the RF algorithm learns only to detect attacks in the training dataset and may fail to detect new attacks. Ma et al. [106] proposed a GRU-based lightweight IDS for CAN bus intrusion detection. They also used a low-complexity feature extraction algorithm to extract features from CAN frames. The proposed model showed near real-time performance and a higher detection rate than the baseline models. However, the usage of the supervised learning approach limits novel attack detection. An attention-based technique was used in the work of NasirEldin et al. [121]. An attention layer was used to capture the most important part of the data, whereas a self-attention layer was used to identify the relationship between each data element. They used positional encoding to capture the positional information. Performance evaluations that used HCRL CH data showed that the proposed model marginally outperformed baseline models, including an LSTM model.

Hossain et al. [62] proposed an IDS for the CAN bus based on LSTM. The authors used both binary and multi-class classification to evaluate the IDS with vanilla LSTM and stacked LSTM models. Experimental results that used the HCRL SA dataset showed that the proposed vanilla LSTM model outperformed the compared survival analysis method. Since both CAN ID and payload have been considered in the model, it can detect both point and contextual anomalies. They used the same model in another work [61]. Hossain et al. [60] used a CNN model instead of the LSTM model proposed in their other work [62]. They collected datasets from three cars and injected anomalous frames to create attacks. The proposed model achieved a high attack detection rate for all attacks. Due to the supervised learning approach used, both of these models [60, 62] cannot detect unknown attacks. The CAN bus attack detection framework introduced by Tariq et al. [155] utilized both rule-based and DL (LSTM) models. DoS, fuzzing, and replay attack were used to evaluate the proposed model. The ensemble model achieved better accuracy than the individual rule-based or LSTM model for all attacks. Detection time analysis showed that the average detection time delay was 0.02 seconds. This was evaluated against three simple attacks that changed the ID frequency significantly. They also introduced CANTransfer, a transfer learning based IDS for CAN bus [156] using the same data, features, and attacks. The authors trained a convolutional LSTM model (ConvLSTM) as a binary classification problem. One-shot transfer learning was used to retrain the model to detect new attacks. DoS attack was used during the training phase, and fuzzing and replay attacks were used with one-shot transfer learning. They could achieve 26.60% performance gain compared to the best baseline model. A deep transfer learning based P-LeNet method used in the work of Mehedi et al. [110] outperformed the baseline models. Transfer learning will help reduce the need for collecting a large amount of data to detect each new type of attack. LSTM-based simple IDS proposed by Kishore et al. [80] outperformed the traditional ML models such as RF and XGBoost.

Rehman et al. [137] proposed CANintelliIDS, a novel approach to detect intrusions in the CAN bus based on CNN and attention-based GRU models. Unlike other approaches that predicted binary classes, this model predicted the attack type. The authors evaluated this algorithm with

Table 7. Summary of CAN Frame-Based Attack Detection in a CAN Bus Using Supervised Learning

Reference	Model	Algorithm	Dataset	Attack	Strengths	Weaknesses
[114] 2021	Traditional ML	DT, RF, SVM, MLP	HCRL OTIDS	DoS, fuzzy, impersonation	High detection rate for impersonation attack	Poor detection for fuzzy attack
[114] 2021	Deep learning	DBL	Public real data (HCRL CH)	DoS, RPM and gear spoofing, fuzzy	Provided more information about predictions	Higher epistemic uncertainty
[99] 2022	Deep learning	CNN and LSTM	Public real data (HCRL CH)	DoS, RPM and gear spoofing, fuzzy	High detection rate	Computational expensive model architecture
[106] 2022	Deep learning	GRU	Public real data (HCRL CH)	DoS, spoofing, fuzzy	Lightweight model	Limited types of attack detection

A dataset description can be found in Section 5.5.1. The complete table is available in the supplementary material associated with this article.

a single attack data sequence and mixed attack data sequence separately. Binary output was compared with recent state-of-the-art baseline models (e.g., [149, 156]). It outperformed all models with a maximum 5.32 F1-score gain. This work proved that DL-based ensemble models could be successfully used to detect different attacks on vehicle networks. However, the computational efficiency of the proposed approach has not been discussed. Lo et al. [100] used a hybrid model of CNN and LSTM networks for in-vehicle attack detection. CNN was used to extract spatial features, whereas LSTM was used to extract temporal features from CAN data frames. Experimental results that used the HCRL CH dataset showed an approximately 100% detection rate. They used the same model in the work of Aldhyani and Alkahtani [4].

Ale et al. [5] used a **Deep Bayesian Learning (DBL)** model to detect and analyze car hacking behaviors. Experimental results that used the HCRL CH dataset showed a slightly lower accuracy than a deterministic DL model. However, the DBL model is capable of providing more information about its prediction, which can help further analysis of abnormal behaviors. Islam et al. [65] developed a hybrid quantum-classical NN to detect an amplitude shift cyber attack on the CAN bus. The usage of the DBC file for feature creation reduces the generalization capability of the proposed model. DNN and incremental learning based IDS was introduced by Lin et al. [95] to address the driving environment and behavior changes. Predicted class labels of the DNN model were used as the labels for online model updates. This approach has a risk of reducing the model performance when the predictions of the original model are incorrect. Rumez et al. [138] employed a similar approach like Kalutarage et al. [71] to develop a hybrid anomaly detection framework for diagnostics communication. In addition to the sequence-based model that uses the n-gram distribution for CAN IDs, the authors used the byte-based model to utilize the CAN messages payload for attack detection. Real and synthetic datasets with three attack types were used for the model evaluation. Their detection framework is only limited to automotive diagnostic communication.

5.4 Physical Characteristics-Based Detection

All of the IDSs discussed previously used the data in the CAN data frame. Loukas et al. [105] proposed a cloud-based cyber-physical IDS for vehicles. To this end, they used both cyber and physical features. Both deep MLP and RNN architecture (LSTM) were used as the algorithms. However, this was tested only on a robotic vehicle. Motivated by the works of Cho and Shin [29] and Choi et al. [31], Xun et al. [178] proposed VehicleEIDS, a novel IDS based on the vehicle voltage signal. This model utilized the unique voltage signals generated by ECUs. The authors extracted differential signals using 14 time-domain features from two vehicles. Finally, a deep support vector domain description (deep SVDD) model was used to develop the VehicleEIDS. This model can distinguish the voltage signal of ECUs with greater than 97% of accuracy. Among the discussed IDSs, this is the only IDS that can be used to identify the attack source. Another advantage of the proposed

model is that deployment can be done in the existing CAN bus without changing the protocol, as this does not require the bandwidth or computing resources of the CAN bus. However, this was tested only against simple attacks such as injection and replay.

5.5 Benchmark Datasets

Data is considered as the core of AI algorithms. The accuracy of AI models highly depends on the availability and quality of the data. This is applicable for AI-based IDSs as well. This section discusses the publicly available datasets that can be used to train and evaluate in-vehicle IDSs, and Table 8 provides the comparison of model evaluation results for benchmark datasets:

- *Car hacking dataset for intrusion detection* (HCRL CH) [51]: This is the most widely used dataset in the literature [9, 20, 71, 109, 140]. It was released by the **Hacking and Countermeasure Research Lab (HCRL)** and publicly available for academic purposes. The dataset was collected from a real vehicle while attacks were being performed. This dataset includes 500 seconds of benign data (collected while driving the car) with four attack types. The attacks are DoS, fuzzing, and two spoofing attacks (RPM and gear). Each of these attack datasets are comprised of 300 intrusions of message injection that lasted for 3 to 5 seconds. Each attack dataset was captured for 30 to 40 minutes. The dataset attributes are timestamp, CAN ID, DLC, payload, and label representing injected messages and normal messages. The dataset captured a fair amount of attack instances. All of these attacks changed the ID frequency significantly. Therefore, frequency-based or sequence-based approaches can easily detect them. Experimental results of the majority of reviewed works proved this by achieving a greater than 99% F1-score for all attacks. Benign data collection was done while driving the vehicle. However, signal decoding [166] showed that the car was not driven while collecting attack data. Therefore, this dataset is unsuitable to evaluate an IDS.
- *CAN dataset for intrusion detection* (HCRL OTIDS) [48]: This dataset is also produced by HCRL along with their remote frame-based CAN IDS [88]. A Kia Soul vehicle was used to collect benign and DoS, fuzzy, and impersonation (masquerade) attack data. This is the only publicly available CAN dataset with remote frames and responses. Dataset attributes are timestamp, CAN ID, DLC, and payload. Unlike the car hacking dataset, labels (ground truth) are not available as an attribute. Instead, attack injection intervals are available in the documentation that seem incorrect [166] and cannot use to label fuzzy and impersonation attacks due to insufficient details such as injected IDs.
- *Survival analysis dataset for automobile IDS* (HCRL SA) [52]: HCRL published this dataset with their frequency-based CAN IDS [53]. This is the only publicly available CAN dataset that contains real attacks on multiple vehicles. Used vehicles are the Hyundai YF Sonata, Kia Soul, and Chevrolet Spark. On each car, they collected benign data and three attack types, including flooding (DoS), fuzzing, and malfunction (spoofing) attacks. Attributes of this dataset are timestamp, CAN ID, DLC, payload, and label representing injected and normal messages. However, these attacks are basic and could be detected with frequency-based or sequence-based IDS due to the change of frequency. Moreover, three benign datasets relevant to each vehicle are not sufficiently large enough to train a good classifier.
- *Car hacking attack and defense challenge* [50]: HCRL collected this data using a Hyundai Avante CN7 for a competition aimed to develop attack and detection techniques for the CAN bus. Benign, flooding (DoS), spoofing, replay, and fuzzing attacks are included with timestamp, ID, DLC, payload, label, and SubClass (attack type) as data attributes. In other HCRL datasets, attack datasets were available in separate files. In contrast, for this dataset, benign and four attacks are available in the same file. There are benign data available in between attacks. However, the benign dataset is likely not sufficient for algorithm training.

- *CAN Signal Extraction and Translation Dataset (HCRL-SET)* [49]: HCRL published this dataset to support CAN analysis research such as signal extraction and translation. The dataset includes 56 CAN traffic logs collected by periodically sending OBD queries while driving in a controlled environment. This consists of 28 unique CAN IDs. This dataset does not have attack data and information related to benign data.
- *SynCAN dataset* [56]: This simulated dataset was published with their CAN IDS CANet [55]. The purpose of this dataset is to train unsupervised CAN IDS. This is the most widely used dataset in the literature to evaluate unsupervised payload-based IDSs [55, 83, 84, 124, 158]. Unlike other datasets discussed earlier, this contains signal values without providing the raw CAN data. Hence, it is suitable to test signal-based IDSs. This dataset consists of training data and six test datasets. Test datasets include one normal dataset and five attack datasets. Five attacks are defined as plateau, continuous, playback, suppress, and flooding. During the suppress attack, the attacker prevented an ECU from sending frames. For the flooding attack, the attacker sent messages of selected ID with a higher frequency. Plateau, continuous, and playback attacks changed the payload of the CAN frames. However, these attacks are simulated attacks and their effect on a real vehicle cannot be verified.
- *TU Eindhoven CAN bus intrusion dataset* [165]: This dataset was published by the department of mathematics and computer science at Eindhoven University of Technology. They used two cars (Opel Astra and Renault Clio) and a CAN bus prototype to collect benign data. Attacks are simulated and consist of diagnostic, fuzzing, replay, suspension, and DoS attacks. However, changing the timestamp of CAN messages at the post-processing stage made this dataset unrealistic to test AI-based CAN IDSs that use time as a feature.
- *CrySyS Lab dataset and CAN log infector* [126]: This is a benign dataset along with a Python script to generate anomalous CAN logs. The dataset was published by the department of networked systems and services at the Budapest University of Technology and Economics. A set of benign data representing driving scenarios such as driving at a constant speed of 30 km per hour, driving at a speed of 40 km per hour and then lane change then stop, and emergency braking from 60 km per hour to 0 are included in this dataset. Even though this is a benign dataset, the authors provided a CAN log infector that can be used to simulate a wide variety of masquerade attacks. However, adding the attacks during post-processing makes this somewhat unrealistic.
- *AEGIS big data project* [68]: This was published as part of the "AEGIS-Advanced big data value chain for public safety and personal security"big data project. This is a benign dataset of 20 hours of driving that has signal data such as wheel speed, steering wheel angle, role, pitch, and accelerometer values per direction. GPS data are also available. This dataset is similar to that of Hanselmann et al. [56], as both datasets provided signal values. However, the unavailability of attack data limits the usage of the dataset for IDS evaluation. This dataset was used to evaluate the work of Khan et al. [77] with simulated attacks.
- *Real ORNL Automotive Dynamometer CAN intrusion dataset* [166]: The **Real ORNL Automotive Dynanometer (ROAD)** dataset is a real dataset with an advanced set of attacks. The authors reviewed the existing CAN datasets and produced this dataset to address their limitations. This dataset consists of 33 attacks equivalent to 30 minutes of driving and 12 benign datasets that cover different driving scenarios (3 hours). One vehicle was used to collect all data. When collecting the attack data, the vehicle was in a dynamometer (under driving conditions). For benign data collection, they used both roads and a dynamometer and performed a variety of normal and unusual benign driving behaviors. This dataset consists of (i) fuzzing attack, which injected random IDs; (ii) targeted ID attacks, which have four variations such as correlated signal (change the wheels' speed), max speedometer (display

Table 8. Comparison of Model Evaluation Results for Benchmark Datasets

Dataset	Attack	Reference	Algorithm	Experiment Platform	F1-Score	Accuracy	Precision	Recall	Detection Latency
HCRL CH	DoS	[109] 2017	KNN	NA	81.4%	NA	97.4%	70%	NA
	Fuzzy	[107] 2017	KNN	NA	85.5%	NA	100%	74%	NA
	RPM spoofing	[109] 2017	KNN	NA	100%	N/A	100%	100%	NA
	Gear spoofing	[109] 2017	KNN	NA	100%	NA	100%	100%	NA
HCRL OTIDS	DoS	[13] 2019	OCSVM	CPU: Core i7 4 GHz; RAM: 4 GB	NA	NA	93.55%	97.01%	1 ms
	Fuzzy	[96] 2020	Autoencoder	NA	98.09%	NA	99.95%	96.3%	NA
	Impersonation	[96] 2020	Autoencoder	NA	98.09%	NA	99.95%	96.3%	NA

NA, not available. Only includes references that presented numerical results with at least one common metric. The complete table is available in the supplementary material associated with this article.

Timestamp	ID	DLC	payload
1480116775.645067	0153	8	00 a1 20 ff 00 ff 00 bf
1480116775.645303	0220	8	f8 03 f3 03 09 00 43 10
1480116775.645750	02b0	5	20 00 00 07 8d
1480116775.645958	0545	8	c8 0f 00 89 2b 00 2b 00

Fig. 5. CAN bus data frame structure (in raw format).

false speed), max engine coolant temperature (activate engine coolant warning light), and reverse light (do not reflect the actual gear status); and (iii) accelerator attacks, which puts the ECU into a compromised mode. For targeted ID attacks, they injected a message with different payload values for selected signals immediately after seen the legitimate message. For each type of targeted ID attack, they produced masquerade attack versions by removing legitimate messages at the post-processing stage. Hence, frequency-based approaches might fail to detect such attacks. Available attributes are timestamp, CAN channel (always can0), ID, and data field (payload) in hexadecimals. Labels are not available as an attribute. However, they provided attack ID and intervals that can help identify attack messages in the data pre-processing stage. Even though the authors claimed that they injected messages immediately after seeing the legitimate messages, it can be noticed that there are multiple IDs between legitimate and injected messages, making it easy to detect with sequence-based IDSs. However, this can be considered as the most comprehensive CAN dataset available to evaluate and compare CAN IDSs for attacks that change any field of CAN frame.

5.6 Feature Selection and Data Pre-Processing

Data pre-processing and feature selection are also considered as critical steps in AI. In the literature, AI-based IDSs used ID, payload, DLC, and timestamp (time) as features to train AI models. Usually, ID and payload values are in hexadecimal (hex) format. In addition to features in the CAN data frame, one work [178] used voltage signals (physical characteristics) as a feature. Figure 5 depicts the standard format of publicly available CAN data [51]. Table 9 provides the comprehensive summary of feature selection and data pre-processing.

5.6.1 ID-Based Features. In ID-based detection, Kalutarage et al. [71] used IDs in hexadecimal format without using any data pre-processing. Limited or no data pre-processing helps reduce the detection latency of the IDS. However, this limits the wide variety of attack detection capabilities, as some attacks do not significantly change the raw data properties. Marchetti and Stabili [107] also used the hexadecimal IDs and created a transition matrix to learn possible ID transitions. This

approach is computationally more efficient than calculating 2-grams in the work of Kalutarage et al. [71]. SVM-based models [3, 13] used ID frequency as the feature. The selection of this feature as the only feature makes the model lightweight. However, it limits the attack detection capability of infrequent IDs. Both Avatefipour et al. [13] and Marchetti and Stabili [107] assigned labels for each message, whereas Kalutarage et al. [71] assigned labels for message windows. The window-based approach helps reduce the false positives even though it requires a small additional time to process all frames in the window. In addition, certain types of attacks might not create point anomalies. Instead, they might create contextual or collective anomalies. The window-based approach is highly beneficial in identifying these types of contextual and collective anomalies. Han et al. [54] and Kuwahara et al. [86] used observation windows to extract features. Kuwahara et al. [86] used a fixed time window and selected total-counting feature and ID-counting feature. The total counting feature counts the number of messages in a window. In contrast, the ID-counting feature is a vector, each of whose elements is the number of messages associated with each ID. Han et al. [54] considered a window between consecutive CAN IDs and defined it as an event-triggered interval. Mean, variance, first quartile, third quartile, interquartile range, skewness, and kurtosis were calculated for each ID as the features. Sharmin and Mansor [142] calculate the time between consecutive CAN IDs as the feature. All of these feature values [54, 86, 142] change as a result of injection attacks. However, the amount of feature value change depends on the attack injection rate. In addition, these features are insufficient to detect more sophisticated attacks such as masquerade attacks. IDSs proposed by both Jedh et al. [67] and Refat et al. [136] used graph-based techniques to extract features. Refat et al. [136] converted a window of CAN IDs into a graph and extracted graph properties such as the number of nodes, number of edges, radius, diameter, density, reciprocity, average clustering coefficient, and assortative coefficient to use as features for ML models. Similarly, Jedh et al. [67] calculated cosine similarity and Pearson correlation between successive time windows. However, graph-based feature selection might be computationally expensive when a vehicle has a large number of ECUs.

An LSTM-based IDS [32] converted the hexadecimal IDs into integer values from 0 to the number of CAN IDs and then numbers were one hot encoded to consider each CAN ID as a class. Output was the softmax probability for each class. Similarly, for the same task, Rajapaksha et al. [134] converted the hexadecimal IDs into integer values. Instead of one-hot encoding, they created the word vectors for each CAN ID. This helps learn the semantic relationship of CAN IDs better than the one-hot encoding approach. However, the size of the word vectors needs to be selected carefully to keep the model lightweight and efficient. In the work of Seo et al. [140], each digit of raw CAN ID was converted to binary and then to one-hot encode vector (concatenation of three binary numbers of 16 digits), which was finally used as an image for the algorithm. The input size selected was 64, and it assigned one label for an image. If the image included at least one attack packet, it was considered an attack image. Song et al. [148] converted a 29-bit ID to binary and considered 29 consecutive IDs into one frame (making it 29×29 two-dimensional grid data frame). The same logic used by Seo et al. [140] was used to define the class label. In the work of Berger et al. [32], 20 consecutive CAN IDs were selected and converted into one-hot encode vectors. This resulted in a 20×42 (42 IDs) input frame to LSTM. Output was softmax probabilities relevant to 42 IDs. Both of these approaches converted CAN ID sequences to two-dimensional grids. This data structure makes it possible to use image processing algorithms such as CNN on CAN data.

5.6.2 Payload-Based Features. Autoencoder-based models used by some authors [83, 84, 103] split the datasets into groups based on the CAN IDs, and each group was processed independently. Even though this reduces the model complexity of each model, dependencies among CAN IDs cannot be exploited to detect some attacks. Novikova et al. [124] grouped the data considering the

Table 9. Summary of Feature Selection and Data Pre-Processing

Based on	Reference	Data Pre-Processing	Features	Strengths	Weaknesses
CAN ID	[71]	N-gram calculation	ID sequences	Limited data pre-processing	N-gram calculations are computationally expensive for large N
CAN Payload	[120]	Calculate posterior and transition probabilities for a sliding window	Posterior probability, transition probability	Contextual anomaly detection	Higher memory requirement to store all probabilities
CAN Frame	[20, 106, 155]	Time difference calculation for consecutive IDs, convert hexadecimal ID and payload to decimal	Time difference, ID, payload	Limited data pre-processing, computationally efficient	Ineffective features selection for unsupervised learning

The complete table is available in the supplementary material associated with this article.

dependencies among payload signals. This overcomes the aforementioned issue. However, they used a manual approach for the grouping. The SynCAN dataset used to evaluate autoencoder-based models [55, 83, 84, 124] includes pre-processed signal values. This can be done by converting an 8-byte hexadecimal payload into eight integer (or decimal) values and then scaling the integer values to the 0-1 range. This helps avoid slow and unstable training due to the large range of signal values. Instead of treating 8 bytes as eight features, Tomlinson et al. [162] combined these bytes into several fields if they represent a single reading from a sensor. They used an algorithm to concatenate these fields for each CAN ID. Similarly, Fenzl et al. [40] also used a field classification approach to align 8 bytes into several fields. This [40, 162] payload value concatenation helps dimensionality reduction. Concatenation algorithms need to be accurate and efficient to avoid incorrect field classifications.

5.6.3 CAN Frame-Based Features. The majority of CAN frame-based IDSs [20, 106, 155] used the timestamp (as a time interval for consecutive IDs), decimal ID, and payload fields as features. In contrast, limited works [117, 133] used the binary ID and payload instead of decimal conversion. This increases the dimensionality of the features. In the work of Tian et al. [159], in addition to the payload values, the authors created entropy-based features using ID and time. The creation of additional features such as entropy-based features helps detect a wide variety of injection attacks. Zero padding was used by the authors [117, 133, 155, 156] to replace the missing values of the CAN payload. This helps obtain a uniform data field to train AI algorithms. For the DNN-based IDS proposed by Zhang et al. [184], they used the CAN ID, number of occurrences in the past second, relative distance between IDs, and change in system entropy as the features. Zhang and Ma [183] extracted additional features using the CAN payload field. The new features set includes the CAN ID, Hamming distance between the data fields of two normal consecutive CAN ID, entropy of data filed, and bytes of importance (most important two bytes). Usage of the Hamming distance of payload data helps detect attacks on infrequent IDs. However, since all features are calculated for ID groups, inter-correlation among IDs cannot be detected for any feature. This limits the wide range of attack detection capability of the proposed solution. Khan et al. [76] used data pre-processing to enhance the scalability and performance efficiency of the proposed IDS. This included feature conversion, feature reduction, and feature normalization. Principal Component Analysis was used to feature reduction. Experimental results showed that the feature pre-processing led to 19.31% accuracy improvement compared to raw data.

6 AI MODEL SECURITY AND RELIABILITY

AI is rapidly changing the automotive industry. The integration of AI capabilities into the modern automobile adds not only sophistication but also a new attack vector and risks. The Society of Automotive Engineers (SAE) defined six levels of vehicle automation, starting from level 0 to level 5. Level 0 is defined as no automation, whereas level 5 is defined as self-driving automation [132].

Perception, prediction, planning, decision making, and control functions of self-driving cars will be fully controlled by AI models [132]. Hence, the reliability of AI models is a serious issue, especially for sensitive applications like vehicles. Risks associated with these vehicles are safety, liability, privacy, cyber security, and industry influence [152]. However, these intelligent models can improve safety, as 90% of vehicle accidents are due to human errors [146]. Cyber security can be considered as a more serious issue, as this can lead to all other risks listed previously.

AI models are vulnerable to a range of cyber attacks. Three types of attacks target the different phases of the ML life cycle [164]. Evasion attacks perform during inference time and try to introduce inputs that lead to incorrect outputs. Poisoning attacks perform during the training stage and change the training data by inserting, editing, or removing to change the model boundaries. Privacy attacks could target any stage and intend to retrieve sensitive data. Gu et al. [47] demonstrated the vulnerabilities of outsourced training (transfer learning) of AI models such as AlexNet and VGG. They implemented a maliciously trained backdoored neural network (BadNets) for the MNIST dataset and more complex traffic sign detection. It showed that the implemented algorithm could misclassify the stop signs as speed-limit signs by using a Post-It note. Papernot et al. [128] developed a black-box adversary that can observe labels given by a DNN model to chosen inputs. They developed a model to substitute the target DNN. To this end, inputs were synthetically generated and classified by the targeted DNN. Dynamic backdooring attacks—random backdoor, the backdoor generating network (BaN), and the conditional backdoor generating network (c-BaN)—were developed to bypass current state-of-the-art defense mechanisms against backdoor attacks [139]. Jagielski et al. [66] developed a poisoning attack that required minimal knowledge on the learning process of linear regression models and validated with a range of datasets and models. They also developed a defense method against all poisoning attacks. Barreno et al. [18] demonstrated a white-box poisoning attack on an IDS system.

If an attacker compromises an IDS, then it will not be able to detect the attacks on vehicle networks. Therefore, it is important to consider the security of AI-based IDSs at the development and deployment stages. AI-based IDSs are vulnerable to white-box, black-box, and model tampering attacks. In a white-box attack, the attacker has full access and knowledge about the AI model, including learned weights and training data. The attacker of a black-box attack has no access and knowledge about the AI model internals or training data. The attacker can only observe output labels predicted by the AI model to the selected inputs. A model tampering attack is an attack through the tampering of the AI model. Wang et al. [168] developed an LSTM model to detect anomalies in the CAN bus and then used a black-box attack to replace the LSTM model with a new victim model. Only a small sample of testing data was required to train a victim model. It took just 50 man-hours to build the victim model, which led to incorrect predictions. Li et al. [92] used an LSTM-based IDS to detect simple CAN payload attacks with a greater than 98% detection rate. They attacked the LSTM IDS using the fast gradient sign method and the basic iterative method. Under these attacks, detection rates of the IDS were 1.58% and 0.53%, respectively. This highlights the importance of security for IVN IDSs. They proposed an adversarial defending algorithm that provided protection against both fast gradient sign method and basic iterative method attacks.

There are several criteria, such as evaluating the goal of the attack, knowledge required to perform the attack, efficiency of the attack, and availability of mitigation, required to assess the attacks. To increase the security of AI models, a broader array of measures such as legal measures, organizational measures, and technical measures outside the AI system need to be taken [21].

7 DISCUSSION

This article focused on the exploitation of AI techniques for IVN IDSs. A novel taxonomy based on detection features and AI algorithms was used to classify the reviewed works. This section

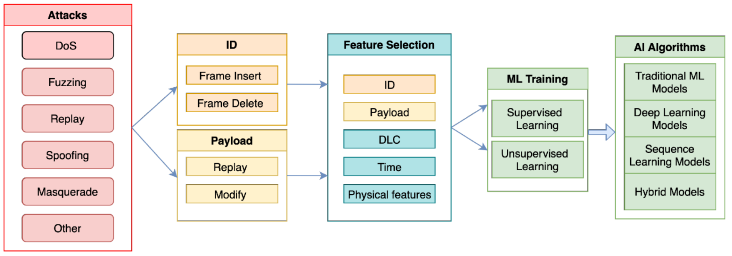

Fig. 6. Development steps of AI-based attack detection in the CAN bus.

discusses the findings of the survey, limitations of current approaches, and future research directions in the development of AI-based IDSs for IVNs.

7.1 Findings

Based on the findings of this review, Figure 6 illustrates the development steps of an AI-based attack detection method in the CAN bus. This includes five stages—namely, attacks, change in the CAN data frames due to these attacks (ID and payload), feature selection, ML training approach, and AI algorithm selection. Listed attacks could change the ID or payload or both fields at the same time to achieve the desired outcome. Frame insertion or deletion is used to change the ID field, whereas replay or modification is used to manipulate the payload field. Selected features should indicate these changes. For instance, selecting the ID as the only feature will limit the detection capability of payload manipulation attacks. Selecting all features will increase the detection power of the algorithm for various attacks with additional computational overhead. However, CAN ID-based IDSs might have higher generalization capability than payload-based IDSs, as the CAN payload is extremely unique to the vehicle brands or models than CAN IDs. The majority of CAN ID-based IDSs have utilized the frequent or sequential behavior of the IDs. Even though the functionality of the IDs of vehicle brands or models is different, frequent or sequential behavior is common for different vehicle brands and models. Therefore, these models have a higher generalization capability. DLC and payload might correlate as DLC is the length of the payload. In this case, DLC can be ignored. More features can be derived through the different fields of the CAN data frame using feature engineering techniques, and this increases the attack detection capability. Physical characteristics such as voltage signals were used as a feature in the work of Xun et al. [178], and this has the capability to identify attack sources that could not identify with other IDSs discussed. Priorities of IDs were not considered in reviewed literature and will be a good feature to explore.

Supervised or unsupervised learning can be used to train traditional ML, DL, sequential learning, and hybrid models. Unsupervised learning algorithms have better capability to detect unknown attacks than supervised learning algorithms. Unsupervised learning requires only benign data (one class) for training and threshold estimation. This is a promising approach for this domain, as collecting benign data is relatively easier than collecting attack data in vehicle networks. This is referred to as one-class classification. OCSVM (a traditional ML model) and autoencoders (DL) were commonly used as the unsupervised learning approaches. Variants of RNN such as LSTM and GRU are capable of capturing long-term and short-term temporal patterns of time series data in IVNs. LSTM and GRU autoencoders were successfully used as unsupervised approaches to detect attacks. Combining LSTM or GRU with other DL algorithms such as CNN (ensemble models) or rule-based models (hybrid models) have increased the attack detection capability for a wide range of attacks. Unsupervised learning tends to produce higher false positives than supervised learning. As a solution, a window-based approach can be used to reduce the false positives. Unsupervised learning can detect

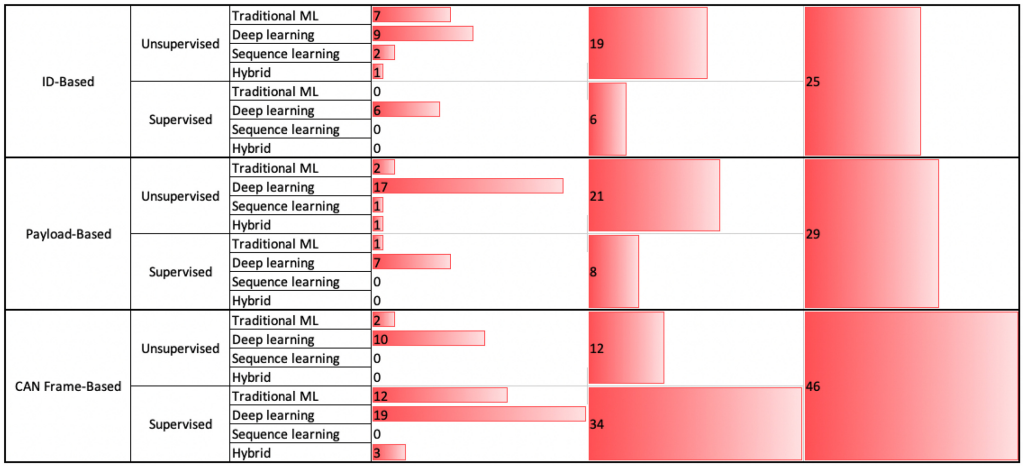

Fig. 7. AI-based IDSs distribution.

a wider range of attacks (including unknown attacks) than supervised learning models. Generally, deep learning models have achieved better accuracy than traditional ML models. However, the high resource requirements and detection latency are the main concerns of DL models given the limited resource availability of IVN devices. Hybrid models and ensemble models have increased the detection power, as these models can improve performance while decreasing the weakness of individual models. A few works have used transfer learning, GAN, and federated learning, which showed promising results in terms of accuracy, new attack detection, and model security. Figure 7 depicts the reviewed AI-based IDS distribution across feature selection, ML training, and AI algorithms. Only two works have used the physical characteristic based features. Therefore, these two works are not included in this distribution. However, 100 works used ID, payload, and CAN frame-based features. The highest number of IDSs are based on the CAN frame. Among these, 34 works have a low generalization capability, as they used supervised learning. In contrast, only a limited number of works have used supervised learning to train ID-based or payload-based IDSs. Due to the complexity of the payload field, the majority of works have used DL-based algorithms to train payload-based IDSs. Overall, 45% of IDSs have used DL-based algorithms to detect attacks on IVNs.

Different attack and deployment environment characteristics require an IDS that employs multiple methods to cover a wide range of attacks with limited resources. Based on the reviewed literature, an unsupervised ensemble model will be the ideal candidate algorithm that can meet this requirement. Table 10 depicts the benefits and drawbacks of AI algorithms used in in-vehicle IDSs.

7.2 Future Research Directions

This section identifies the limitation of current approaches and highlights future research directions for securing IVNs (CAN bus).

7.2.1 Availability of Benchmark Datasets. The performance of an AI-based algorithm highly depends on the data it uses. The usage of low-quality data in AI algorithms leads to bad outputs. The poor quality of publicly available datasets can be identified as a limitation for IDS research in this area. These datasets in particular suffered from the simulation of the attack under realistic conditions. Section 5.5 discussed the benefits and drawbacks of the existing datasets. It is difficult to evaluate, compare, and improve the IDSs without having a proper dataset. There are three reasons

Table 10. Benefits and Drawbacks of Commonly Used AI Algorithms in In-Vehicle IDSs

Algorithms	Benefits	Drawbacks
OCSVM	Use only one-class (benign) data to train the classifier	Non-linear kernel took much time to optimize [30]
SVM	Better learning ability for small samples [145]	Sensitive to kernel function parameters [97]
ANN	Flexibility and adaptability to environmental changes [44], able to train with non-linear data [97]	Long model training time [145] and lack of model explainability [44].
Boosting, KNN, RF, NB	Better learning ability to small samples, train quickly[97], high model explainability [127]	Low accuracy compared to DL models [97]
K-means	Class label not required (unsupervised training)	Sensitive to outliers [127], sensitive to parameter *K* [97]
LSTM	Possibility to use only one-class (benign) data to train the classifier, suitable for sequential data (CAN bus data) [62]	Long model training time, lack of model explainability [91], required a large dataset for training [182]
DNN, CNN, BDN, GAN	High generalization capability [180], good at pattern recognition problems [91]	Long model training time [87], lack of model explainability [91], required a large dataset for training [182]
Autoencoder	Capability to detect point, contextual, and collective anomalies by identifying variable correlations	Computationally expensive, required a large dataset for training
N-gram	Suitable for sequential data (CAN bus data), no domain knowledge required about CAN data, context awareness, possibility to use only one-class (benign) data to train the classifier	Difficulty to capture all data relevant to normal behavior
HMM	Suitable for sequential data (CAN bus data), no domain knowledge required about CAN data, possibility to use only one-class (benign) data to train the classifier	Highly dependent on assumptions about the system [44]

for this limitation. First, it is costly to produce vehicle networks data with real attacks except for simple message injection attacks. Second, there is a risk involved with creating realistic attack data for running vehicles on public roads and, third, issues with the disclosure of sensitive information [166]. Often, researchers used datasets created by themselves with synthetic attacks that were not reflected in real-world situations. Considering the publicly available CAN datasets, the ROAD dataset [166] will be the best dataset to use to evaluate and compare in-vehicle IDSs, as it consists of multiple real attack types along with benign datasets under various driving conditions. Usage of multiple datasets is another feasible solution.

7.2.2 Accuracy and Detection of Low Frequent Attacks. ECUs in a modern vehicle generate about 2,000 CAN frames per second to the CAN bus [140]. Therefore, even 1% of false-negative rate miss 200 attack frames per second. Missing a detection of a particular attack may lead to serious safety problems. Detecting a normal message as an attack also brings unwanted countermeasures that cause inconvenience for the driver. Even though 99% of detection accuracy is a good achievement in other application domains, this might not be enough in this domain. Various attacks and different characteristics make it hard to improve the detection rate. The majority of proposed solutions were not able to detect low frequent (low-volume) attacks, as these attacks have little effect on CAN bus data behavior. DL-based (particularly unsupervised methods) ensemble models and hybrid models are possible future research directions to improve the accuracy and detection capability of low frequent attacks. Moreover, transformer-based models are also a possible direction, as these models have successfully been used in other domains for time series forecasting [174]. Combining CAN data frame features with physical characteristics features to improve the performance will be an interesting direction to study in the future.

7.2.3 Detection Latency. Message transmission in IVNs happens in real time. IVN IDSs should detect and take appropriate countermeasures in real or near real time. However, the majority of reviewed DL-based literature was not able to detect attacks in real or near real time. DL-based IDSs can utilize the large number of computational resources in the cloud to improve the detection time. However, since vehicles are moving objects, connection stability is a key factor to consider for cloud deployments. Edge computing will be another option despite the computationally

constrained environment. This is an area to explore in the future with different experiments under real-world conditions.

7.2.4 Evaluation Metrics. IDSs in the literature evaluated their proposed models using collected real data, public real data, or synthetic data. Since the evaluation of collected real and synthetic data was done under different adversarial settings, it is challenging to compare their security uniformly. Performance comparisons for public real or synthetic datasets are possible, as they share the same benign and attack data. However, the majority of reviewed works did not use common evaluation metrics to compare their security. For example, a few works used accuracy and precision, whereas others used F1-score, precision, and recall as the evaluation metrics. Comparing the performance using only one common metric such as precision or recall is challenging. The Matthews correlation coefficient [26] is considered a more reliable metric for binary classification problems such as anomaly detection. Some works only presented visual evaluations such as bar or line charts. This also makes the model comparison much more difficult. Therefore, it is vital to use a few metrics such as the Matthews correlation coefficient, F1-score, precision, recall, false-positive rate, and false-negative rate to make a fair comparison. Accuracy as a metric is inappropriate in this case, as all discussed attack datasets in Section 5.5 are highly imbalanced. Detection latency is another critical factor for an in-vehicle IDS. However, only limited works evaluated their models for detection latency and discussed the used experimental platform. Hence, including these metrics in the evaluation criteria helps identify more effective methods and improve attack detection in IVNs.

7.2.5 Unsupervised Learning. Unsupervised learning (OCSVM, autoencoders) is well suited for the CAN bus, as CAN bus dataflow is predictable and constant [161]. Another reason is that collecting attack data is more expensive in vehicle networks than benign data collection. In unsupervised learning, only benign data is used to model normal behavior, and a threshold is determined to detect anomalies. However, one major limitation for this approach is the need for a large dataset that sufficiently represents the normal profile. To this end, streaming learning can be considered as a future research direction. The model needs to be deployed in a vehicle, and the parameters and threshold can be updated for a sufficiently large time to cover various normal driving conditions.

7.2.6 Requirement of Large Datasets. Usually, AI algorithms require a large dataset for model training. However, as discussed earlier, the availability of realistic attack and benign datasets is a major limitation in this domain. Learning from a few examples is a key challenge for IVN attack detection. Algorithms such as transfer learning [163], one-shot learning [167], and zero-shot learning [176] were used in other domains such as image recognition and Natural Language Processing applications to address this challenge. Adapting them to vehicle network data could be future research directions to utilize small datasets to detect new attack types.

7.2.7 Cost of Implementation. The majority of reviewed literature was not focused on deployment requirements and countermeasures. ECUs in vehicle networks have limited memory storage, computing power, and bandwidth. IDS development and deployment are bounded by these resources. IDSs can be deployed as host-based IDSs or network-based IDSs. Host-based IDSs are not a viable solution for vehicles, as they require a change in ECUs that are not cost effective. Therefore, deploying a network-based IDS as an additional node in the CAN bus would be the most appropriate solution. Deploying the IDS in clouds can be considered as another feasible solution.

7.2.8 Protecting IDS. Even though AI-based models can identify anomalies in vehicle networks with a high detection rate, these models themselves are vulnerable to cyber attacks such as white-box, black-box, and tempering attacks. None of the discussed literature focuses on protecting the

proposed models from cyber attacks except the models proposed by Aliyu et al. [8] and Li et al. [92]. Aliyu et al. [8] claimed that blockchain technology could be used to improve the security of IDSs. In contrast, Li et al. [92] proposed a defending scheme against adversarial attacks on LSTM-based IDSs. Developing a secure IDS for IVNs under an adversarial setting is a challenging future research direction. Solutions used in other domains can also be adapted to vehicle networks.

8 CONCLUSION

Modern automobiles are equipped with various communication networks, multiple sensors, actuators, cameras, radars, and communication devices to improve performance, efficiency, intelligent services, and safety of passengers. This increased complexity and connectivity make vehicles vulnerable to cyber attacks. Both in-vehicle and VANET networks are exposed to such attacks. IDSs are used to identify cyber attacks in vehicle networks. AI-based IDSs are proven to be effective for vehicle networks due to their high generalization capability.

This study reviewed and categorized 102 recent AI proposals (IDSs) to protect IVNs (particularly on the CAN bus). To this end, we introduced a novel taxonomy based on attacks, CAN data frame features, and supervised and unsupervised AI algorithms. In particular, we outlined attack characteristics, feature usage, effective algorithms to detect these attacks, and the security of AI models. This study also reviewed benchmark datasets available to train and evaluate AI algorithms. Development steps of AI-based attack detection discussed the findings with regard to each step and guided cyber attack detection in the CAN bus using AI algorithms.

Unsupervised learning using traditional ML algorithms such as OCSVM and DL algorithms like autoencoders showed promising results. In general, DL algorithms showed high detection capability over traditional ML models. LSTM and GRU-based ensemble learning and hybrid models (rules and ML based) can be used to overcome many limitations in the attack detection of the CAN bus. Transfer learning, one-shot learning, zero-shot learning, and federated learning can be identified as future trends in CAN bus IDS research. As concluding remarks, we outlined key limitations and reasons for limitations and proposed possible future research directions for IVN security.

REFERENCES

[1] Dogukan Aksu and Muhammed Ali Aydin. 2022. MGA-IDS: Optimal feature subset selection for anomaly detection framework on in-vehicle networks-CAN bus based on genetic algorithm and intrusion detection approach. *Computers & Security* 118 (2022), 102717.

[2] Omar Y. Al-Jarrah, Carsten Maple, Mehrdad Dianati, David Oxtoby, and Alex Mouzakitis. 2019. Intrusion detection systems for intra-vehicle networks: A review. *IEEE Access* 7 (2019), 21266–21289.

[3] Mamdooh Al-Saud, Ali M. Eltamaly, Mohamed A. Mohamed, and Abdollah Kavousi-Fard. 2019. An intelligent data-driven model to secure intravehicle communications based on machine learning. *IEEE Transactions on Industrial Electronics* 67, 6 (2019), 5112–5119.

[4] Theyazn H. H. Aldhyani and Hasan Alkahtani. 2022. Attacks to automatous vehicles: A deep learning algorithm for cybersecurity. *Sensors* 22, 1 (2022), 360.

[5] Laha Ale, Scott A. King, and Ning Zhang. 2021. Deep Bayesian learning for car hacking detection. *arXiv:2112.09333* (2021). https://doi.org/10.48550/ARXIV.2112.09333

[6] Asma Alfardus and Danda B. Rawat. 2021. Intrusion detection system for CAN bus in-vehicle network based on machine learning algorithms. In *Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics, and Mobile Communication Conference (UEMCON'21)*. IEEE, Los Alamitos, CA, 0944–0949. https://doi.org/10.1109/UEMCON53757.2021.9666745

[7] Emad Aliwa, Omer Rana, Charith Perera, and Peter Burnap. 2021. Cyberattacks and countermeasures for in-vehicle networks. *ACM Computing Surveys* 54, 1 (2021), 1–37.

[8] Ibrahim Aliyu, Marco Carlo Feliciano, Sélinde Van Engelenburg, Dong Ok Kim, and Chang Gyoon Lim. 2021. A blockchain-based federated forest for SDN-enabled in-vehicle network intrusion detection system. *IEEE Access* 9 (2021), 102593–102608.

[9] Abdulaziz Alshammari, Mohamed A. Zohdy, Debatosh Debnath, and George Corser. 2018. Classification approach for intrusion detection in vehicle systems. *Wireless Engineering and Technology* 9, 4 (2018), 79–94.
[10] Flora Amato, Luigi Coppolino, Francesco Mercaldo, Francesco Moscato, Roberto Nardone, and Antonella Santone. 2021. CAN-bus attack detection with deep learning. *IEEE Transactions on Intelligent Transportation Systems* 22, 8 (2021), 5081–5090.
[11] Afia Anjum, Paul Agbaje, Sena Hounsinou, and Habeeb Olufowobi. 2022. In-vehicle network anomaly detection using extreme gradient boosting machine. In *Proceedings of the 2022 11th Mediterranean Conference on Embedded Computing (MECO'22)*. IEEE, Los Alamitos, CA, 1–6. https://doi.org/10.1109/MECO55406.2022.9797224
[12] Javed Ashraf, Asim D. Bakhshi, Nour Moustafa, Hasnat Khurshid, Abdullah Javed, and Amin Beheshti. 2020. Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2020), 4507–4518.
[13] Omid Avatefipour, Ameena Saad Al-Sumaiti, Ahmed M. El-Sherbeeny, Emad Mahrous Awwad, Mohammed A. Elmeligy, Mohamed A. Mohamed, and Hafiz Malik. 2019. An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning. *IEEE Access* 7 (2019), 127580–127592.
[14] Omid Avatefipour and Hafiz Malik. 2017. State-of-the-art survey on in-vehicle network communication (CAN-Bus) security and vulnerabilities. *International Journal of Computer Science and Network* 6, 6 (2017), 720–727.
[15] Prashanth Balaji and Majid Ghaderi. 2021. NeuroCAN: Contextual anomaly detection in controller area networks. In *Proceedings of the 2021 IEEE International Smart Cities Conference (ISC2'21)*. IEEE, Los Alamitos, CA, 1–7. https://doi.org/10.1109/ISC253183.2021.9562830
[16] Gianmarco Baldini. 2021. Intrusion detection systems in in-vehicle networks based on bag-of-words. In *Proceedings of the 2021 5th Cyber Security in Networking Conference (CSNet'21)*. IEEE, Los Alamitos, CA, 41–48. https://doi.org/10.1109/CSNet52717.2021.9614644
[17] Vita Santa Barletta, Danilo Caivano, Antonella Nannavecchia, and Michele Scalera. 2020. Intrusion detection for in-vehicle communication networks: An unsupervised Kohonen SOM approach. *Future Internet* 12, 7 (2020), 119.
[18] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D. Joseph, and J. Doug Tygar. 2006. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, Computer, and Communications Security*. ACM, New York, NY, 16–25. https://doi.org/10.1145/1128817.1128824
[19] Dheeraj Basavaraj and Shahab Tayeb. 2022. Towards a lightweight intrusion detection framework for in-vehicle networks. *Journal of Sensor and Actuator Networks* 11, 1 (2022), 6.
[20] Ivo Berger, Roland Rieke, Maxim Kolomeets, Andrey Chechulin, and Igor Kotenko. 2018. Comparative study of machine learning methods for in-vehicle intrusion detection. In *Computer Security*. Springer, 85–101. https://doi.org/10.1007/978-3-030-12786-2_6
[21] Christian Berghoff, Arndt Von Twickel, and Matthias Neu. 2020. Vulnerabilities of connectionist AI applications: Evaluation and defence. *Frontiers in Big Data* 3 (2020), 23.
[22] Safa Boumiza and Rafik Braham. 2017. Intrusion threats and security solutions for autonomous vehicle networks. In *Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA'17)*. IEEE, Los Alamitos, CA, 120–127. https://doi.org/10.1109/AICCSA.2017.42
[23] Zhiqiang Cai, Aohui Wang, Wenkai Zhang, M. Gruffke, and H. Schweppe. 2019. 0-days & mitigations: Roadways to exploit and secure connected BMW cars. *Black Hat USA* 2019 (2019), 39.
[24] Robert N. Charette. 2009. This Car Runs on Code. Retrieved July 1, 2021 from https://spectrum.ieee.org/transportation/systems/this-car-runs-on-code.
[25] Mingqiang Chen, Qingling Zhao, Zhe Jiang, and Rui Xu. 2021. Intrusion detection for in-vehicle CAN networks based on auxiliary classifier GANs. In *Proceedings of the 2021 International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS'21)*. IEEE, Los Alamitos, CA, 186–191.
[26] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21, 1 (2020), 1–13.
[27] Kyong-Tak Cho and Kang G. Shin. 2016. Error handling of in-vehicle networks makes them vulnerable. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, 1044–1055. https://doi.org/10.1145/2976749.2978302
[28] Kyong-Tak Cho and Kang G. Shin. 2016. Fingerprinting electronic control units for vehicle intrusion detection. In *Proceedings of the 25th USENIX Security Symposium (USENIX Security'16)*. 911–927. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho.
[29] Kyong-Tak Cho and Kang G. Shin. 2017. Viden: Attacker identification on in-vehicle networks. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, New York, NY, 1109–1123. https://doi.org/10.1145/3133956.3134001
[30] Valliappa Chockalingam, Ian Larson, Daniel Lin, and Spencer Nofzinger. 2016. Detecting Attacks on the CAN Protocol with Machine Learning. EECS 588: Computer & Network Security. University of Michigan.

[31] Wonsuk Choi, Kyungho Joo, Hyo Jin Jo, Moon Chan Park, and Dong Hoon Lee. 2018. VoltageIDS: Low-level communication characteristics for automotive intrusion detection system. *IEEE Transactions on Information Forensics and Security* 13, 8 (2018), 2114–2129.

[32] Araya Kibrom Desta, Shuji Ohira, Ismail Arai, and Kazutoshi Fujikawa. 2020. ID sequence analysis for intrusion detection in the CAN bus using long short term memory networks. In *Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops'20)*. IEEE, Los Alamitos, CA, 1–6.

[33] Araya Kibrom Desta, Shuji Ohira, Ismail Arai, and Kazutoshi Fujikawa. 2020. MLIDS: Handling raw high-dimensional CAN bus data using long short-term memory networks for intrusion detection in in-vehicle networks. In *Proceedings of the 2020 30th International Telecommunication Networks and Applications Conference (ITNAC'20)*. IEEE, Los Alamitos, CA, 1–7.

[34] Araya Kibrom Desta, Shuji Ohira, Ismail Arai, and Kazutoshi Fujikawa. 2022. Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. *Vehicular Communications* 35 (2022), 100470.

[35] Yongqi Dong, Kejia Chen, Yinxuan Peng, and Zhiyuan Ma. 2022. Comparative study on supervised versus semi-supervised machine learning for anomaly detection of in-vehicle CAN network. *arXiv preprint arXiv:2207.10286* (2022).

[36] Xuting Duan, Huiwen Yan, Daxin Tian, Jianshan Zhou, Jian Su, and Wei Hao. 2021. In-vehicle CAN bus tampering attacks detection for connected and autonomous vehicles using an improved isolation forest method. *IEEE Transactions on Intelligent Transportation Systems*. Early access, December 15, 2021.

[37] Guillaume Dupont, Jerry den Hartog, Sandro Etalle, and Alexios Lekidis. 2019. A survey of network intrusion detection systems for controller area network. In *Proceedings of the 2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES'19)*. IEEE, Los Alamitos, CA, 1–6.

[38] Jürgen Dürrwang, Johannes Braun, Marcel Rumez, Reiner Kriesten, and Alexander Pretschner. 2018. Enhancement of automotive penetration testing with threat analyses results. *SAE International Journal of Transportation Cybersecurity and Privacy* 1, 2 (2018), 91–112.

[39] Zeinab El-Rewini, Karthikeyan Sadatsharan, Daisy Flora Selvaraj, Siby Jose Plathottam, and Prakash Ranganathan. 2020. Cybersecurity challenges in vehicular communications. *Vehicular Communications* 23 (2020), 100214.

[40] Florian Fenzl, Roland Rieke, Yannick Chevalier, Andreas Dominik, and Igor Kotenko. 2020. Continuous fields: Enhanced in-vehicle anomaly detection using machine learning models. *Simulation Modelling Practice and Theory* 105 (2020), 102143.

[41] Florian Fenzl, Roland Rieke, and Andreas Dominik. 2021. In-vehicle detection of targeted CAN bus attacks. In *Proceedings of the 16th International Conference on Availability, Reliability, and Security*. ACM, New York, NY, 1–7. https://doi.org/10.1145/3465481.3465755

[42] Guillermo A. Francia and Eman El-Sheikh. 2021. Applied machine learning to vehicle security. In *Machine Intelligence and Big Data Analytics for Cybersecurity Applications*. Springer, 423–442.

[43] Lulu Gao, Fei Li, Xiang Xu, and Yong Liu. 2019. Intrusion detection system using SOEKS and deep learning for in-vehicle security. *Cluster Computing* 22, 6 (2019), 14721–14729.

[44] Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security* 28, 1–2 (2009), 18–28.

[45] Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, and Witold Klaudel. 2020. Deep learning for in-vehicle intrusion detection system. In *Proceedings of the International Conference on Neural Information Processing*. 50–58.

[46] George Grekousis. 2019. Artificial neural networks and deep learning in urban geography: A systematic review and meta-analysis. *Computers, Environment and Urban Systems* 74 (2019), 244–256.

[47] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg. 2019. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access* 7 (2019), 47230–47244.

[48] Hacking and Countermeasure Research Lab. 2020. CAN Dataset for Intrusion Detection (OTIDS). Retrieved August 1, 2021 from https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset.

[49] Hacking and Countermeasure Research Lab. 2020. CAN Signal Extraction and Translation Dataset. Retrieved July 1, 2022 from https://ocslab.hksecurity.net/Datasets/can-signal-extraction-and-translation-dataset.

[50] Hacking and Countermeasure Research Lab. 2020. Car Hacking: Attack & Defense Challenge. Retrieved August 1, 2021 from https://ocslab.hksecurity.net/Datasets/carchallenge2020.

[51] Hacking and Countermeasure Research Lab. 2020. Car-Hacking Dataset for the Intrusion Detection. Retrieved August 1, 2021 from https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset.

[52] Hacking and Countermeasure Research Lab. 2020. Survival Analysis Dataset for Automobile IDS. Retrieved August 1, 2021 from https://ocslab.hksecurity.net/Datasets/survival-ids.

[53] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. 2018. Anomaly intrusion detection method for vehicular networks based on survival analysis. *Vehicular Communications* 14 (2018), 52–63.
[54] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. 2021. Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network. *IEEE Transactions on Information Forensics and Security* 16 (2021), 2941–2956.
[55] Markus Hanselmann, Thilo Strauss, Katharina Dormann, and Holger Ulmer. 2020. CANet: An unsupervised intrusion detection system for high dimensional CAN bus data. *IEEE Access* 8 (2020), 58194–58205.
[56] Markus Hanselmann, Thilo Strauss, Katharina Dormann, and Holger Ulmer. 2020. SynCAN Dataset. Retrieved August 1, 2021 from https://github.com/etas/SynCAN/blob/master/README.md.
[57] Yuchu He, Zhijuan Jia, Mingsheng Hu, Chi Cui, Yage Cheng, and Yanyan Yang. 2022. The hybrid similar neighborhood robust factorization machine model for CAN bus intrusion detection in the in-vehicle network. *IEEE Transactions on Intelligent Transportation Systems* 23, 9 (2022), 16833–16841.
[58] Thien-Nu Hoang and Daehee Kim. 2022. Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *arXiv preprint arXiv:2204.01193* (2022).
[59] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. 2008. Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures. In *Proceedings of the International Conference on Computer Safety, Reliability, and Security*. 235–248.
[60] Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. 2020. An effective in-vehicle CAN bus intrusion detection system using CNN deep learning approach. In *Proceedings of the 2020 IEEE Global Communications Conference (GLOBECOM'20)*. IEEE, Los Alamitos, CA, 1–6.
[61] Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. 2020. Long short-term memory-based intrusion detection system for in-vehicle controller area network bus. In *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC'20)*. IEEE, Los Alamitos, CA, 10–17.
[62] Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. 2020. LSTM-based intrusion detection system for in-vehicle CAN bus communications. *IEEE Access* 8 (2020), 185489–185502.
[63] Tianxiang Huang, Jianying Zhou, Yi Wang, and Anyu Cheng. 2017. On the security of in-vehicle hybrid network: Status and challenges. In *Proceedings of the International Conference on Information Security Practice and Experience*. 621–637.
[64] Kazuki Iehira, Hiroyuki Inoue, and Kenji Ishida. 2018. Spoofing attack using bus-off attacks against a specific ECU of the CAN bus. In *Proceedings of the 2018 15th IEEE Annual Consumer Communications and Networking Conference (CCNC'18)*. IEEE, Los Alamitos, CA, 1–4.
[65] Mhafuzul Islam, Mashrur Chowdhury, Zadid Khan, and Sakib Mahmud Khan. 2022. Hybrid quantum-classical neural network for cloud-supported in-vehicle cyberattack detection. *IEEE Sensors Letters* 6, 4 (2022), 1–4.
[66] Matthew Jagielski, Alina Oprea, Battista Biggio, Chang Liu, Cristina Nita-Rotaru, and Bo Li. 2018. Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP'18)*. IEEE, Los Alamitos, CA, 19–35.
[67] Mubark Jedh, Lotfi Ben Othmane, Noor Ahmed, and Bharat Bhargava. 2021. Detection of message injection attacks onto the CAN bus using similarities of successive messages-sequence graphs. *IEEE Transactions on Information Forensics and Security* 16 (2021), 4133–4146.
[68] Christian Kaiser, Alexander Stocker, and Andreas Festl. 2020. Automotive CAN Bus Data: An Example Dataset from the AEGIS Big Data Project. Retrieved August 1, 2021 from https://zenodo.org/record/3267184#.YRB6m1NKijR.
[69] Nevrus Kaja. 2019. *Artificial Intelligence and Cybersecurity: Building an Automotive Cybersecurity Framework Using Machine Learning Algorithms*. Ph. D. Dissertation. University of Michigan–Dearborn.
[70] Soner Can Kalkan and Ozgur Koray Sahingoz. 2020. In-vehicle intrusion detection system on controller area network with machine learning models. In *Proceedings of the 2020 11th International Conference on Computing, Communication, and Networking Technologies (ICCCNT'20)*. IEEE, Los Alamitos, CA, 1–6.
[71] Harsha Kumara Kalutarage, M. Omar Al-Kadri, Madeline Cheah, and Garikayi Madzudzo. 2019. Context-aware anomaly detector for monitoring cyber attacks on automotive CAN bus. In *Proceedings of the ACM Computer Science in Cars Symposium*. ACM, New York, NY, 1–8. https://doi.org/10.1145/3359999.3360496
[72] Dong Mug Kang, Sang Hun Yoon, Dae Kyo Shin, Young Yoon, Hyeon Min Kim, and Soo Hyun Jang. 2021. A study on attack pattern generation and hybrid MR-IDS for in-vehicle network. In *Proceedings of the 2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC'21)*. IEEE, Los Alamitos, CA, 291–294.
[73] Min-Joo Kang and Je-Won Kang. 2016. Intrusion detection system using deep neural network for in-vehicle network security. *PLoS One* 11, 6 (2016), e0155781.
[74] Georgios Karopoulos, Georgios Kambourakis, Efstratios Chatzoglou, José L. Hernández-Ramos, and Vasileios Kouliaridis. 2022. Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy. *Electronics* 11, 7 (2022), 1072.

[75] Abdollah Kavousi-Fard, Morteza Dabbaghjamanesh, Tao Jin, Wencong Su, and Mahmoud Roustaei. 2020. An evolutionary deep learning-based anomaly detection model for securing vehicles. *IEEE Transactions on Intelligent Transportation Systems* 22, 7 (2020), 4478–4486.

[76] Izhar Ahmed Khan, Nour Moustafa, Dechang Pi, Waqas Haider, Bentian Li, and Alireza Jolfaei. 2021. An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*. Early access, August 20, 2021.

[77] Zadid Khan, Mashrur Chowdhury, Mhafuzul Islam, Chin-Ya Huang, and Mizanur Rahman. 2020. Long short-term memory neural network-based attack detection model for in-vehicle network security. *IEEE Sensors Letters* 4, 6 (2020), 7500904.

[78] Zadid Khan, Mashrur A. Chowdhury, Mhafuzul Islam, Chin-Ya Huang, and Mizanur Rahman. 2019. In-vehicle false information attack detection and mitigation framework using machine learning and software defined networking. *arXiv abs/1906.10203* (2019).

[79] Takeshi Kishikawa, Ryo Hirano, Yoshihiro Ujiie, Tomoyuki Haga, Hideki Matsushima, Kazuya Fujimura, and Jun Anzai. 2019. Vulnerability of FlexRay and countermeasures. *SAE International Journal of Transportation Cybersecurity and Privacy* 2, 1 (2019), 21–33.

[80] Ch. Kishore, D. Chandrasekhar Rao, and H. S. Behera 2022. Deep learning approach for anamoly detection in CAN bus network: An intelligent LSTM-based intrusion detection system. In *Computational Intelligence in Pattern Recognition*. Lecture Notes in Networks and Systems, Vol. 480. Springer, 531–544.

[81] Dan Klinedinst and Christopher King. 2016. *On Board Diagnostics: Risks and Vulnerabilities of the Connected Vehicle*. Technical Report. CERT Coordination Center, Carnegie Mellon University.

[82] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, et al. 2010. Experimental security analysis of a modern automobile. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy*. IEEE, Los Alamitos, CA, 447–462.

[83] Vipin Kumar Kukkala, Sooryaa Vignesh Thiruloga, and Sudeep Pasricha. 2020. INDRA: Intrusion detection using recurrent autoencoders in automotive embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 39, 11 (2020), 3698–3710.

[84] Vipin Kumar Kukkala, Sooryaa Vignesh Thiruloga, and Sudeep Pasricha. 2021. LATTE: LSTM self-attention based anomaly detection in embedded automotive platforms. *ACM Transactions on Embedded Computing Systems* 20, 5s (2021), 1–23.

[85] B. Vinodh Kumar and J. Ramesh. 2014. Automotive in vehicle network protocols. In *Proceedings of the 2014 International Conference on Computer Communication and Informatics*. IEEE, Los Alamitos, CA, 1–5.

[86] Takuya Kuwahara, Yukino Baba, Hisashi Kashima, Takeshi Kishikawa, Junichi Tsurumi, Tomoyuki Haga, Yoshihiro Ujiie, Takamitsu Sasaki, and Hideki Matsushima. 2018. Supervised and unsupervised intrusion detection based on CAN message frequencies for in-vehicle network. *Journal of Information Processing* 26 (2018), 306–313.

[87] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C. Suh, Ikkyun Kim, and Kuinam J. Kim. 2019. A survey of deep learning-based network anomaly detection. *Cluster Computing* 22, 1 (2019), 949–961.

[88] H. Lee, S. H. Jeong, and H. K. Kim. 2017. OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In *Proceedings of the 2017 15th Annual Conference on Privacy, Security, and Trust (PST'17)*. 57–5709.

[89] Nandi Leslie. 2021. An unsupervised learning approach for in-vehicle network intrusion detection. In *Proceedings of the 2021 55th Annual Conference on Information Sciences and Systems (CISS'21)*. IEEE, Los Alamitos, CA, 1–4.

[90] Matan Levi, Yair Allouche, and Aryeh Kontorovich. 2018. Advanced analytics for connected car cybersecurity. In *Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring'18)*. IEEE, Los Alamitos, CA, 1–7.

[91] Hang Li. 2018. Deep learning for natural language processing: Advantages and challenges. *National Science Review* 5, 1 (2018), 24–26.

[92] Yi Li, Jing Lin, and Kaiqi Xiong. 2021. An adversarial attack defending system for securing in-vehicle networks. In *Proceedings of the 2021 IEEE 18th Annual Consumer Communications and Networking Conference (CCNC'21)*. IEEE, Los Alamitos, CA, 1–6.

[93] Alessandro Liberati, Douglas G. Altman, Jennifer Tetzlaff, Cynthia Mulrow, Peter C. Gøtzsche, John P. A. Ioannidis, Mike Clarke, Philip J. Devereaux, Jos Kleijnen, and David Moher. 2009. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration. *Journal of Clinical Epidemiology* 62, 10 (2009), e1–e34.

[94] Hsiao-Chung Lin, Ping Wang, Kuo-Ming Chao, Wen-Hui Lin, and Jia-Hong Chen. 2022. Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks. *Electronics* 11, 14 (2022), 2180.

[95] Jiaying Lin, Yehua Wei, Wenjia Li, and Jing Long. 2021. Intrusion detection system based on deep neural network and incremental learning for in-vehicle CAN networks. In *Proceedings of the International Conference on Ubiquitous Security*. 255–267.

[96] Yubin Lin, Chengbin Chen, Fen Xiao, Omid Avatefipour, Khalid Alsubhi, and Arda Yunianta. 2020. An evolutionary deep learning anomaly detection framework for in-vehicle networks-CAN bus. *IEEE Transactions on Industry Applications*. Early access, July 17, 2020.

[97] Hongyu Liu and Bo Lang. 2019. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences* 9, 20 (2019), 4396.

[98] Jiajia Liu, Shubin Zhang, Wen Sun, and Yongpeng Shi. 2017. In-vehicle network attacks and countermeasures: Challenges and future directions. *IEEE Network* 31, 5 (2017), 50–58.

[99] Nai-Wei Lo and Hsiao-Chien Tsai. 2007. Illusion attack on VANET applications—A message plausibility problem. In *Proceedings of the 2007 IEEE GLOBECOM Workshops*. IEEE, Los Alamitos, CA, 1–8.

[100] Wei Lo, Hamed Alqahtani, Kutub Thakur, Ahmad Almadhor, Subhash Chander, and Gulshan Kumar. 2022. A hybrid deep learning based intrusion detection system using spatial-temporal representation of in-vehicle network traffic. *Vehicular Communications* 35 (2022), 100471.

[101] Siti-Farhana Lokman, Abu Talib Othman, and Muhammad-Husaini Abu-Bakar. 2019. Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. *EURASIP Journal on Wireless Communications and Networking* 2019, 1 (2019), 1–17.

[102] Siti Farhana Lokman, Abu Talib Othman, Shahrulniza Musa, and Muhamad Husaini Abu Bakar. 2019. Deep contractive autoencoder-based anomaly detection for in-vehicle controller area network (CAN). In *Progress in Engineering Technology*. Springer, 195–205.

[103] Stefano Longari, Daniel Humberto Nova Valcarcel, Mattia Zago, Michele Carminati, and Stefano Zanero. 2021. CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network. *IEEE Transactions on Network and Service Management* 18, 2 (2021), 1913–1924. https://doi.org/10.1109/TNSM.2020.3038991

[104] George Loukas, Eirini Karapistoli, Emmanouil Panaousis, Panagiotis Sarigiannidis, Anatolij Bezemskij, and Tuan Vuong. 2019. A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles. *Ad Hoc Networks* 84 (2019), 124–147.

[105] George Loukas, Tuan Vuong, Ryan Heartfield, Georgia Sakellari, Yongpil Yoon, and Diane Gan. 2017. Cloud-based cyber-physical intrusion detection for vehicles using deep learning. *IEEE Access* 6 (2017), 3491–3508.

[106] Haoyu Ma, Jianqiu Cao, Bo Mi, Darong Huang, Yang Liu, and Shaoqian Li. 2022. A GRU-based lightweight system for CAN intrusion detection in real time. *Security and Communication Networks* 2022 (2022), 5827056.

[107] Mirco Marchetti and Dario Stabili. 2017. Anomaly detection of CAN bus messages through analysis of ID sequences. In *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV'17)*. IEEE, Los Alamitos, CA, 1577–1583.

[108] Moti Markovitz and Avishai Wool. 2017. Field classification, modeling and anomaly detection in unknown CAN bus networks. *Vehicular Communications* 9 (2017), 43–52.

[109] Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. 2017. Car hacking identification through fuzzy logic algorithms. In *Proceedings of the 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'17)*. IEEE, Los Alamitos, CA, 1–7.

[110] Sk. Tanzir Mehedi, Adnan Anwar, Ziaur Rahman, and Kawsar Ahmed. 2021. Deep transfer learning based intrusion detection system for electric vehicular networks. *Sensors* 21, 14 (2021), 4736.

[111] Charlie Miller and Chris Valasek. 2015. Remote Exploitation of an Unaltered passenger vehicle. *Black Hat USA* 2015, S91 (2015), 1–91.

[112] Charlie Miller and Chris Valasek. 2016. Can Message Injection. Retrieved July 1, 2021 from http://illmatics.com/can%20message%20injection.pdf.

[113] Omar Minawi, Jason Whelan, Abdulaziz Almehmadi, and Khalil El-Khatib. 2020. Machine learning-based intrusion detection system for controller area networks. In *Proceedings of the 10th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet'20)*. ACM, New York, NY, 41–47. https://doi.org/10.1145/3416014.3424581

[114] Tarek Moulahi, Salah Zidi, Abdulatif Alabdulatif, and Mohammed Atiquzzaman. 2021. Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus. *IEEE Access* 9 (2021), 99595–99605.

[115] Michael Müter and Naim Asaj. 2011. Entropy-based anomaly detection for in-vehicle networks. In *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV'11)*. IEEE, Los Alamitos, CA, 1110–1115.

[116] Michael Müter, André Groll, and Felix C. Freiling. 2010. A structured approach to anomaly detection for in-vehicle networks. In *Proceedings of the 2010 6th International Conference on Information Assurance and Security*. IEEE, Los Alamitos, CA, 92–98.

[117] Shu Nakamura, Koh Takeuchi, Hisashi Kashima, Takeshi Kishikawa, Takashi Ushio, Tomoyuki Haga, and Takamitsu Sasaki. 2021. In-vehicle network attack detection across vehicle models: A supervised-unsupervised hybrid approach. In *Proceedings of the 2021 IEEE International Intelligent Transportation Systems Conference (ITSC'21)*. IEEE, Los Alamitos, CA, 1286–1291.

[118] Minki Nam, Seungyoung Park, and Duk Soo Kim. 2021. Intrusion detection method using bi-directional GPT for in-vehicle controller area networks. *IEEE Access* 9 (2021), 124931–124944.
[119] Harini Narasimhan, R. Vinayakumar, and Nazeeruddin Mohammad. 2021. Unsupervised deep learning approach for in-vehicle intrusion detection system. *IEEE Consumer Electronics Magazine*. Early access, September 30, 2021.
[120] Sandeep Nair Narayanan, Sudip Mittal, and Anupam Joshi. 2016. OBD_SecureAlert: An anomaly detection system for vehicles. In *Proceedings of the 2016 IEEE International Conference on Smart Computing (SMARTCOMP'16)*. IEEE, Los Alamitos, CA, 1–6.
[121] Ahmed NasrEldin, Ayman M. Bahaa-Eldin, and Mohamed A. Sobh. 2021. In-vehicle intrusion detection based on deep learning attention technique. In *Proceedings of the 2021 16th International Conference on Computer Engineering and Systems (ICCES'21)*. IEEE, Los Alamitos, CA, 1–7.
[122] Nicolas Navet, Yeqiong Song, Francoise Simonot-Lion, and Cédric Wilwert. 2005. Trends in automotive communication systems. *Proceedings of the IEEE* 93, 6 (2005), 1204–1223.
[123] Sen Nie, Ling Liu, and Yuefeng Du. 2017. Free-fall: Hacking Tesla from wireless to CAN bus. *Black Hat USA* 25 (2017), 1–16.
[124] Elena Novikova, Vu Le, Matvey Yutin, Michael Weber, and Cory Anderson. 2020. Autoencoder anomaly detection on large CAN bus data. In *Proceedings of the 2nd Workshop on Deep Learning Practice for High-Dimensional Sparse Data with KDD 2020 (DLP-KDD'20)*.
[125] Internet of Business. 2020. Connected Cars Report: 125 Million Vehicles by 2022, 5G Coming. Retrieved July 1, 2021 from https://internetofbusiness.com/worldwide-connected-car-market-to-top-125-million-by-2022/.
[126] Laboratory of Cryptography and System Security. 2020. CrySyS Lab Dataset. Retrieved August 1, 2021 from https://www.crysys.hu/research/vehicle-security/.
[127] Salima Omar, Asri Ngadi, and Hamid H. Jebur. 2013. Machine learning techniques for anomaly detection: An overview. *International Journal of Computer Applications* 79, 2 (2013), 33–41.
[128] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. 506–519.
[129] Seunghyun Park and Jin-Young Choi. 2020. Hierarchical anomaly detection model for in-vehicle networks using machine learning algorithms. *Sensors* 20, 14 (2020), 3934.
[130] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR. *Black Hat Europe* 11, 2015 (2015), 995.
[131] Tomáš Pevnỳ. 2016. Loda: Lightweight on-line detector of anomalies. *Machine Learning* 102, 2 (2016), 275–304.
[132] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. 2020. Securing connected and autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. *IEEE Communications Surveys & Tutorials* 22, 2 (2020), 998–1026.
[133] Hongmao Qin, Mengru Yan, and Haojie Ji. 2021. Application of controller area network (CAN) bus anomaly detection based on time series prediction. *Vehicular Communications* 27 (2021), 100291.
[134] Sampath Rajapaksha, Harsha Kalutarage, M. Omar Al-Kadri, Garikayi Madzudzo, and Andrei V. Petrovski. 2022. Keep the moving vehicle secure: Context-aware intrusion detection system for in-vehicle CAN bus security. In *Proceedings of the 2022 14th International Conference on Cyber Conflict: Keep Moving! (CyCon'22)*, Vol. 700. IEEE, Los Alamitos, CA, 309–330.
[135] Gopi Krishnan Rajbahadur, Andrew J. Malton, Andrew Walenstein, and Ahmed E. Hassan. 2018. A survey of anomaly detection for connected vehicle cybersecurity and safety. In *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV'18)*. IEEE, Los Alamitos, CA, 421–426.
[136] Rafi Ud Daula Refat, Abdulrahman Abu Elkhail, Azeem Hafeez, and Hafiz Malik. 2021. Detecting CAN bus intrusion by applying machine learning method to graph based features. In *Proceedings of the SAI Intelligent Systems Conference*. 730–748.
[137] Abdul Rehman, Saif Ur Rehman, Mohibullah Khan, Mamoun Alazab, and Thippa Reddy. 2021. CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE Transactions on Network Science and Engineering* 8, 2 (2021), 1456–1466.
[138] Marcel Rumez, Jinghua Lin, Thomas Fuchß, Reiner Kriesten, and Eric Sax. 2020. Anomaly detection for automotive diagnostic applications based on N-grams. In *Proceedings of the 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC'20)*. IEEE, Los Alamitos, CA, 1423–1429.
[139] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. 2020. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675* (2020).
[140] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. 2018. GIDS: Gan based intrusion detection system for in-vehicle network. In *Proceedings of the 2018 16th Annual Conference on Privacy, Security, and Trust (PST'18)*. IEEE, Los Alamitos, CA, 1–6.

[141] Shashwat Khandelwal and Shanker Shreejith. 2022. A lightweight multi-attack CAN intrusion detection system on hybrid FPGAs. *International Conference on Field Programmable Logic and Applications (FPL'22)*.
[142] Shaila Sharmin and Hafizah Mansor. 2021. Intrusion detection on the in-vehicle network using machine learning. In *Proceedings of the 2021 3rd International Cyber Resilience Conference (CRC'21)*. IEEE, Los Alamitos, CA, 1–6.
[143] Shaila Sharmin, Hafizah Mansor, Andi Fitriah Abdul Kadir, and Normaziah A. Aziz. 2021. Using streaming data algorithm for intrusion detection on the vehicular controller area network. In *Proceedings of the International Conference on Ubiquitous Security*. 131–144.
[144] Dongxian Shi, Ming Xu, Ting Wu, and Liang Kou. 2021. Intrusion detecting system based on temporal convolutional network for in-vehicle CAN networks. *Mobile Information Systems* 2021 (2021), 1440259.
[145] Jayveer Singh and Manisha J. Nene. 2013. A survey on machine learning techniques for intrusion detection systems. *International Journal of Advanced Research in Computer and Communication Engineering* 2, 11 (2013), 4349–4355.
[146] Santokh Singh. 2015. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey*. Technical Report. U.S. Department of Transportation.
[147] Hyun Min Song and Huy Kang Kim. 2021. Self-supervised anomaly detection for in-vehicle network using noised pseudo normal data. *IEEE Transactions on Vehicular Technology* 70, 2 (2021), 1098–1108.
[148] Hyun Min Song, Ha Rang Kim, and Huy Kang Kim. 2016. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *Proceedings of the 2016 International Conference on Information Networking (ICOIN'16)*. IEEE, Los Alamitos, CA, 63–68.
[149] Hyun Min Song, Jiyoung Woo, and Huy Kang Kim. 2020. In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications* 21 (2020), 100198.
[150] Hiroki Suda, Masanori Natsui, and Takahiro Hanyu. 2018. Systematic intrusion detection technique for an in-vehicle network based on time-series feature extraction. In *Proceedings of the 2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL'18)*. IEEE, Los Alamitos, CA, 56–61.
[151] Heng Sun, Miaomiao Chen, Jian Weng, Zhiquan Liu, and Guanggang Geng. 2021. Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism. *IEEE Transactions on Vehicular Technology* 70, 10 (2021), 10880–10893.
[152] Araz Taeihagh and Hazel Si Min Lim. 2019. Governing autonomous vehicles: Emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport Reviews* 39, 1 (2019), 103–128.
[153] Daiki Tanaka, Makoto Yamada, Hisashi Kashima, Takeshi Kishikawa, Tomoyuki Haga, and Takamitsu Sasaki. 2019. In-vehicle network intrusion detection and explanation using density ratio estimation. In *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC'19)*. IEEE, Los Alamitos, CA, 2238–2243.
[154] Vinayak Tanksale. 2020. Anomaly detection for controller area networks using long short-term memory. *IEEE Open Journal of Intelligent Transportation Systems* 1 (2020), 253–265.
[155] Shahroz Tariq, Sangyup Lee, Huy Kang Kim, and Simon S. Woo. 2020. CAN-ADF: The controller area network attack detection framework. *Computers & Security* 94 (2020), 101857.
[156] Shahroz Tariq, Sangyup Lee, and Simon S. Woo. 2020. CANTransfer: Transfer learning based intrusion detection on a controller area network using convolutional LSTM network. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. 1048–1055.
[157] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz. 2016. Anomaly detection in automobile control network data with long short-term memory networks. In *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA'16)*. IEEE, Los Alamitos, CA, 130–139.
[158] Sooryaa Vignesh Thiruloga, Vipin Kumar Kukkala, and Sudeep Pasricha. 2022. TENET: Temporal CNN with attention for anomaly detection in automotive cyber-physical systems. In *Proceedings of the 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC'22)*. IEEE, Los Alamitos, CA, 326–331.
[159] Daxin Tian, Yuzhou Li, Yunpeng Wang, Xuting Duan, Congyu Wang, Wenyang Wang, Rong Hui, and Peng Guo. 2017. An intrusion detection system based on machine learning for CAN-bus. In *Proceedings of the International Conference on Industrial Networks and Intelligent Systems*. 285–294.
[160] Andrew Tomlinson, Jeremy Bryans, and Siraj Ahmed Shaikh. 2018. Towards viable intrusion detection methods for the automotive controller area network. In *Proceedings of the 2nd ACM Computer Science in Cars Symposium*. 1–9.
[161] Andrew Tomlinson, Jeremy Bryans, and Siraj Ahmed Shaikh. 2018. Using a one-class compound classifier to detect in-vehicle network attacks. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1926–1929.
[162] Andrew Tomlinson, Jeremy Bryans, and Siraj Ahmed Shaikh. 2021. Using internal context to detect automotive controller area network attacks. *Computers & Electrical Engineering* 91 (2021), 107048.
[163] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global, 242–264.

[164] Ankit Tripathi. 2019. Machine Learning: With Great Power Come New Security Vulnerabilities. Retrieved August 1, 2021 from https://securityintelligence.com/machine-learning-with-great-power-come-new-security-vulnerabilities/.
[165] Department of Mathematics TU Eindhoven and Computer Science. 2020. TU Eindhoven CAN Bus Intrusion Dataset. Retrieved August 1, 2021 from https://doi.org/10.4121/uuid:b74b4928-c377-4585-9432-2004dfa20a5d
[166] Miki E. Verma, Michael D. Iannacone, Robert A. Bridges, Samuel C. Hollifield, Bill Kay, and Frank L. Combs. 2020. ROAD: The real ORNL automotive dynamometer controller area network intrusion detection dataset (with a comprehensive CAN IDS dataset survey and guide). *arXiv preprint arXiv:2012.14600* (2020).
[167] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. *Advances in Neural Information Processing Systems* 29 (2016), 3630–3638.
[168] Yi Wang, Dan Wei Ming Chia, and Yajun Ha. 2020. Vulnerability of deep learning model based anomaly detection in vehicle network. In *Proceedings of the 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS'20)*. IEEE, Los Alamitos, CA, 293–296.
[169] Armin Wasicek, Mert D. Pesé, André Weimerskirch, Yelizaveta Burakova, and Karan Singh. 2017. Context-aware intrusion detection in automotive control systems. In *Proceedings of the 5th ESCAR USA Conference*. 21–22.
[170] Marc Weber, Simon Klug, Eric Sax, and Bastian Zimmer. 2018. Embedded hybrid anomaly detection for automotive CAN communication. In *Proceedings of the 9th European Congress on Embedded Real Time Software and Systems (ERTS'18)*.
[171] Pengcheng Wei, Bo Wang, Xiaojun Dai, Li Li, and Fangcheng He. 2022. A novel intrusion detection model for the CAN bus packet of in-vehicle network based on attention mechanism and autoencoder. *Digital Communications and Networks*. Early access, May 5, 2022.
[172] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. 1–10.
[173] Samuel Woo, Hyo Jin Jo, and Dong Hoon Lee. 2014. A practical wireless attack on the connected car and security protocol for in-vehicle CAN. *IEEE Transactions on Intelligent Transportation Systems* 16, 2 (2014), 993–1006.
[174] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. 2020. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317* (2020).
[175] Wufei Wu, Renfa Li, Guoqi Xie, Jiyao An, Yang Bai, Jia Zhou, and Keqin Li. 2019. A survey of intrusion detection for in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems* 21, 3 (2019), 919–933.
[176] Yongqin Xian, Bernt Schiele, and Zeynep Akata. 2017. Zero-shot learning—The good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4582–4591.
[177] Junchao Xiao, Hao Wu, and Xiangxue Li. 2019. Internet of Things meets vehicles: Sheltering in-vehicle network through lightweight machine learning. *Symmetry* 11, 11 (2019), 1388.
[178] Yijie Xun, Yilin Zhao, and Jiajia Liu. 2022. VehicleEIDS: A novel external intrusion detection system based on vehicle voltage signals. *IEEE Internet of Things Journal* 9, 3 (2022), 2124–2133.
[179] Clinton Young, Joseph Zambreno, Habeeb Olufowobi, and Gedare Bloom. 2019. Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test* 36, 6 (2019), 48–55.
[180] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* (2016).
[181] Jiayan Zhang, Fei Li, Haoxi Zhang, Ruxiang Li, and Yalin Li. 2019. Intrusion detection system using deep learning for in-vehicle security. *Ad Hoc Networks* 95 (2019), 101974.
[182] Lei Zhang, Ji Liu, Bob Zhang, David Zhang, and Ce Zhu. 2019. Deep cascade model-based face recognition: When deep-layered learning meets small data. *IEEE Transactions on Image Processing* 29 (2019), 1016–1029.
[183] Linxi Zhang and Di Ma. 2022. A hybrid approach toward efficient and accurate intrusion detection for in-vehicle networks. *IEEE Access* 10 (2022), 10852–10866.
[184] Linxi Zhang, Lyndon Shi, Nevrus Kaja, and D. Ma. 2018. A two-stage deep learning approach for CAN intrusion detection. In *Proceedings of the Ground Vehicle Systems Engineering Technology Symposium (GVSETS'18)*. 1–11.
[185] Aiguo Zhou, Zhenyu Li, and Yong Shen. 2019. Anomaly detection of CAN bus messages using a deep neural network for autonomous vehicles. *Applied Sciences* 9, 15 (2019), 3174.
[186] Wu Zhou, Hao Fu, and Shray Kapoor. 2021. CANGuard: Practical intrusion detection for in-vehicle network via unsupervised learning. In *Proceedings of the 2021 IEEE/ACM Symposium on Edge Computing (SEC'21)*. IEEE, Los Alamitos, CA, 454–458.
[187] Konglin Zhu, Zhicheng Chen, Yuyang Peng, and Lin Zhang. 2019. Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM. *IEEE Transactions on Vehicular Technology* 68, 5 (2019), 4275–4284.

Received 1 September 2021; revised 6 August 2022; accepted 25 October 2022