Computer Networks 277 (2026) 112031

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey paper

A survey of learning-based intrusion detection systems for in-vehicle networks

Muzun Althunayyan [a,b,*], Amir Javed [a], Omer Rana [a]

[a] Cardiff University School of Computer Science & Informatics, Cardiff, CF24 3AA, United Kingdom
[b] Majmaah University Computer Sciences and Information Technology College, Al Majma'ah, 15362, Saudi Arabia

ARTICLE INFO

Keywords:
CAN bus
In-vehicle network
Cyberattack
Intrusion detection system
Machine learning
Deep learning
Federated learning

ABSTRACT

Connected and Autonomous Vehicles (CAVs) have advanced modern transportation by improving the efficiency, safety, and convenience of mobility through automation and connectivity, yet they remain vulnerable to cybersecurity threats, particularly through the insecure Controller Area Network (CAN) bus. Cyberattacks can have devastating consequences in connected vehicles, including the loss of control over critical systems, necessitating robust security solutions. In-vehicle Intrusion Detection Systems (IDSs) offer a promising approach by detecting malicious activities in real time. This survey provides a comprehensive review of state-of-the-art research on learning-based in-vehicle IDSs, focusing on Machine Learning (ML), Deep Learning (DL), and Federated Learning (FL) approaches. Based on the reviewed studies, we critically examine existing IDS approaches, categorising them by the types of attacks they detect-known, unknown, and combined known-unknown attacks-while identifying their limitations. We also review the evaluation metrics used in research, emphasising the need to consider multiple criteria to meet the requirements of safety-critical systems. Additionally, we analyse FL-based IDSs and highlight their limitations. By doing so, this survey helps identify effective security measures, address existing limitations, and guide future research toward more resilient and adaptive protection mechanisms, ensuring the safety and reliability of CAVs.

1. Introduction

Connected and Autonomous Vehicles (CAVs) are expected to play an important role in future transportation systems [1], offering transformative benefits in safety, mobility, efficiency, and economic productivity. For instance, the Society of Motor Manufacturers and Traders (SMMT) projects that widespread CAV adoption could contribute £62 billion annually to the UK economy by 2030 [2]. However, the same technological advancements that enable these benefits also introduce new vulnerabilities, expanding the attack surface and exposing CAVs to sophisticated and evolving cyber threats [3].

CAVs rely on Electronic Control Units (ECUs) to manage and control various functions. These ECUs communicate via standardised in-vehicle communication protocols, such as Controller Area Network (CAN), Local Interconnect Network (LIN), FlexRay, and Media Oriented Systems Transport (MOST). Among these protocols, the CAN bus is the most widely adopted due to its speed, reliability, and ease of use [4]. Although originally designed for industrial applications, the CAN bus has become the de facto standard for in-vehicle communication [5]. Despite its advantages, the CAN protocol lacks fundamental security features, including sender authentication and encryption, as it was not originally developed with security considerations [6].

The increasing interconnectivity of CAVs exposes them to a range of cyberattacks. Attackers can access modern vehicles either physically, through ports such as USB or the onboard diagnostic (OBD)-II port, or remotely through wireless technologies such as Wi-Fi, Bluetooth, and LTE [7]. In 2023, the number of large-scale incidents, potentially affecting thousands to millions of mobility assets, grew 2.5-fold compared to 2022. Additionally, 95% of cyberattacks are conducted remotely, with 85% being long-range [8]. These vulnerabilities make vehicles susceptible to attacks that could have devastating consequences, including loss of control over critical systems like braking, steering, and acceleration [9]. A recent incident [10] involved cybersecurity researcher Ian Tabor discovering tampering on his Toyota RAV4, particularly around the front bumper and headlight area. Initially suspecting vandalism, Tabor soon realised the vehicle had been targeted by a cyberattack. Investigations revealed that attackers had accessed the car's CAN bus through exposed wiring, allowing them to inject malicious signals. Through this manipulation, the attackers were able to gain entry and start the engine, successfully stealing the vehicle without using a key. Moreover, a

* Corresponding author.
E-mail addresses: AlthunayyanMS@cardiff.ac.uk (M. Althunayyan), javeda7@cardiff.ac.uk (A. Javed), ranaof@cardiff.ac.uk (O. Rana).

https://doi.org/10.1016/j.comnet.2026.112031
Received 9 April 2025; Received in revised form 22 December 2025; Accepted 16 January 2026
Available online 23 January 2026

notorious example is the Jeep hack, where attackers remotely manipulated the vehicle's braking and steering functions, leading to dangerous driving conditions [11]. Similarly, vulnerabilities in BMW and Toyota Lexus models have been exploited, demonstrating the persistent threat to vehicle security [12,13]. Such incidents emphasise the critical need for strong security measures to defend against both data breaches and physical threats.

Given the severity of these threats, CAN bus security has become a major research focus. According to McKinsey's analysis, by 2030, nearly 95% of the new vehicles will be connected to external networks, further highlighting the need for effective security solutions [14]. One promising approach is the implementation of **Intrusion Detection Systems (IDSs)**, which monitor network traffic for malicious activity. In the context of in-vehicle networks, an IDS is typically installed on an ECU and analyses incoming messages to detect abnormalities. However, conventional IDS technologies designed for traditional networks cannot be applied directly to in-vehicle systems due to resource constraints and the real-time requirements of automotive environments.

Research on the development of in-vehicle IDSs has expanded considerably in recent years, fueled by the discovery of various vulnerabilities and the urgent need to improve the security of in-vehicle networks and detect cyberattacks. Researchers have explored various approaches to building these systems. IDSs can be classified as either signature-based, for detecting known attacks, or anomaly-based, for identifying new, unknown attacks [15]. Anomaly-based IDSs are further categorised into statistical, Machine Learning (ML), rule-based, and physical fingerprinting methods [16].

This survey specifically examines the development of learning-based in-vehicle IDSs, with a focus on Machine Learning (ML), Deep Learning (DL), and Federated Learning (FL) approaches. It aims to identify effective security strategies, overcome existing challenges, and guide future research toward more robust and adaptive protection mechanisms to enhance the reliability and safety of CAVs. The emphasis on ML- and DL-based approaches is driven by their strong generalization capabilities and ability to process large volumes of traffic data [16]. Additionally, FL has recently gained attention among researchers due to its potential to enhance both security and privacy.

In this survey, we review the state-of-art research on ML-based, DL-based and FL-based in-vehicle IDSs, aiming to identify limitations and research gaps in the current literature.

Contribution To summarise, the contributions of this paper are as follows:

- We present a comprehensive literature review employing a structured search strategy to systematically gather research papers published up to January 2025.
- We provide an overview of the CAN protocol, its vulnerabilities, entry points, and attack scenarios, offering an understanding of the CAN bus protocol while exploring potential attack scenarios and their impacts.
- We introduce a classification framework for IDS methodologies based on the types of attacks they detect, including known, unknown, and combined known-unknown threats. Additionally, we present summary tables and highlight the limitations of each approach to identify research gaps.
- We analyse the evaluation metrics used in research studies, emphasising the importance of considering multiple factors-such as performance, time complexity, and memory overhead-when developing in-vehicle IDSs to ensure they meet the requirements of safety-critical systems.
- We provide insights into FL-based IDSs, discussing their advantages while also identifying limitations in addressing security and privacy challenges within connected vehicle environments.
- We outline key open challenges and propose future research directions to enhance the development of more robust, efficient, and effective in-vehicle intrusion detection approaches.

The remainder of this paper is organised as follows. Section 2 presents the contextual and background information. Section 3 reviews similar surveys and highlights our contributions. Section 4 outlines the search methodology used to collect relevant papers. Section 5 reviews existing ML- and DL-based in-vehicle IDSs, while Section 6 focuses on FL-based IDSs. Section 7 explores prospective research directions. Finally, Section 8 concludes the paper.

2. Background

This section provides a brief overview of in-vehicle protocols, with a particular focus on the CAN protocol, including its description, functionality, and aspects relevant to cyberattacks. In addition, it examines the vulnerabilities, entry points, and attack scenarios of CAN.

2.1. In-vehicle network

The in-vehicle network serves as the internal communication system connecting various ECUs within a vehicle [17]. These ECUs are interconnected embedded devices that handle key vehicle operations, such as engine management, airbag control, and climate control. The number and type of ECUs in a vehicle vary depending on the manufacturer and model, with modern vehicles incorporating up to 100 ECUs alongside basic functions [18]. These ECUs, along with sensors, actuators, radars, cameras, and communication devices, collaborate to enhance vehicle performance, efficiency, smart functionalities, and safety by gathering and analysing various data [19]. ECUs communicate using standard protocols such as CAN, FlexRay, LIN, and MOST [20]. CAN is considered the de facto protocol among in-vehicle communication protocols [4].

2.2. Controller area network

The Controller Area Network (CAN) protocol, developed by Robert Bosch in 1985, was designed to reduce the weight, complexity, and cost of wiring. Due to its high speed and efficiency, CAN has become the most widely used in-vehicle communication protocol in connected and autonomous vehicles [4]. CAN operates as a message-based broadcast protocol, where the ECUs transmit data in pre-defined frames. Since the system uses a broadcast mechanism, each message is sent to all ECUs on the network.

2.3. CAN bus data frame

The structure of a CAN frame is defined by a database-like file called the DataBase CAN (DBC) file, which is confidential and proprietary to the vehicle manufacturer. This file contains essential information about the representation of CAN bus data [21]. A CAN data frame comprises seven fields that enable communication between ECUs. Fig. 1 illustrates the standard CAN frame format, which includes the following fields:

- **Start of Frame (SOF):** This field serves to notify other nodes of the start of a CAN frame transmission by transmitting a dominant '0' bit.
- **Arbitration Field:** This field contains the Identifier (ID), which specifies the target ECU and determines the message priority; lower ID values indicate higher priority. In the standard format, the ID is 11 bits long, while the extended format uses 29 bits. The field also includes the Remote Transmission Request (RTR) bit, which distinguishes between data frames and remote frames.
- **Control Field:** This field is 6 bits long and includes a 4-bit Data Length Code (DLC), which indicates the length of the data field, an Identifier Extension (IDE) bit, which specifies whether the ID field is standard (11 bits) or extended (29 bits), and a Reserved Bit (RB) for future use.
- **Data Field:** Also referred to as the payload, this field contains the actual vehicle parameter values interpreted by the receiving ECU, with a size ranging from 0 to 8 bytes (0 to 64 bits).

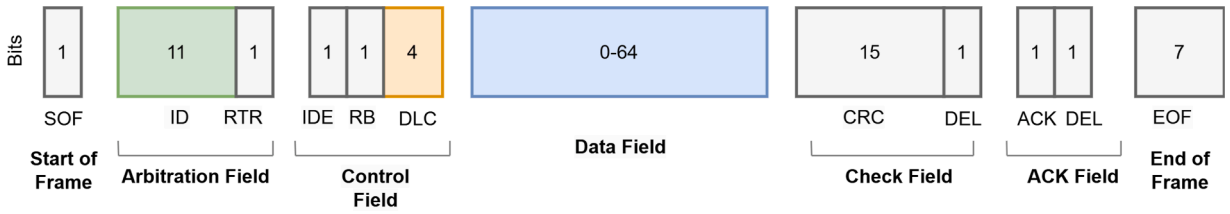


Fig. 1. Standard CAN data frame.

- **Check Field:** This field consists of a 15-bit Cyclic Redundancy Check (CRC) followed by a 1-bit delimiter (DEL), and is used to detect errors and maintain data integrity during message transmission by verifying the validity of the frame.
- **Acknowledge Field (ACK):** This field consists of 2 bits: a 1-bit ACK and a 1-bit DEL. The ACK bit is used to receive confirmation from the receiving node that the CAN message was received correctly.
- **End of Frame (EOF):**This field signals the end of the CAN message transmission.

In this survey, we focus on the coloured fields (ID, DLC, and Data) shown in Fig. 1.

2.4. CAN vulnerabilities

The CAN bus was introduced to reduce costs, simplify installation, and improve real-time communication efficiency within vehicles. However, it is vulnerable to cyberattacks due to several inherent vulnerabilities [7,22,23], including the following:

- *Lack of authentication:* Due to the lack of authentication on the CAN bus, any ECU can transmit a frame using the CAN ID of another ECU [9]. Each ECU broadcasts and receives all data on the bus, then determines whether a message is intended for it. However, the CAN protocol is inherently unable to prevent unauthorised devices from joining the network and sending malicious messages to all ECUs. As a result, attackers can exploit compromised ECUs to spoof and send fake CAN packets, leading to spoofing and message injection attacks [16].
- *Lack of encryption:* Due to time constraints, CAN messages are not encrypted [7], allowing cyberattackers to easily capture and analyse them for further attacks. Lack of encryption makes CAN traffic vulnerable to sniffing, spoofing, modification, and replay attacks [9].
- *Broadcast domain:* The CAN bus functions as a broadcast domain, where all ECUs receive the transmitted frames. Each ECU then checks the data and determines whether to process or disregard it [18]. If an ECU is compromised, it can intercept and monitor all messages transmitted across the CAN network, enabling an eavesdropping attack [24].
- *ID-based priority:* The CAN network prioritises messages based on their IDs, with lower IDs having higher priority [16]. Attackers can exploit this by repeatedly sending frames with low IDs, resulting in a Denial-of-Service (DoS) attack [21].
- *External Interfaces:* The attack surface of the CAN bus network is expanded by external interfaces such as the OBD-II port, used for vehicle maintenance and diagnostics; the Telematics Unit, which provides connectivity to the vehicle via Wi-Fi, Bluetooth, GPS, and mobile data interfaces; and the Infotainment Unit, which delivers information and entertainment to the driver through a head display unit, including features like CD/DVD players and USB ports. These interfaces create additional entry points for potential cyberattacks [16,25].

2.5. In-vehicle network entry points

Attackers may reach the CAN bus or specific ECUs through either direct physical access or remote connections [7,22,26,27]. These entry points serve as gateways for initiating a range of attacks, exploiting the inherent vulnerabilities described in Section 2.4 in-vehicle networks. This section discusses the entry points attackers can exploit to access the in-vehicle network, either physically or remotely.

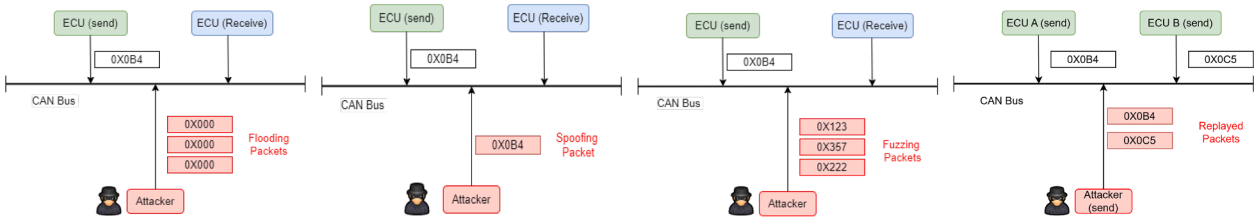
2.5.1. Physical access

Physical access allows an attacker–such as a mechanic, valet, car renter, or anyone with even brief access to the vehicle–to directly interact with its internal systems. This access, even for a short time, can provide opportunities to exploit vulnerabilities through various physical entry points, including:

- **OBD-II Port:** The OBD-II port, commonly located under the dashboard in most vehicles, provides the simplest and most direct access to a vehicle's primary CAN buses. This port offers sufficient access to potentially compromise the full range of automotive systems [27]. Designed primarily for vehicle maintenance and engine diagnostics, the OBD-II port allows mechanics to connect scanning tools and capture data packets generated by malfunctioning subsystems. Despite its intended purpose, the port's accessibility makes it a significant security vulnerability. Attackers can easily connect to the OBD-II port to extract information or install malware onto the vehicle's systems, disconnecting afterward to leave no physical evidence [28]. Alternatively, attackers may deploy a remote device to the port, or enabling continuous data collection or exploitation over time. Since the OBD-II port is required for maintenance and diagnostics, it will always pose a security risk [22].
- **Aftermarket Components:** Peripheral components such as USB ports, CD players, and third-party add-ons also pose security risks [22,23]. For example, malicious devices, including FM radios, USB connectors, or CD players purchased from unverified or aftermarket sources, can introduce malware into the vehicle's system. While these components may be more affordable, they can compromise the vehicle's security [28].

2.5.2. Remote access

An external attacker can exploit wireless interfaces commonly implemented in modern vehicles, such as Bluetooth, Wi-Fi, cellular networks, and GPS, without requiring physical proximity to the vehicle. Once these interfaces are accessed, the attacker can transmit malicious commands or traffic over the CAN bus network [7]. Koscher et al. [29] highlight the feasibility of executing various types of remote injection attacks on in-vehicle networks. For example, vulnerabilities in telematics systems or vehicle-to-cloud communications can enable the remote injection of messages, disrupting the network. Specific methods include using malicious Windows Media Audio (WMA) files or sending malicious packets to the telematics unit via 3G Internet Relay Chat (IRC) [29]. Moreover, Woo et al. [30] conducted a wireless attack, successfully taking control of a target vehicle by utilising malware installed on a smartphone. These examples highlight the significant security risks posed by wireless interfaces in connected vehicles.



(a) DoS attack (b) Spoofing attack (c) Frame fuzzification attack (d) Replay attack

Fig. 2. CAN bus attacks.

2.6. Attack scenarios

Since an attacker may access the in-vehicle network, either physically or remotely, through one of the entry points outlined in Section 2.5, the following are common attack scenarios:

2.6.1. Denial-of-service (DoS) attack

- **Attack Definition and Method:** A Denial-of-Service (DoS) attack aims to prevent normal system functioning by blocking messages from reaching their destination [28]. One common method to achieve this is to increase the busload using high-priority IDs [16]. Since message priority on the CAN bus is determined by the message ID, an attacker can exploit the arbitration mechanism by flooding the network with low-ID (i.e. high-priority) frames, thereby preventing other ECUs from transmitting [31]. Fig. 2a illustrates how a dominant message with CAN ID 0x0000 delays one with a lower priority, such as ID 0x0B4.
- **Attack Scenario:** We assume the attacker has compromised the in-vehicle network, either physically or remotely, through one of the entry points outlined in Section 2.5. Leveraging this access, the attacker injects high-priority messages into the CAN bus without requiring prior knowledge of the CAN bus traffic. The arbitration mechanism prioritises these malicious messages, taking control of the bus and blocking critical communications, such as those from the engine control unit or braking system. For instance, while the vehicle is in motion, an attacker carrying out this attack could disable cruise control or activate emergency braking, preventing critical messages from reaching the appropriate ECU in time and creating potentially hazardous driving conditions. Within seconds, the network's capacity becomes overwhelmed, causing delays that severely compromise vehicle safety.
- **Attack Impact:** A successful DoS attack not only delays normal messages by occupying the bus [32], but also prevents other ECUs from transmitting frames to the in-vehicle network, significantly impacting network availability [33]. Such attacks can lead to a complete breakdown of ECU communication and severe disruption of the entire CAN bus network system [23,34], compromising the safety of drivers, occupants, and other road users [35].

2.6.2. Spoofing attack

- **Attack Definition and Method:** During a spoofing attack, an unauthorised attacker targets valid CAN IDs and injects fake messages to control specific functions. Since CAN IDs appear legitimate, distinguishing between real and spoofed messages becomes challenging, leading to system malfunctions [31]. Fig. 2b illustrates a spoofing attack where an attacker, using the spoofed CAN ID 0x0B4, targets the legitimate CAN ID 0x0B4.
- **Attack Scenario:** We assume the attacker has compromised the in-vehicle network, either physically or remotely, through one of the entry points outlined in Section 2.5.In this attack, we assume that the attacker has some knowledge of CAN bus traffic, and one method to achieve this is by connecting a malicious device to eavesdrop on all broadcast traffic, capturing data transmitted across the network. During this reconnaissance phase, the attacker analyses the traffic to identify patterns in ECU behaviour, such as specific CAN IDs, payload structures, and message transmission intervals. Armed with this knowledge, the attacker selects a target ECU, such as the speedometer. Next, the attacker gains remote access to the internal network and crafts spoofed messages by replicating the target ECU's CAN ID and injecting false speed readings into the bus.
- **Attack Impact:** Spoofing attacks can cause system malfunctions and disrupt vehicle operations [31]. They pose significant threats to personal safety, particularly when targeting critical ECUs responsible for essential functions such as braking or steering [36].

2.6.3. Frame fuzzification attack

- **Attack Definition and Method:** The goal of a frame fuzzification attack is to inject random messages into the CAN bus network, making them appear as legitimate traffic. Attackers may either use prior knowledge of CAN IDs and payloads obtained through CAN bus sniffing or carry out the attack blindly, treating the CAN system as a black box [37]. In such an attack, the attacker might alter the CAN ID, the CAN payload, or both simultaneously [7]. Since the range of valid CAN packets is relatively small, even simple fuzzing of packets can cause significant damage [28]. Fig. 2c illustrates a frame fuzzification attack, where the attacker generates and injects random CAN IDs (e.g. 0x123, 0x357, and 0x222), which are illegitimate. As a result, all ECUs receive a high volume of functional messages. For example, Chockalingam et al. [29] introduced Gaussian noise to create a frame fuzzification attack on CAN data.
- **Attack Scenario:** We assume the attacker has compromised the in-vehicle network, either physically or remotely, through one of the entry points outlined in Section 2.5. Without prior knowledge of CAN frames, the attacker is able to inject random malicious CAN frames. Using techniques such as fuzzing, the attacker transmits random or malformed messages into the CAN bus to provoke unintended system behaviours or identify exploitable vulnerabilities that could be leveraged in future attacks. Additionally, through reverse engineering, the attacker monitors legitimate traffic to deduce the structure and purpose of CAN messages, enabling the creation of malicious packets to execute specific commands targeting particular ECUs.
- **Attack Impact:** Frame fuzzification attacks can compromise ECUs, triggering unexpected vehicle behaviours such as steering wheel shaking, erratic signal lights, and unintended gear shifts [32,34]. These behaviours can confuse the driver, potentially resulting in poor decisions or accidents. Such attacks not only disrupt normal vehicle functions but also threaten operational integrity, compromise data privacy, and endanger personal safety, posing significant risks to passengers and other road users.

2.6.4. Replay attack

- **Attack Definition and Method:** In replay attacks, an attacker captures a legitimate message and resends it later without any changes [38]. For example, an attacker can store a speedometer value and retransmit it to the network at a later time [4]. Fig. 2c illustrates a replay attack, where the attacker sniffs the legitimate CAN messages (e.g., 0X0B4 and 0X0C5) and later re-injects them into the CAN bus.

- **Attack Scenario:** We assume the attacker has gained access to the in-vehicle network, either physically or remotely, through one of the entry points outlined in Section 2.5. Without prior knowledge of CAN traffic, the attacker connects a malicious device to sniff and store legitimate messages. These are later re-injected into the network unmodified, inserted between original transmissions.
- **Attack Impact:** Even though this attack is easy to carry out, as it requires no prior knowledge of traffic operation, it can pose serious safety threats to both vehicles and passengers [16]. Koscher et al. [28] demonstrated replay attacks to manipulate the radio and various body control module functions in the CAN bus. Although the replayed packet is a valid subsequence, the replayed packet disrupts the original packet sequence. Consequently, this can lead to severe issues such as continuous CAN packet transmission requests, deadline violations, and inversion of the CAN arbitration priority scheme. Furthermore, the altered packet sequence prevents the vehicle from functioning properly, as the packets are no longer transmitted sequentially, violating protocol requirements [21].

3. Related work

This section presents a review of current survey literature on IDS approaches applied to in-vehicle networks, highlighting their contributions and how our survey differs. Several studies have contributed taxonomic frameworks and structured classifications to better characterise the landscape of IDS development in this domain. Most of these surveys examine the security of in-vehicle networks more broadly and include IDS as one of several approaches, alongside ML and DL [4,7,9,21,24–26,38–44]. Karopoulos et al.[39] compiled a meta-taxonomy that consolidates the key classification features of in-vehicle IDSs proposed in existing surveys, offering a unified perspective on their development. Dupont et al.[24] categorised in-vehicle IDSs based on the required message count for attack detection, the data utilised, and the design of the detection model. Tomlinson et al.[40] categorised CAN IDS approaches into signature detection and anomaly detection, with the latter further subdivided into statistical, knowledge-based, and ML methods. Rajbahadur et al.[41] conducted a survey on anomaly detection for securing CAVs, introducing a taxonomy comprising three main categories and nine subcategories. They further classified the surveyed studies across 38 dimensions. While the study offers valuable insights, it lacks individual paper summaries and practical implementation guidance. Lampe and Meng [44] reviewed automotive intrusion detection methodologies, categorising IDSs into non-learning, traditional ML, and DL approaches. They further classified IDSs along six dimensions: analysis, deployment, detection, evaluation, learning, and monitoring.

Wu et al. [25] categorised in-vehicle IDSs into fingerprinting, parameter monitoring, information theory, and ML-based approaches. Al-Jarrah et al.[4] reviewed intra-vehicle IDSs and categorised them into flow-level, payload-level, and integrated approaches. Jo et al.[38] classified in-vehicle countermeasures into four categories: preventative protection, IDSs, authentication, and post-protection, and further divided IDS techniques into CAN packet-based and ECU hardware characteristic-based approaches. Loukas et al. [42] proposed a detailed taxonomy emphasising IDS characteristics and architectures across different vehicle platforms, categorising audit techniques into statistical,ML , and rule-based methods. Lokman et al.[21] presented a taxonomy for classifying IDS research according to four dimensions: deployment strategies, attack techniques, technical challenges, and detection methods, and further categorised anomaly-based IDSs into frequency-based, ML-based, statistical, and hybrid approaches. Young et al.[9] categorised CAN bus IDSs into signature-based and anomaly-based approaches, while Quadar et al.[43] classified detection methods into fingerprint-based, time- and frequency-based, and ML-based categories. However, the number of ML-based IDSs reviewed in [4,9,21,25,38,42,43] remains limited. In contrast to surveys focusing exclusively on IDSs, Aliwa et al. [7] adopted a broader perspective by combining cryptographic solutions with IDS approaches to protect vehicular data. Similarly, Limbasiya et al.[26] conducted a systematic survey that extensively analysed various Attack Detection and Prevention System (ADPS) categories for CAVs.

Table 1
Comparison with in-vehicle IDS surveys.

Reference	Year	Search Strategy	ML-DL Specific	FL-based IDSs	Evaluation metrics
[40]	2018		○[a]		
[41]	2018	●[a]	○[a]		
[4]	2019		○[a]		●[a]
[25]	2019		○[a]		
[42]	2019		○[a]		
[21]	2019		○[a]		
[9]	2019		○[a]		
[24]	2019		○[a]		
[38]	2021		○[a]		
[7]	2021		○[a]		
[39]	2022		○[a]		
[26]	2022	●[a]	○[a]		
[48]	2023		●[a]		●[a]
[16]	2023	●[a]	●[a]		○
[46]	2023		●[a]		○[a]
[45]	2023		●[a]		○[a]
[44]	2023		○[a]		○[a]
[43]	2024		○[a]		
[47]	2024		●[a]	○	○[a]
Our Survey	2025	●[a]	●[a]	●[a]	●[a]

[a] ●: Extensive, ○: Partial

Other surveys have focused exclusively on ML/DL-based IDSs [16,45–48]. Rajapaksha et al.[16] adopted the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) methodology and proposed a taxonomy of AI-based IDSs. Similarly, Nagarajan et al.[45] presented a comprehensive review of ML-based IDSs for in-vehicle and inter-vehicle communications. Focusing more specifically on DL approaches, Lampe and Meng [46] provided a comprehensive overview of DL-based IDSs in automotive networks, categorising them based on their topologies and techniques, such as DNN-, CNN-, LSTM-, attention-, transformer-, and GAN-based IDSs. Expanding the scope, Almehdhar et al. [47] categorised IDS techniques into conventional ML, DL, and hybrid models. They also explored emerging technologies such as FL and Transfer Learning. Similarly, Taslimasa et al.[48] conducted an extensive review of IDSs proposed for Internet of Vehicles (IoV) networks that utilise ML and DL algorithms.

Even though existing in-vehicle surveys have made significant contributions to the field, they have certain limitations. As shown in Table 1, few surveys follow a structured search methodology to ensure full coverage and a comprehensive review. In addition, none of these surveys review FL-based IDS, except for the work in [47], which briefly presents some studies in this area. Although Chellapandi et al. [49] provide a survey on FL for CAVs, they do not include any in-vehicle IDSs and instead focus on FL applications such as driver monitoring, motion control, trajectory and steering angle prediction, and object detection. Lastly, although some surveys have reviewed the performance metrics used in the reviewed papers, we provide a comprehensive review of all evaluation metrics, including performance, time, and memory requirements, emphasising the need to include these metrics to develop deployable solutions. **To the best of our knowledge, this survey is the first to provide a comprehensive review of ML, DL, and FL-based IDS for in-vehicle networks.** Table 1 compares this survey with prior studies on in-vehicle IDSs, emphasising the main contributions of the current work.

4. Methodology

In this section, we outline the search strategy used to collect the reviewed papers.

4.1. Search strategy

We followed Kitchenham's method [50], a well-established guide for identifying relevant literature. Although originally designed for software engineering, it has been widely applied in other fields, including cybersecurity [51]. We conducted an automatic search using Google Scholar to minimise publisher bias [52] and identify key sources. Based on this, we compiled a list of publishers and conferences for a subsequent manual search. To ensure comprehensive coverage, we employed a snowballing approach to locate related papers. Relevant results were filtered and analysed (from any time up to and including January 2025) to construct this literature review. Fig. 3 shows the adopted search strategy.

4.1.1. Data sources and search strategy

To begin, we formulated several search queries on Google Scholar using keywords that combine terms representing our area of research and those frequently found in paper keywords. Logical operators "AND" and "OR" were utilised to ensure comprehensive results. These queries generated a range of papers, some of which were only loosely relevant. We then performed a hybrid search with advanced queries across selected journals and databases, including IEEE Xplore, Scopus, ACM, MDPI, Springer, and ScienceDirect. Three search strategies were employed during this process: automatic, manual, and snowballing.

1. **Automatic search** We carried out this stage using the advanced search function in Google Scholar, employing key terms such as "controller area network", "CAN bus", "in-vehicle", "intrusion detection system", "IDS", "anomaly detection", "unknown attacks", and "federated". Initially, using the filter "anywhere in the article," we retrieved an unwieldy number of results (53,890). To refine this, we applied the filter "in the title of the article," reducing the results to 95 papers. Only peer-reviewed papers were included in our analysis.
2. **Manual search** In this stage, we applied more complex queries, including specific journals and libraries, such as IEEE Xplore, Scopus, ACM, MDPI, Springer, and ScienceDirect.
3. **Snowballing** The snowballing technique was applied to the papers identified through automatic and manual searches. This approach included both forward and backward snowballing. Forward snowballing (or citation analysis) locates papers that are cited in the papers found in the initial stages. Backward snowballing (or reference analysis) looks at the reference lists of the papers found in the initial search process. References included in the selected papers were chosen based on a review of the title, abstract, and the paper's structure.

4.1.2. Selection strategy

The selection strategy applied inclusion and exclusion criteria, followed by a quality assessment to ensure high-quality studies were selected.

- **Filtering Irrelevant Papers**: Several papers gathered through manual, automated, and snowballing methods were not directly relevant to our study and were therefore eliminated. Elimination was conducted in two steps. First, papers were excluded based on the title, keywords, abstract, and, when necessary, the conclusion. This step determined whether a paper progressed to the next stage. Elimination was performed after each search (automated, manual, and snowballing) to reduce the number of papers. Those that passed the initial stage were then assessed using the inclusion and exclusion criteria.
- **Inclusion and Exclusion Criteria** At this stage, specific inclusion and exclusion criteria were defined, whereby any paper fulfilling one or more exclusion criteria was removed from consideration. The exclusion criteria for each paper were as follows: (i) not written in English; (ii) was a review or survey paper; (iii) lacked a full version (e.g., only a poster or abstract); (iv) did not employ an ML or DL approach; (v) required reverse engineering; (vi) used other data alongside CAN bus data; and (vii) employed other approaches such as statistical, rule-based, or physical fingerprinting methods. Papers that were not excluded were then evaluated according to an inclusion list of other criteria. If no inclusion criteria were met, the paper was rejected. The inclusion criteria were as follows: (i) focused on the CAN bus protocol rather than other in-vehicle protocols; and (ii) focused on attack detection as the primary goal. Non-peer-reviewed papers were included only if they were strictly relevant to the topic and had a high citation rate, or if the author was well-known in the field.

As shown in Fig. 3, reading the full paper is the final step in the search and selection process, filtering out the collected papers from the previous steps. The papers from the automatic search were reduced from 10 to 9 after filtering by reading the full text. The manual search yielded 59 papers from an initial 1,831; snowballing reduced 57 papers to 18. Combined with 9 from the automatic search, a total of 86 papers were selected for analysis. Of these, 38 focused on known attack detection, 28 on unknown attack detection, 11 on IDSs capable of detecting both known and unknown attacks, and 9 on FL-based IDSs. Fig. 4 illustrates the number of collected papers in each category, highlighting that known attack detection is the most researched area, while significantly less work has been conducted on IDSs capable of detecting both known and unknown attacks, as well as on FL-based IDSs.

5. Intrusion detection systems for in-vehicle networks

In this section, we begin by introducing IDSs for in-vehicle networks and highlighting the differences between in-vehicle IDSs and those used for other applications. We then provide an overview of in-vehicle IDS approaches. Additionally, we categorise the collected papers into three categories: known attack detection, unknown attack detection, and work capable of detecting both known and unknown attacks for analysis. Fig. 5 illustrates the categories and subcategories of the reviewed literature. Lastly, we review all the evaluation metrics employed across the reviewed studies.

5.1. Intrusion detection system for in-vehicle networks

According to NIST SP 800-94, intrusion detection involves "monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents" [53]. Therefore, an IDS is typically recognised as a software or hardware system that automatically identify suspicious activity in a network [9]. In vehicle networks, IDSs are crucial for identifying malicious attacks [7]. They can be implemented as either host-based or network-based systems [9]. Host-based IDSs are installed on each vehicle's ECU, allowing comprehensive monitoring of internal ECU operations. In contrast, network-based IDSs are deployed within the CAN network or central gateways to oversee all network traffic. However, implementing host-based IDSs in vehicles is not feasible, as they demand ECU modifications that are not cost-effective [7]. In contrast, adding a network-based IDS as a standalone node on the CAN bus is a more feasible and practical solution, as it avoids the need for any CAN bus modifications [16]. Unlike IDSs in other applications, in-vehicle IDSs are limited by computational power, memory constraints, and communication capabilities. This is because modern ECUs in vehicles are primarily powered by 32-bit embedded processors, with limited computational performance and memory resources [25].

5.2. Overview of in-vehicle IDS approaches

In recent years, there has been a substantial increase in research on in-vehicle IDSs, motivated by the urgent need to strengthen in-vehicle network security and detect cyber attacks. Various strategies have been examined by researchers for designing such systems. IDSs can be classified as either signature-based, for detecting known attacks, or anomaly-based, for identifying new, unknown attacks [15].

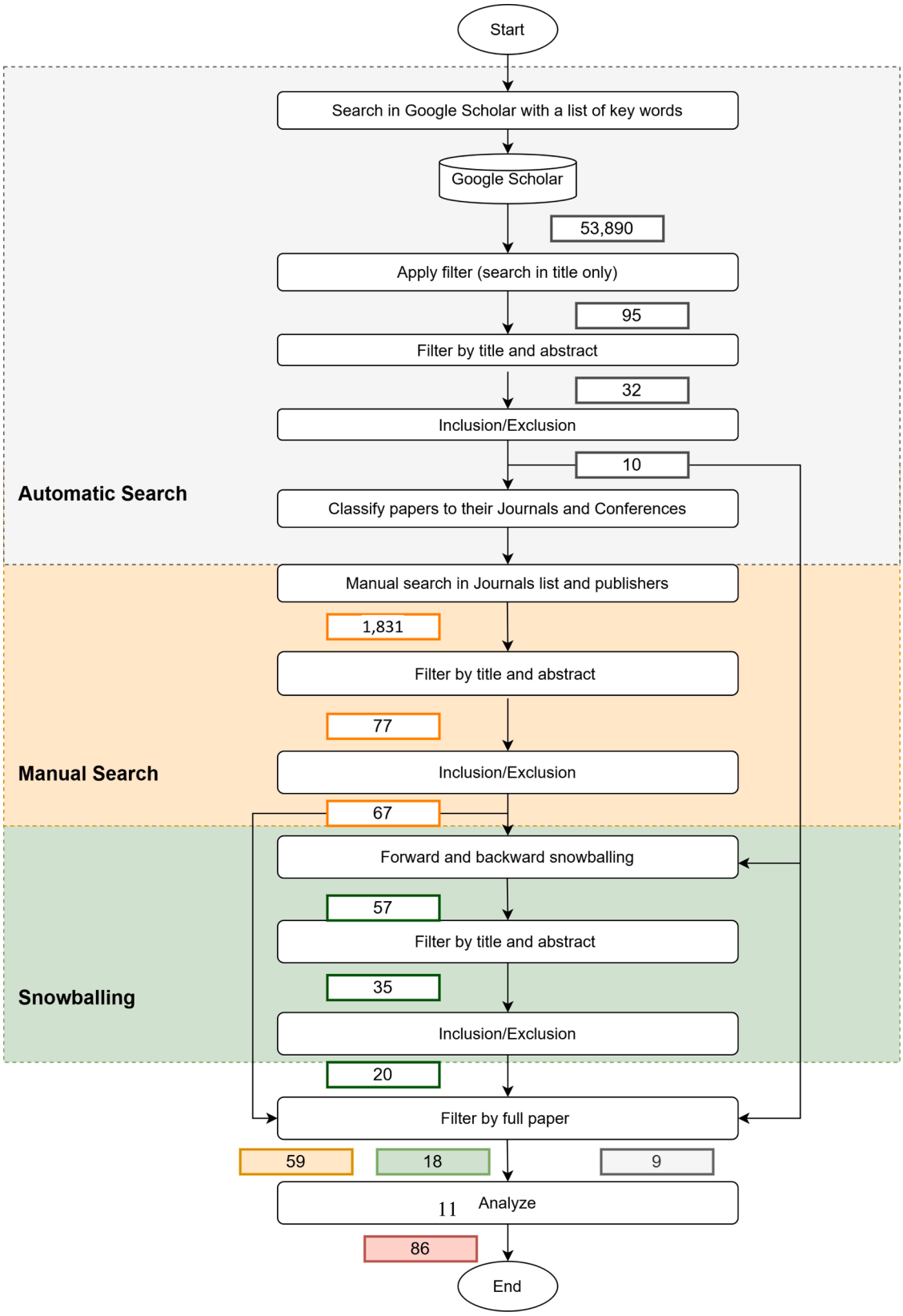


Fig. 3. Search and selection processes flowchart.

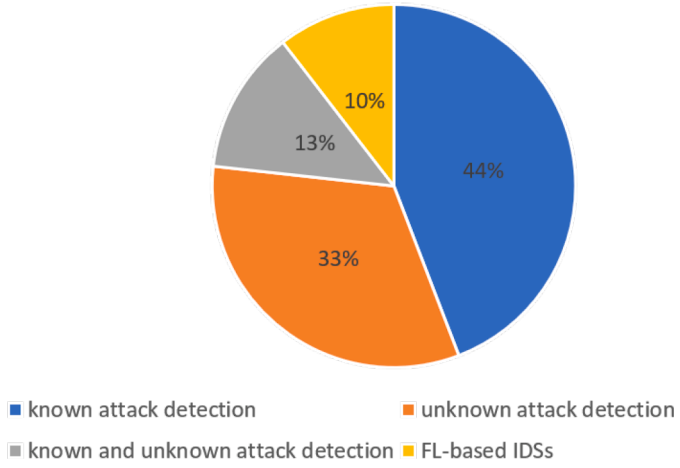


Fig. 4. Distribution of collected papers by category.

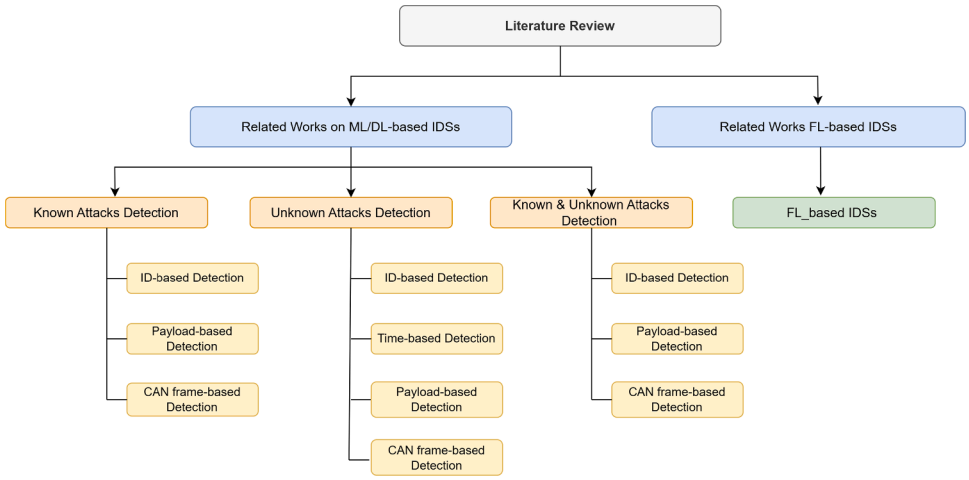


Fig. 5. Categories of reviewed literature.

Anomaly-based IDSs are further categorised into statistical, rule-based, ML, and physical fingerprinting approaches [16]. However, this survey specifically focuses on reviewing research that applies ML and DL techniques to in-vehicle IDS design. This focus arises from the widespread use of ML and DL-based IDSs to process large volumes of CAN traffic data. These approaches efficiently extract and pre-process raw CAN data, which is critical as vehicle manufacturers often do not provide detailed specifications for decoding these raw data [16]. This section is divided into three categories: known attack detection, unknown attack detection, and work capable of detecting both known and unknown attacks.

5.3. Known attacks detection

As mentioned in Section 4.1, there are 38 papers on IDSs focusing on known attack detection. In this section, we analyse these papers and discuss the existing methodologies used to detect or classify known threats in-vehicle networks. Detection of known attacks typically relies on supervised learning, which requires labelled data. This section is divided into four subsections, with three focusing on the features used to construct the models, specifically ID based detection, payload based detection, and CAN frame based detection, and a concluding subsection providing a comparative discussion. Each subsection examines different approaches for identifying malicious activities, emphasizing their strengths and limitations. Fig. 6 illustrates previous work on detecting known attacks, showing that most studies utilised a DL approach and used CAN frames as input features.

5.3.1. ID-based detection

Attacks such as injecting or deleting frames alter certain properties of message ID sequences compared to normal messages. This section presents research where the authors utilised these properties and used CAN IDs solely as an input feature to develop IDSs for detecting known attacks.

Song et al. [54] utilised the sequential behaviour of CAN data to identify message injection attacks. During these attacks, frequent frame injections resulted in distinct ID pattern changes, which were leveraged for detection. The authors relied solely on the bit-wise CAN ID sequence, which was processed directly as input, without requiring any further feature engineering. They introduced a deep convolutional neural network (DCNN) model by simplifying the Inception-ResNet architecture, reducing unnecessary complexity to achieve an optimised input size of ($29 \times 29 \times 1$) and a binary classification output.

Refat et al. [55] transformed sliding windows of CAN IDs into graph representations to analyse in-vehicle network traffic. They extracted seven structural features from each graph, which were then used as inputs to train two traditional ML algorithms: k-nearest neighbours (KNN) and support vector machine (SVM) models. Experimental findings revealed that graph-based features outperformed the traditional CAN bus features.

Nandam et al. [56] employed a Long Short-Term Memory (LSTM) model that leverages the CAN ID of incoming messages to detect potential DoS attacks. A sequence of previous messages is stored and combined with the current message to form the input, enabling the model to predict and detect DoS attacks effectively.

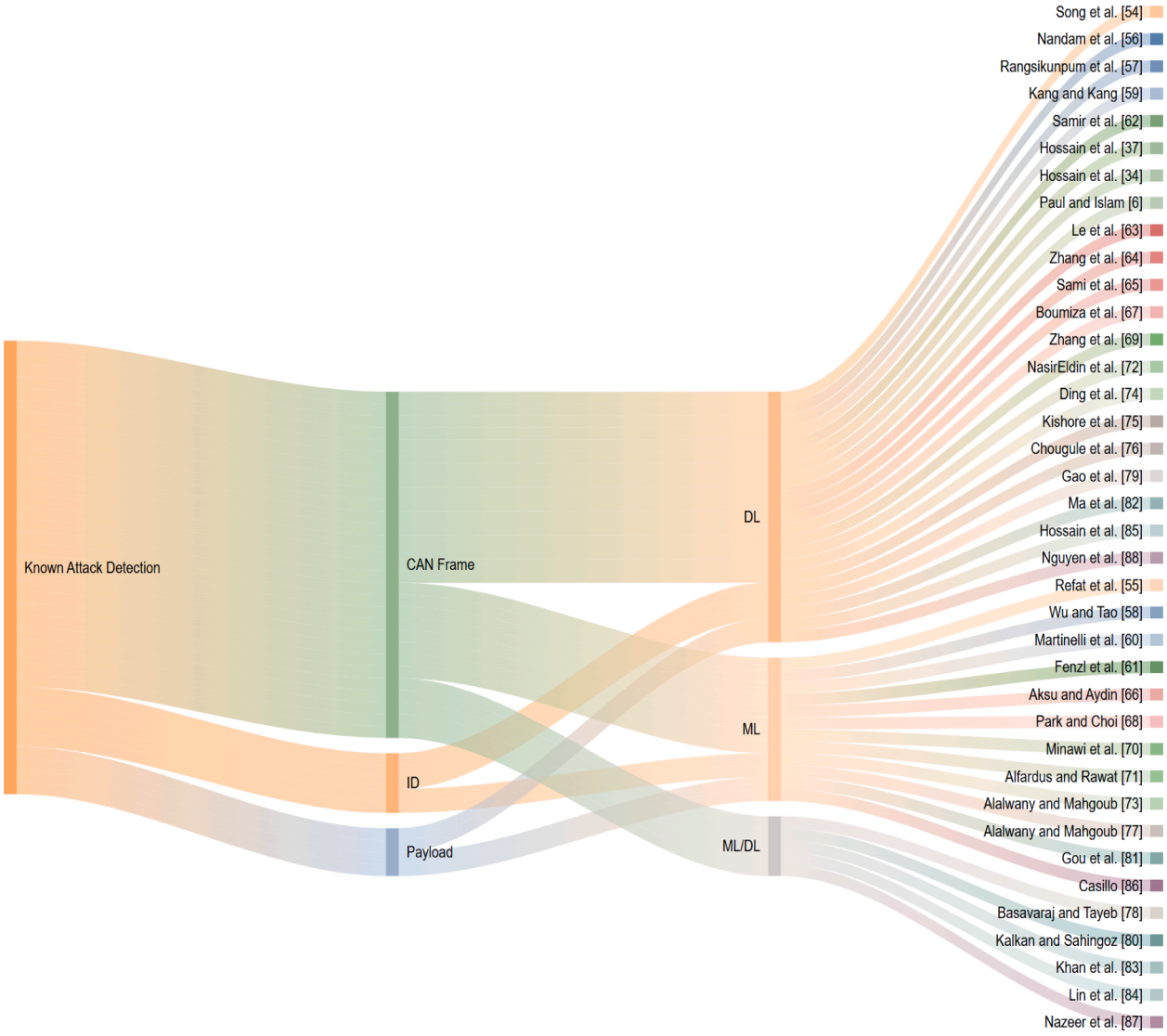


Fig. 6. Related work on known attack detection.

Rangsikunpum et al. [57] introduced a binarised neural network (BNN)-based IDS for detecting CAN bus attacks. Their main goal was to implement the model on a low-cost Field Programmable Gate Array (FPGA) device, optimising for low power consumption, minimal execution time, and high accuracy. By leveraging a 1-bit BNN model, the implementation is resource-efficient, allowing deployment on cost-effective FPGA devices with reduced power requirements. Moreover, the IDS employs a two-stage architecture: the first stage identifies the presence of an attack, and the second stage, triggered only upon detecting an attack, performs detailed attack classification.

Wu and Tao [58] proposed a model based on ensemble learning using a Stacking integration approach. The method incorporates a meta-classifier composed of DTs, Extra Trees (ET), and extreme gradient boosting (XGBoost). Final classification predictions are made by linearly combining input features and weights through a SoftMax meta-learner. Ensemble learning in this approach utilises the prediction results as new features, along with the true labels, to train the meta-learner.

Although ID-based detection methods for known attacks achieve lightweight models by reducing input dimensionality and relying solely on the CAN ID, this simplification limits their ability to detect payload manipulation attacks. Moreover, their effectiveness is typically limited to the specific attack types they were trained on, rendering them less robust against unknown or novel attacks.

5.3.2. *Payload-based detection*

Some attacks manipulate CAN payload values according to their nature, causing changes in payload sequence patterns while retaining valid CAN IDs, as is the case with spoofing attacks. In this section, we review IDSs that utilise this property and use the CAN payload as an input feature to detect known attacks.

Kang and Kang [59] introduced a deep neural network (DNN) model to defend against malicious attacks. The authors utilised the 8-byte CAN payload to extract features, employing mode and value information to reduce the feature dimensionality. The DNN model was initialised using weights from a separate Deep Belief Network. A template-matching approach was subsequently used to compare training samples with incoming CAN packets to identify malicious messages. Although the proposed model demonstrated improved detection performance, its dependence on mode and value information extracted from CAN payloads presents significant challenges, particularly in the absence of the DBC file.

Similarly, Martinelli et al. [60] utilised the eight CAN payload features to assess whether these features can effectively discriminate between attacks and normal messages. To address this question, they employed four fuzzy classification algorithms to detect and categorise four CAN bus attack types. The algorithms comprise two variants of fuzzy-rough KNN, the discernibility-based classifier, and a fuzzy unordered rule induction method. The classification analysis was performed using

the Weka3 tool. Experimental results revealed that feature vector i is highly effective in accurately differentiating between injected and normal messages.

Fenzl et al. [61] employed DTs trained using genetic programming (GP) to identify malicious activity within the CAN network. Their approach focuses solely on message payloads, with models trained individually for each CAN ID within the CAN bus training data. The authors compared their method with artificial neural networks (ANNs). The experimental findings revealed that, for most intrusions, the accuracy of the ANN was slightly higher, and the ANN had a significantly lower training time; however, the proposed GP method demonstrated significantly improved detection time.

Samir et al. [62] investigated two DL-based IDSs: one leveraging LSTM and the other utilising a one-dimensional convolutional neural network (CNN). These supervised learning algorithms serve as classifiers capable of categorising attacks into different types. The authors employed two public datasets as well as a manually created dataset, developed using the ICSim simulation tool, to incorporate a wider range of scenarios and attack types. Experiments demonstrate that the LSTM-based IDS performs better than the CNN-based IDS, benefiting from its capability to capture temporal patterns for detecting a range of CAN bus attacks.

Payload-based detection approaches for known attacks utilise the data payload to identify payload manipulation. However, they often overlook manipulations in the CAN ID field, making them ineffective against spoofing or ID-based attacks. Like ID-based methods, their effectiveness is typically restricted to the specific attack types they were trained on, rendering them less robust against unknown or novel attacks.

5.3.3. CAN frame-based detection

Rather than using only CAN IDs or CAN payload as features, IDSs in existing studies have integrated multiple features to identify pattern variations in CAN data sequences. This approach enables the detection of both alterations in CAN IDs and manipulations of the payload. This section reviews IDSs that use CAN IDs and payload as input features to detect known attacks. Some studies also incorporate the DLC feature or time intervals between successive CAN IDs, in combination with CAN ID and payload.

Hossain et al. [37] proposed an LSTM-based IDS that uses CAN ID, DLC, and payload features to identify both point as well as contextual anomalies. The IDS is trained on both benign and attack data, enabling it to distinguish between normal and malicious messages, as well as identify specific types of attacks. CAN messages were collected from a real Toyota hybrid car, and three attack scenarios were generated. The proposed LSTM-based approach demonstrated improved performance over the survival analysis method by achieving a higher detection rate.

Following their earlier research, Hossain et al. [34] introduced a 1D CNN model as an alternative to their earlier LSTM-based approach. They collected normal datasets from three vehicles: Toyota, Subaru, and Suzuki, and generated attack scenarios by injecting anomalous frames. The model demonstrated strong detection performance across all attack types. However, they identified the fuzzy attack as the most critical, as its similarity to legitimate CAN traffic makes it particularly difficult to detect.

Similarly, Paul and Islam [6] proposed an ANN model trained on benign and malicious samples from DoS and fuzzy datasets to detect unauthorised messages on the CAN bus. The model demonstrated a high detection accuracy in distinguishing between legitimate and anomalous messages, achieving nearly negligible rates of false positives (FPs) and false negatives (FNs).

Le et al. [63] designed an IDS for multiclass classification that integrates autoencoder (AE) models with a time-embedded transformer. The AE-based packet-level extraction model captures a compressed representation of each CAN frame within a sequence, while the transformer, enhanced with timestamp encoding, serves as the sequence-level feature extractor.

Zhang et al. [64] introduced a Binarized CNN (BCNN)-based IDS, designed to leverage the temporal and spatial characteristics of CAN messages. The proposed IDS consists of an input generator and a BCNN model. The input generator converts CAN messages from feature vectors into image form, enabling the BCNN model to capture their temporal and spatial features. The second component employs the BCNN model to process the output images from the input generator. Experimental results demonstrated that the BCNN model is four times faster and requires less memory compared to a 32-bit CNN-based IDS.

Sami et al. [65] introduced the Network Embedded System Laboratory's IDS (NESLIDS), which employs a supervised DL algorithm based on a DNN. NESLIDS is designed as an anomaly detection system to identify three known attacks.

Aksu and Aydin [66] introduced a meta-heuristic algorithm, the Modified Genetic Algorithm (MGA), to select a subset of features by removing irrelevant ones, thereby improving classification performance and reducing dimensionality. They evaluated the effectiveness of the feature selection process using five classifiers: Support Vector Classifier (SVC), k-Nearest Neighbors Classifier (KNC), Decision Tree Classifier (DTC), Logistic Regression Classifier (LRC), and Linear Discriminant Analysis Classifier (LDAC).

Boumiza et al. [67] developed a CAN bus IDS based on a Multi-Layer Perceptron (MLP) neural network. It initially segments the data using the CAN packet ID field and applies the K-means clustering algorithm to create subclusters. It then extracts mode and frequency features from each subcluster to train the neural network. The proposed IDS processes each CAN ID separately and combines their individual results to generate a final score, triggering an alert when an attack is detected.

Park and Choi [68] proposed a multi-labeled hierarchical classification (MLHC) IDS to detect message injection attacks. MLHC identifies and categorises attacks based on previously labelled attack data. The authors assessed the method's performance using four ML algorithms: DT, SGD, kNN, and RF. Experimental findings indicated that the RF algorithm achieved high accuracy, while the DT algorithm provided efficient detection speed.

Zhang et al. [69] proposed a Convolutional Encoder Network (CEN)-based IDS for detecting intrusions in CAN networks. The architecture combines an encoder for dimensionality reduction, a CNN to increase model depth, and Inception ResNet to accelerate the training process. Additionally, the authors introduced a Feature-based Sliding Window method to extract features from both the CAN Data Field and CAN IDs. Experimental results highlight the method's effectiveness in improving detection performance.

Minawi et al. [70] introduced an ML-based IDS comprising three layers: the CAN Message Input Layer, the Threat Detection Layer, and the Alert Layer. The Threat Detection Layer utilises ML algorithms such as Random Forest (RF), Random Tree (RT), Naive Bayes (NB), and Stochastic Gradient Descent (SGD) with hinge loss to detect different types of attacks. Additionally, this layer is designed with multiple modules, each tailored to detect specific types of attacks.

similarly, Alfardus and Rawat [71] used the same proposed IDS in [70] but with four different ML algorithms, including RF, KNN, SVM, and Multilayer Perceptron (MLP), to identify malicious activity on the CAN bus network.

NasirEldin et al. [72] designed an IDS incorporating an attention mechanism for intrusion detection. The model consists of an attention layer that prioritises the most significant features by computing attention scores between inputs and the target, followed by a self-attention layer to identify relationships between data elements. Experimental findings indicate that the proposed model delivered better performance than baseline models, such as the LSTM.

Alalwany and Mahgoub [73] introduced an ML-based IDS using supervised ML models, including RF, DT, Gaussian Naïve Bayes (GaussianNB), Logistic Regression (LR), AdaBoost, KNN, XGBoost, and Gradient Boosting. To further enhance attack detection accuracy, the authors employed three ensemble methods: voting, stacking, and bagging

to combine all supervised models. This approach leverages the diverse strengths of individual models, allowing them to work together effectively in the classification process. Compared to individual models, the ensemble classifiers outperformed the supervised classifiers by enhancing their effectiveness through the use of diverse learning mechanisms to support one another.

Ding et al. [74] developed a Bidirectional LSTM (Bi-LSTM) IDS with a sliding window strategy. A two-dimensional input data sample set was constructed using the sliding window, and the Bi-LSTM network was trained on these features to learn a classifier for intrusion detection. The experimental results demonstrate that the proposed model achieves superior performance compared to other network models, except when detecting DoS attacks. Similarly, Kishore et al. [75] proposed a Bi-LSTM that analyses input data ibidirectionally to identify unusual behaviour within the CAN bus network.

Chougule et al. [76] proposed HybridSecNet, a hybrid two-stage LSTM-CNN IDS consisting of two classification phases: the first stage uses an LSTM to classify input as either benign or malicious. If an attack is identified in the first stage, a second stage is activated, which uses a CNN-based multiclass classifier to determine and classify the specific type of attack.

Moreover, Alalwany and Mahgoub [77] proposed an in-vehicle IDS that enhances the accuracy of detecting and classifying CAN bus attacks by integrating ensemble techniques with the Kappa Architecture. While the Kappa Architecture provides capability for real-time detection, the ensemble approach enhances detection accuracy by combining classifiers such as RF, DT, and XGBoost. The study demonstrated that combining the strengths of multiple models through ensemble methods significantly improved detection accuracy and system robustness.

Basavaraj and Tayeb [78] designed a lightweight DNN-based model for detecting and classifying CAN bus attacks. The proposed model outperformed baseline models, including RF, DTs, and the kNN algorithm.

Gao et al. [79] proposed a CNN and Bi-LSTM model with multi-head attention for attack detection and classification. The CNN module enhances feature extraction, the Bi-LSTM module captures sequential features and relationships, and the multi-head attention module identifies further correlations between features.

Kalkan and Sahingoz [80] applied several ML algorithms, including RF, ANN, NB, LR, bagging, and ADA boosting. Experimental results demonstrated that tree-based and ensemble learning algorithms achieved superior performance. However, the authors did not specify the features used for training, leading to the assumption that all features were included.

Gou et al. [81] proposed an adaptive tree-based ensemble network (ATBEN) aimed at improving IoV security. ATBEN leverages a variety of ML models, including XGBoost, LightGBM, RF, and ET, as base estimators, stacking them into layers within the network. The cascading connections between layers facilitate precise and efficient multiclass classification. The authors validated the effectiveness of the proposed IDS by assessing its performance against a range of cyberattacks targeting both in-vehicle systems and external networks within the IoV.

Ma et al. [82] proposed a GRU-based IDS and, to improve efficiency, applied a low-complexity feature extraction algorithm to derive features from CAN frames. Experimental results showed that the GRU-based IDS achieved near real-time performance and outperformed baseline models in detection accuracy.

Khan et al. [83] proposed DivaCAN, an IDS that combines DL models with conventional ML methods through an ensemble of base classifiers, including DNN, MLP, light gradient-boosting machines, ET, RF, Bagging, and KNN. To improve detection performance, a meta-classifier adaptively integrates the outputs of base classifiers, assigning weights based on their performances and correlations. This work addresses the trade-off between FPs and time complexity in CAN bus IDS.

Lin et al. [84] developed a CNN-based IDS that utilises the VGG16 classifier to capture attack behaviour characteristics and classify threats. Feature vectors were transformed into feature images, which were then input into the VGG16 model for accurate categorisation of cyber threats in-vehicle networks. To ensure high precision in predicting the stability of network intrusion detection, the approach combines the VGG16 model with the XGBoost ensemble learning algorithm, enabling effective analysis of suspicious network traffic.

Hossain et al. [85] proposed an LSTM-based IDS for detecting attacks in-vehicle network. The CAN message data was collected using the Vehicle Spy 3 tool. To evaluate the IDS, the authors employed both binary and multi-class classification approaches, utilizing vanilla LSTM and stacked LSTM models. Since the dataset originally contained no attacks, the authors simulated various CAN bus attacks, including DoS, Fuzzing, and Spoofing, on Toyota Hybrid car using a Python-based program.

Casillo [86] proposed an embedded IDS for automotive systems by adopting Bayesian Networks for the rapid identification of malicious messages on the CAN bus. The CAN bus dataset was generated by simulating vehicle driving for approximately 24 h on a city track within the CARLA environment. During the simulation, the vehicle was subjected to attacks to replicate potential intrusion scenarios based on specific use cases.

Nazeer et al. [87] introduced a hybrid approach, DeepXG, which utilised XGBoost and DNN models to detect and classify attacks on the CAN bus. The XGBoost is trained on the dataset to extract critical features and reduce computational complexity, while the DNN leverages these learned representations to detect anomalies and intrusions.

Nguyen et al. [88] introduced a Transformer attention network-based IDS designed to analyse a single message. The proposed IDS includes two models: one that processes individual messages and another leveraging sequential CAN IDs. The initial model effectively identifies DoS, fuzzy, and spoofing attacks but cannot detect replay attacks due to its reliance on single-message analysis. To address this limitation, the second model was designed to identify replay attacks by incorporating sequential CAN ID information. Additionally, transfer learning is utilised to enhance the performance of models trained on limited datasets from different car types.

Table 2 summarises the related work on known attack detection methods, including the learning approach, binary or multi-class classification, dataset used, detectable attacks, and the employed algorithm. In Table 2, we assume that papers that do not explicitly state the input features used are referring to CAN frame features.

CAN frame-based methods leverage the entire CAN frame, including ID and payload, enabling the detection of both ID and payload manipulation attacks. While all known attack detection methods (ID-based, payload-based, and CAN frame-based) demonstrate high accuracy and low false alarm rates (FAR), their effectiveness relies heavily on well-labelled and balanced datasets. However, obtaining such labelled data remains a significant challenge for researchers [53]. Moreover, the labeling process is often time-consuming, prone to errors, and tedious [97]. Additionally, the main limitation of these studies is that none of the models are capable of detecting new attacks or deviations from the known attacks they were trained on. As attackers continuously develop new and previously unseen methods to evade detection, supervised learning-based models often cannot detect attack patterns that were not included in the training data [98]. This limitation can pose significant security risks. To address this, the next section discusses approaches designed to identify new, unknown threats.

5.3.4. Comparative discussion

For ID-based approaches, Song et al. [54] demonstrate that a DCNN-based IDS outperforms LSTM, ANN, SVM, kNN, NB, and decision tree models, underscoring its effectiveness in modelling the sequential and temporal characteristics of CAN traffic. Similarly, Rangsikunpum et al. [57] report detection rates above 99% using a BNN-based IDS, with state-of-the-art inference efficiency. In comparison, Refat et al. [55]

Table 2
Summary of related work on known attack detection methods.

Reference	Year	ML \ DL	Category	Classification Type	Dataset	ID	Payload	Attack Types	Algorithm
ID-Based Attack Detection									
[54]	2020	DL	Supervised	Binary	Car Hacking [89]	✓		Message Injection	DCNN
[55]	2022	ML	Supervised	Binary	Car Hacking [89]	✓		DoS, Fuzzy, RPM Spoofing	SVM, KNN
[56]	2022	DL	Supervised	Binary	car-hacking [54]	✓		DoS	LSTM
[57]	2024	DL	Supervised	Binary \ Multi-class	Car Hacking [89]	✓		(Gear, RPM) Spoofing, DoS, Fuzzy	BNN
[58]	2024	ML	Supervised	Binary	Car Hacking [89]	✓		(Gear, RPM) Spoofing, DoS, Fuzzy	Ensemble model (DT, ET , XGBoost)
Payload-Based Attack Detection									
[59]	2016	DL	Supervised	Binary	Simulation		✓	Injection Attacks	DNN
[60]	2017	ML	Supervised	Binary	Car Hacking [89]		✓	DoS, Fuzzy,(Gear, RPM) Spoofing	KNN
[61]	2021	ML	Supervised	Binary	Car Hacking [89], Tesla Model X data, Renault Zoe electric car data		✓	(RPM, Gear) Spoofing	DT, GP
[62]	2024	DL	Supervised	Multi-class	Car Hacking[89], OTIDS [32], Own		✓	DoS, Fuzzy, Spoofing, Replay	CNN, LSTM
CAN Frame -Based Attack Detection									
[37]	2020	DL	Supervised	Binary \					
Multi-class	Own	✓	✓	DoS, Fuzzy, Spoofing	LSTM				
[34]	2020	DL	Supervised	Binary \ Multi-class	Own	✓	✓	DoS, Fuzzy, Spoofing	CNN
[6]	2021	DL	Supervised	Binary	OTIDS [32]	✓	✓	DoS, Fuzzy	ANN
[63]	2024	DL	Supervised	Multi-class	car-hacking [54], ROAD [90]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Fabrication, Masquerade	AE, Time-embedded Transformer
[64]	2024	DL	Supervised	Binary	Own	✓	✓	Replay, Spoofing	BCNN
[65]	2020	DL	Supervised	Binary	OTIDS [32], ML350 [91]	✓	✓	DoS, Fuzzy, Impersonation	DNN
[66]	2022	ML	Supervised	Binary \ Multi-class	car-hacking [54]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	SVC, LRC, DTC, KNC, LDAC
[67]	2019	DL	Supervised	Binary	Dataset [92]	✓	✓	Frequency modification, Data-content modification	MLP
[68]	2020	ML	Supervised	Binary \ Multi-class	Survival Analysis Dataset [93]	✓	✓	Fuzzy, Flooding, Malfunction	SGD, kNN, DT, RF
[69]	2020	DL	Supervised	Multi-class	Car Hacking [89], car-hacking [54]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	CEN
[70]	2020	ML	Supervised	Binary	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	RT, RF, SGD, NB
[71]	2021	ML	Supervised	Binary	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	KNN, RF, SVM, MLP
[72]	2021	DL	Supervised	Binary	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	Attention-based model
[73]	2022	ML	Supervised	Binary	Car Hacking: Attack & Defence Challenge 2020 [94]	✓	✓	Flooding, Spoofing, Replay, Fuzzy	LR, GaussianNB, k-NN, RF, Gradient Boosting, AdaBoost, DT, XGBoost
[74]	2022	DL	Supervised	Binary	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	Bi-LSTM
[75]	2024	DL	Supervised	Binary	Car Hacking: Attack & Defence Challenge 2020 [94]	✓	✓	Flooding, Spoofing, Replay, Fuzzy	Bi-LSTM
[76]	2024	DL	Supervised	Binary \ Multi-class	Car Hacking [89]	✓	✓	DoS, Fuzzy, (Gear, RPM) Spoofing	LSTM-CNN
[77]	2024	ML	Supervised	Multi-class	Car Hacking: Attack & Defence Challenge 2020 [94]	✓	✓	DoS, Spoofing, Replay, Fuzzy	RF, DT, XGBoost
[78]	2022	DL \ ML	Supervised	Multi-class	CAN dataset [95]	✓	✓	Reconnaissance, DoS, Fuzzy	DNN
[79]	2023	DL	Supervised	Multi-class	Car Hacking [89]	✓	✓	DoS, (Gear, RPM Spoofing, Fuzzy	CNN, bi_LSTM
[80]	2020	ML \ DL	Supervised	Binary	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	RF, bagging, ADA boosting, NB, LR, ANN
[81]	2023	ML	Supervised	Multi-class	car-hacking [54]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	XGBoost, LightGBM, RF, ET
[82]	2022	DL	Supervised	Binary	Car Hacking [89]	✓	✓	DoS, Spoofing, Fuzzy	GRU
[83]	2024	ML \ DL	Supervised	Multi-class	OTIDS [32]	✓	✓	DoS, Fuzzy, Impersonation	DNN, MLP, light gradient-boosting machine, ET, RF, Bagging, KNN
[84]	2022	DL \ ML	Supervised	Multi-class	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	VGG16, XGBoost
[85]	2020	DL	Supervised	Binary \ Multi-class	Own	✓	✓	DoS, Fuzzy, Spoofing	LSTM
[86]	2019	ML	Supervised	Binary	Simulation	✓	✓	Turn right, Turn left, Brake	Bayesian Network
[87]	2024	ML \ DL	Supervised	Multi-class	Own	✓	✓	Flooding, Replay, Spoofing	XGBoost, DNN
[88]	2023	DL	Supervised	Binary \ Multi-class	Car Hacking [89], IVN [96], Survival Analysis Dataset [93]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Replay, Malfunction	Transformer

Table 3
Comparison of resource requirements for existing known attack detection methods.

	Reference	Latency (ms)	Model Size	Trainable Parameters
	[54]	5	–	1.76 Million
ID-based	[57]	0.26	4.85 Mb	–
	[59]	2–5	–	–
Payload-based	[61]	–	–	1101
	[63]	0.24	–	259,000
	[64]	0.6	7.49 Mb	–
	[68]	0.34 - 1.89	–	–
	[82]	1.37	–	–
CAN frame-based	[88]	0.091	–	–

achieve accuracies of 97.92% and 97.99% using SVM and kNN on graph-based CAN representations, while Wu and Tao [58] report up to 99% accuracy for several attack types but reduced performance for fuzzy attacks. Nandam et al. [56] provide only limited evaluation metrics, with accuracy of around 80%. Regarding payload-based approaches, Kang and Kang [59] report that their method achieves around 98% average detection accuracy while maintaining real-time response with modest computational complexity. Similarly, Martinelli et al. [60] report that Fuzzy RoughNN achieves perfect precision and recall for gear and rpm attacks and high precision for DoS and fuzzy attacks, though with lower recall. Samir et al. [62] further report strong LSTM performance, with accuracy ranging from 90.18% to 99.21% across multiple datasets, alongside high precision and low false positive and false negative rates. Regarding CAN frame-based approaches, many studies report detection rates above 99%, largely attributable to training on datasets that include attack samples. However, not all methods maintain such performance across attack types. For example, the BCNN-based IDS proposed by Zhang et al. [64] achieves lower accuracies of 96.82% under replay attacks and 94.19% under spoofing attacks. Similarly, Khan et al. [83] introduce DivaCAN, which attains 94.93% precision, 94.98% recall, and an F1-score of 94.97%. Furthermore, Alalwany and Mahgoub [73,77] demonstrate that stacking ensemble techniques can consistently outperform individual supervised models. In Table 3, we compare key metrics, including latency, model size, and the number of trainable parameters, which are essential for deploying known attack detection methods in resource-constrained, real-time automotive environments. Only a limited number of existing studies report these metrics. For example, although [54] uses only the CAN ID as input and simplifies the Inception ResNet architecture to reduce complexity, it still exhibits the longest detection time and the highest number of trainable parameters. Moreover, [88] presents an IDS that employs two DL models, while still achieving the fastest detection latency.

5.4. Unknown attacks detection

As mentioned in Section 4.1 there are 28 papers on IDSs focusing on unknown attack detection or anomaly detection. In this section, we analyse these papers and discuss the existing methodologies used to detect new, unknown threats in-vehicle networks. Detection of unknown attacks typically relies on unsupervised learning, where models are trained solely on normal data and identify deviations from established patterns of normal traffic as potential anomalies. Consequently, such models are particularly effective in detecting previously unseen attacks [99]. The section is organised into five subsections: four based on the features used to build the model, namely ID based detection, time based detection, payload based detection, and CAN frame based detection, and a last subsection presenting a comparative discussion. Each subsection examines different approaches for identifying malicious activities, emphasizing their strengths and limitations. Fig. 7 illustrates previous work on detecting unknown attacks. As depicted in the figure, similar to known attack detection, most studies on unknown attack detection utilised a DL approach and used CAN frames as input features.

5.4.1. ID-based detection

This section reviews research where authors used only CAN IDs as the input feature to develop IDSs for detecting new, unknown attacks.

Avatefipour et al. [100] developed an IDS that integrates a modified One-Class Support Vector Machine (OCSVM) with the Modified Bat Algorithm (MBA) for anomaly detection. The model was built using normal traffic, which exhibits recurring patterns in CAN IDs under normal conditions. Any deviation from this normal traffic, such as increased message occurrence frequency or message flooding, is detected as malicious activity. The authors compared the proposed model with baseline Isolation Forest and classical OCSVM models, finding that the MBA-OCSVM achieved the highest true positive rate and the lowest FAR compared to both alternatives.

Rajapaksha et al. [101] proposed CAN-CID, a context-aware IDS aimed at addressing the computational inefficiency of N-gram-based models while detecting a wide range of cyberattacks on the CAN bus. CAN-CID utilises an ensemble approach that combines a Gated Recurrent Unit (GRU) network and a time-based model. The single-layer GRU network detects anomalous ID sequences and minimises detection latency, while the time-based model identifies anomalies using time-based thresholds. The anomaly-to-total-ID ratio within an observation window is then used to classify the window as either anomalous or benign. This study highlights the effectiveness of ensemble models in identifying various types of attacks targeting the CAN bus.

Khandelwal and Shreejith [102] presented a convolutional autoencoder (CAE) model designed to detect zero-day attacks, using only benign CAN messages for training. Leveraging Vitis-AI tools, they quantised the model to optimise performance on resource-constrained platforms. The proposed IDS demonstrates superior classification accuracy on multiple unseen attacks, achieving a 1.3x improvement in processing latency and an approximately 2x reduction in power usage compared to existing IDSs.

Guidry et al. [103] employed a One-Class Support Vector Machine (OC-SVM) to identify abnormal traffic on the CAN bus. Instead of utilising raw CAN bus data, three distinct features were extracted for each unique CAN ID: the average frequency of appearance of a CAN ID, the average time interval between consecutive appearances of a CAN ID, and the standard deviation of transmission times for CAN IDs. These features were selected because they rely on the temporal and behavioural characteristics of message transmissions rather than the data content within the messages. The model was trained on CAN bus data collected under normal operating conditions, making it well-suited for detecting unknown attacks in vehicular networks.

These methods can detect attacks targeting the CAN ID field, making them suitable for identifying previously unseen ID-level attacks. However, they are ineffective against intrusions that manipulate only the payload, as they typically overlook payload-level attacks.

5.4.2. Time-based detection

This section reviews studies that use only the timestamp as the input feature for IDSs. Sharmin and Mansor [104] is the only study that utilises the timestamp as the sole feature to detect unknown attacks. They proposed an Isolation Forest (iForest) anomaly detection algorithm, which detects message injection attacks by analysing time intervals between consecutive messages with the same CAN ID. The model was trained on normal CAN traffic and tested on attack datasets. Analysis of normal traffic revealed that ECUs transmit messages at fixed intervals, with each CAN ID following a unique timing pattern. Message injection and replay attacks disrupt these intervals, making them detectable. In contrast, studies that combine the timestamp with additional features are discussed in the payload-based detection or CAN frame detection sections. Although this approach is lightweight and efficient, with linear time

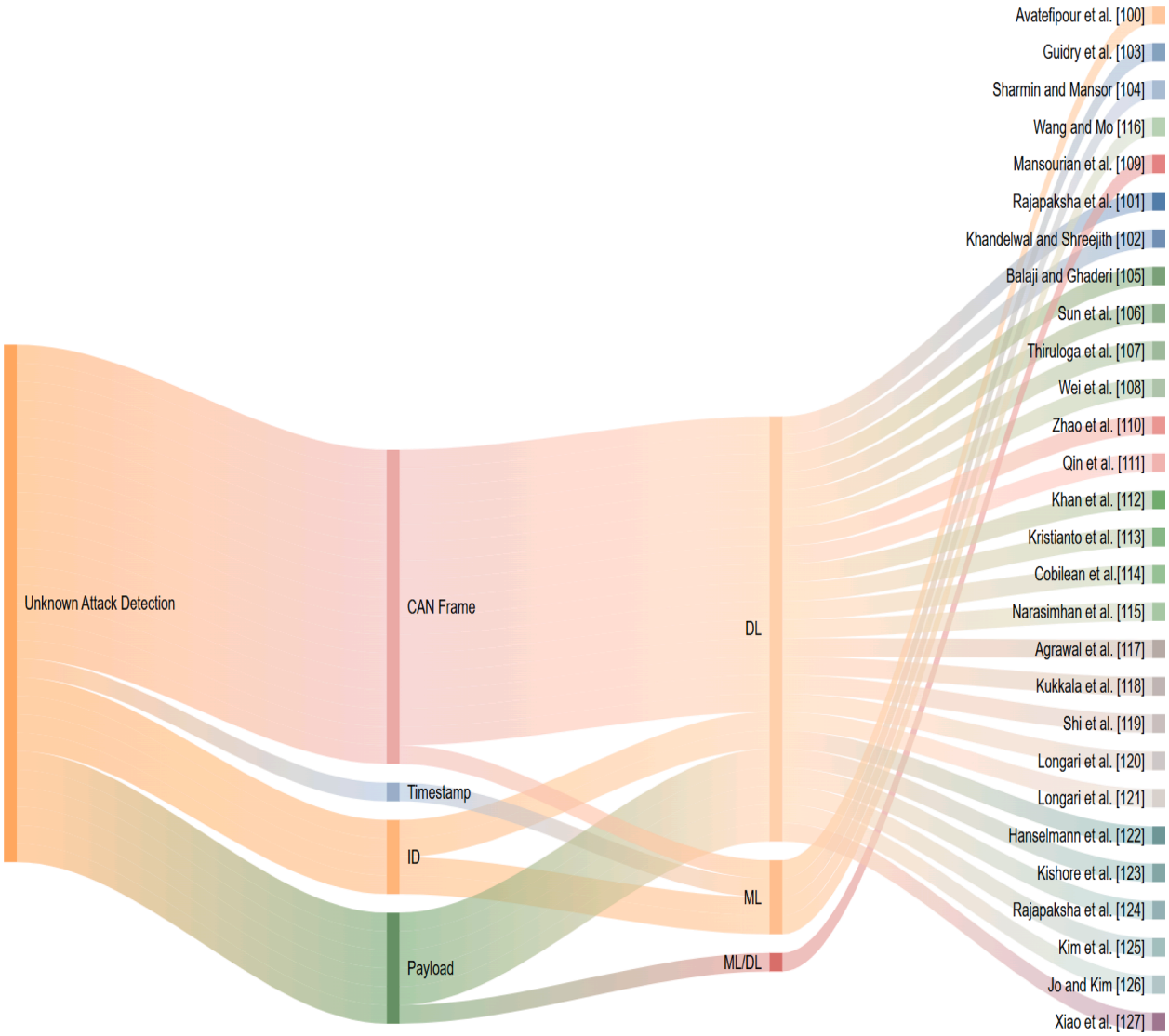


Fig. 7. Related work on unknown attack detection.

complexity and low resource requirements, it can only detect timing-based anomalies and fails to identify attacks that do not disrupt message intervals.

5.4.3. Payload-based detection

This section discusses IDSs that use the 8-byte CAN payload as an input feature to identify new, unknown attacks.

Balaji and Ghaderi [105] proposed NeuroCAN, a contextual anomaly detection model that consists of an embedding layer and LSTM to learn the spatio-temporal correlations among CAN payload values. The embedding layer linearly transforms the input from each CAN ID, applies a sigmoid activation, and aggregates the outputs across all IDs. The output is then passed through an LSTM and a final output layer, forming a prediction-based anomaly detection model. Incorporating payload values from other IDs as context allows the model to learn inter-ID correlations. However, training a separate model for each CAN ID leads to substantial memory usage and computational overhead.

Sun et al. [106] designed a CNN-LSTM-based IDS incorporating an attention mechanism. To extract abstract features, the model employs one-dimensional convolution, followed by a bi-directional LSTM to capture time dependencies. The bit flip rate was employed to detect continuous fields from the 64-bit payload, resulting in a 41-bit smaller signal, which is more efficient than directly predicting the full 64-bit. Experiments demonstrated that this approach reduces data dimensionality and improves model training efficiency. The pre-processed data was given to the neural network model to predict the output signal and determine whether the received signal was abnormal. The proposed model improved attack detection accuracy by 2.5% compared to related research.

Thiruloga et al. [107] proposed TENET, an anomaly detection framework built on temporal convolutional neural attention (TCNA) networks. TENET receives a sequence of signal values from a message and employs CNNs to predict the signal values of the subsequent message instance by learning the normal data's underlying probability distribution. A DT-based classifier was then employed as the attack detector. Experimental results showed that TENET achieved a 3.32% improvement in detection accuracy, a 32.7% reduction in the false negative rate, and 94.62% fewer model parameters compared to a baseline model. However, the model processed data ID-wise, training separate models for each ID, which limits its ability to detect anomalies, such as collective anomalies, that arise from interactions between different CAN IDs.

Wei et al. [108] introduced AMAEID, a multi-layer denoising autoencoder model. The model takes only the 8-byte payload of the CAN

message as input. It first transforms the raw hexadecimal payload into binary format, then applies a multi-layer denoising autoencoder to extract deeper hidden features that represent the underlying characteristics of the message. Additionally, AMAEID utilises an attention mechanism and a fully connected layer to classify messages as normal or abnormal. Experimental results demonstrate that AMAEID surpasses traditional ML algorithms like DT, KNN, and LinearSVC. However, the model was trained and tested using only two CAN IDs.

Mansourian et al. [109] introduced an anomaly-based IDS that comprises three modules: an LSTM model, a prediction error calculator, and a Gaussian Naïve Bayes (GNB) classifier. The LSTM is trained on benign CAN messages to learn the typical sequential behaviour of each ECU. After training, it predicts the next payload based on previous data and evaluates it against the actual received value. In the event of an attack, the trained LSTM network generates inaccurate predictions, causing a noticeable increase in prediction error. The GNB classifier then classifies messages as either normal or an attack based on these prediction errors.

Zhao et al. [110] introduced the Same Origin Method Execution (SOME) attack, which mimics the period, clock skew, and voltage of normal messages, making detection by existing IDSs challenging. To address this, they developed a GAN-based IDS, named GVIDS, which employs one-hot encoding to represent data and converts data frames into CAN images. This approach is effective as attacks either directly alter frame data or disrupt frame sequences, indirectly modifying all the consecutive data fields. Results from experiments on two real vehicles demonstrate that GVIDS successfully detects SOME attacks as well as other existing attack types.

By focusing on the payload data, these methods are sensitive to content manipulation and capable of detecting novel attacks that deviate from normal data patterns. Nevertheless, their effectiveness is limited when attackers target the CAN ID field, which they typically ignore.

5.4.4. CAN frame-based detection

This section reviews IDSs that use the CAN frame (CAN IDs and payload) as input features to detect new, unknown attacks. Some studies also incorporate the DLC feature and/or time differences between consecutive CAN IDs, in combination with CAN ID and payload.

Qin et al. [111] proposed an LSTM-based approach for detecting anomalous behaviour, using CAN data collected from a real vehicle network that included simulated attacks such as tampering and packet injection. CAN ID and payload were converted from hexadecimal to binary representations instead of decimal, which increased the dimensionality of the features. Anomaly detection in the message stream of the CAN bus was performed for each ID separately. Experimental results showed that the proposed model achieved over 90% accuracy in detecting anomalous data.

Khan et al. [112] used a bidirectional LSTM model with an improved feature processing method to mitigate zero-day attacks. The proposed IDS is a multi-stage system, where the initial stage employs a state-based Bloom filter technique to verify the states of incoming data, while the second stage uses a bidirectional LSTM classifier to detect cyberattacks. They implemented data pre-processing techniques to enhance the IDS's scalability and performance, including feature conversion, dimensionality reduction, and normalisation. Principal Component Analysis was used to reduce feature dimensionality. The findings indicated a 19.31% increase in accuracy when the pre-processing steps were applied, compared to using unprocessed data.

Kristianto et al. [113] introduced a lightweight unsupervised IDS using a simple Recurrent Neural Network (RNN). The authors suggest deploying the IDS model at each domain gateway, leveraging the computational resources of the gateways to handle only domain-specific messages. This approach enables the gateway to be optimised for detecting malicious messages within its domain while maintaining a lightweight design. The IDS achieves up to a 94% reduction in parameters compared to existing models, significantly decreasing memory usage and energy consumption. Despite this reduction in size, the proposed models demonstrate only a slight decrease in accuracy compared to current solutions.

Cobilean et al. [114] developed a Transformer neural network-based IDS designed to predict anomalous behaviour within CAN protocol communication. The Transformer model is trained to predict the next message in the communication sequence. An anomaly is flagged when the difference between the predicted and actual messages exceeds a predefined threshold. A key advantage of this model is that it does not require labelled attack data for learning the communication sequence.

Narasimhan et al. [115] proposed an unsupervised two-stage approach that combines DL with a probabilistic model for anomaly detection. In the first stage, an autoencoder (AE) is used to extract optimal features that differentiate between normal data and attacks on the CAN bus. Unlike other autoencoder-based models that utilise the reconstructed signal for anomaly detection, this model leverages the latent space as input to a Gaussian Mixture Model (GMM). In the second stage, the GMM clusters these features into normal and attack categories. Experimental results showed that the proposed approach outperformed existing methods on various datasets. For evaluation, a real dataset from a Mercedes ML350 was used; however, as this dataset contained only four CAN IDs, the practical applicability of the model may be constrained.

Wang and Mo [116] developed a CAN bus anomaly detection model using FLXGBoost algorithm. To address the challenge posed by the large volume of traffic data messages with limited features, they introduced a newly defined feature: information entropy, which serves as an additional set of features in the CAN message data domain.

Agrawal et al. [117] proposed NovelADS, an IDS that utilises CNNs and LSTMs to detect anomalies in CAN network traffic. NovelADS captures spatio-temporal features and long-term dependencies from CAN messages. The DL models are trained on benign data, and the system classifies incoming CAN messages as benign or malicious using a reconstruction-based thresholding approach.

Kukkala et al. [118] introduced INDRA, an IDS that employs a GRU-based recurrent autoencoder to learn latent representations of normal CAN traffic and identify malicious behaviour on the CAN bus. At runtime, the trained autoencoder monitors deviations from normal behaviour to identify potential intrusions. Signal-level intrusion scores, calculated as the difference between predicted and actual signal values, are used to detect anomalous signals. Separate autoencoder models are trained for each CAN ID, enabling ID-specific anomaly detection model.

Shi et al. [119] introduced an IDS called IDS-DEC, which integrates a spatiotemporal self-encoder employing LSTM and CNN (LCAE) with an entropy-based deep embedding clustering approach. The LSTM component models the sequential nature of the data, capturing long-term dependencies in the time-series data from the CAN bus. Additionally, as network data can be represented as a multidimensional matrix with spatial structure, CNNs are employed to extract key features, thereby enhancing the accuracy and efficiency of detection. Experimental results indicate that the proposed IDS outperforms traditional ML algorithms and other deep clustering approaches.

Longari et al. [120] proposed CANnolo, an IDS that employs LSTM autoencoders for identifying anomalies on the CAN bus. CANnolo analyses CAN message streams to construct a model of normal data sequences and identifies anomalies by measuring the discrepancy between reconstructed sequences and their corresponding real sequences. The authors partitioned the dataset into groups based on CAN IDs, with each group processed independently and trained on separate models. While this approach simplifies the training process, it limits the system's ability to detect signal correlations, thereby reducing its effectiveness in identifying anomalies such as collective anomalies [16].

To enhance the architecture of CANnolo and reduce its computational demands to better meet the real-time requirements of the automotive domain, Longari et al. [121] introduced CANdito, an unsupervised IDS that leverages LSTM autoencoders for anomaly detection us-

ing signal reconstruction. CANdito regenerates the time series of CAN messages for each ID and determines anomaly scores by calculating the reconstruction error.

Hanselmann et al. [122] proposed CANet, an IDS based on an LSTM autoencoder, where a separate LSTM model was assigned to each CAN ID, and their outputs were combined into a single latent vector. Anomalies were detected by evaluating the difference between the original and reconstructed signals. The model showed high detection accuracy while maintaining low rates of false positives and false negatives across different attack types.

Similarly, Kishore et al. [123] introduced an anomaly detection technique using LSTM networks. The model surpasses traditional tree-based ML algorithms, including AdaBoost, GBoost, Bagging, XGBoost, and LGBM.

Rajapaksha et al. [124] introduced an ensemble IDS that combines a GRU network and a novel AE model called Latent AE to identify cyberattacks on the CAN bus. The GRU network analyses the CAN ID field, while Latent AE focuses on the CAN payload field to identify anomalies. To improve efficiency, Latent AE incorporates Cramér's statistic-based feature selection and a transformed CAN payload structure. By utilising a compact latent space, it overcomes the issue of high FNs in traditional AEs caused by overgeneralisation. Experimental findings reveal that the ensemble IDS enhances attack detection and addresses the limitations of the individual models.

Kim et al. [125] introduced an IDS employing multiple LSTM-Autoencoders, designed to capture distinct patterns of normal network behaviour by utilising features such as transmission intervals and changes in payload values. The IDS comprises a feature sequence extractor, LSTM-Autoencoder models, and an anomaly detection module. The time interval sequence extractor is responsible for calculating the time gaps between consecutive frames with the same ID, thereby generating a temporal sequence for each ID. Similarly, the Hamming distance sequence extractor computes the Hamming distances between the payloads of consecutive frames within ID-based streams. These feature sequences are processed by the LSTM-Autoencoders to produce reconstructed sequences. The anomaly detector evaluates the differences between the original time interval and Hamming distance sequences and their reconstructed counterparts, using these differences to determine whether the frame sequences are normal or anomalous.

Jo and Kim [126] proposed an IDS based on the Transformer architecture, which predicts the next data point based on the flow of previously input data. The IDS can detect attacks affecting both the temporal and spatial aspects of the data, as CAN data comprise temporal information recorded over time and spatial information recorded across devices. This two-dimensional data is used to train the model, achieving higher performance compared to using one-dimensional data.

Xiao et al. [127] proposed an anomaly detection IDS that employs a Convolutional LSTM Network (ConvLSTM), which accounts for both temporal and spatial correlations. The ConvLSTM model is first trained on benign CAN data, and its predictions are used to calculate the correlation coefficient with actual data. Abnormal behaviour is detected by comparing the correlation coefficients between the predicted and real data. Experimental results indicate that the ConvLSTM model maintains a stable correlation coefficient for normal data, while the coefficient for attack data declines rapidly over time. Compared to the LSTM model, the ConvLSTM model more effectively captures the underlying features of benign data, producing a more consistent correlation coefficient for attack-free states. Furthermore, the sharp drop in the correlation coefficient for attack data can facilitate the detection of unknown attacks.

Table 4 summarises the related work on unknown attack detection methods, including the learning approach, dataset used, detectable attacks, and employed algorithm. In Table 4, we assume that papers that do not explicitly state the input features used are referring to CAN frame features.

CAN frame-based methods integrate both ID and payload features, allowing for a more comprehensive anomaly detection strategy that captures a wider range of unknown attack types. Most existing anomaly-based IDSs (including ID-based, time-based, payload-based, and CAN frame-based methods) are trained exclusively on normal traffic and use binary classification to flag deviations. While it is crucial to detect new, previously unknown attacks, as attackers may introduce novel zero-day attacks that do not fit existing patterns, it is equally important to assign fine-grained labels to known attacks. Identifying the specific attack type can be highly beneficial for selecting appropriate countermeasures and conducting post-attack analysis [131]. Thus, there is a need for a comprehensive in-vehicle IDS that addresses both known attacks and new, unknown attacks while meeting deployment requirements. To address this, the next section discusses work proposed with the ability to identify and classify known attacks while also identifying new, unknown attacks.

5.4.5. Comparative discussion

CAN ID sequence-based detection has demonstrated high effectiveness. Rajapaksha et al. [101] report F1-scores above 99% for 13 attack types, although performance declines on other datasets, likely due to differences in the number of available CAN ID sequences. Similarly, Khandelwal and Shreejith [102] achieve classification accuracy exceeding 99.5% on unseen DoS, fuzzing, and spoofing attacks, outperforming state-of-the-art unsupervised learning based IDSs. Sharmin and Mansor [104] uniquely rely on timestamps for unknown attack detection, showing promise but limited to CAN IDs with fixed transmission intervals rather than event-triggered IDs. Regarding payload-based approaches, Sun et al. [106] report that their CNN LSTM-based IDS achieves an average F1-score of 0.951. In a similar vein, NeuroCAN, proposed by Balaji and Ghaderi [105], attains over 95% detection accuracy across multiple datasets, with F1-scores ranging from 0.95 to 1. Notably, the highest performance is reported by Mansourian et al. [109], whose method outperforms baseline approaches and achieves near perfect detection accuracy and F1-score. Regarding CAN frame-based approaches, several studies report consistently high detection performance. Khan et al. [112] show that their framework achieves accuracy between 98% and 99% across multiple datasets. Similarly, Kristianto et al. [113] report strong results, with an average F1-score of 0.96, accuracy of 0.98, and recall of 0.99, although performance is slightly lower for gear spoofing attacks. High and stable detection performance is also demonstrated by Shi et al. [119], whose model achieves approximately 99% accuracy, an F1-score close to 99%, and a low false alarm rate of 0.5% across various attack types. In contrast, the unsupervised LSTM-based approach proposed by Kishore et al. [123] exhibits lower performance, with an accuracy of 93.06%, precision of 0.9298, and recall of 0.9234. Overall, these results indicate that unsupervised learning approaches generally achieve lower performance than supervised methods. More recently, Transformer-based models have been explored for CAN frame level anomaly detection. For instance, Cobilean et al. [114] report a recall of 1.0 and an F1-score of 0.9873, indicating the potential of attention-based architectures in this context. Table 5 summarises latency, model size, and trainable parameters relevant to IDS deployment in resource-constrained, real-time automotive environments. It should be noted that the longest reported latencies in [117] and [120] do not specify the batch or packet size, limiting direct comparison. Otherwise, all proposed approaches exhibit low detection latency. In addition, most methods have low model sizes and a limited number of trainable parameters, except [124], which shows a substantial increase in model size compared to the others. This generally low latency and compactness can be attributed to the fact that most anomaly detection approaches are trained using only a single class, namely normal data.

5.5. Known and unknown attacks detection

To address the limitations of previous approaches and further improve the robustness and detection capability of in-vehicle IDSs, 10 papers found from the search strategy in Section 4.1 developed IDSs capa-

Table 4
Summary of related work on unknown attack detection methods.

Reference	Year	ML \ DL	Category	Dataset	ID	Payload	Attack Types	Algorithm
ID-Based Attack Detection								
[100]	2019	ML	Unsupervised	Own, Dodge [128], OTIDS [32]	✓		Injection	MBA-OCSVM
[101]	2022	DL	Unsupervised	ROAD [90], car-hacking [54], Survival Analysis Dataset [93]	✓		Fabrication, Suspension, Masquerade	GRU
[102]	2023	DL	Unsupervised	car-hacking [54]	✓		DoS, Fuzzy, (Gear, RPM) Spoofing	AE
[103]	2023	ML	Unsupervised	Own	✓		Random ID, Zero ID, Replay	OC-SVM
Time-Based Attack Detection								
[104]	2021	ML	Unsupervised	Car Hacking [89]			Injection attacks	iForest
Payload-Based Attack Detection								
[105]	2021	DL	Unsupervised	Two Public Datasets from [89]		✓	Flood, Replay, Drop, Spoofing, Fuzzy	LSTM
[106] Extraction and Translation [129]	2021	DL ✓	Unsupervised Flood, Replay, Drop, Spoofing, Fuzzy	CAN Signal			CNN-LSTM	
[107]	2022	DL	Unsupervised	Simulation		✓	Plateau, Continuous Change,Playback, Suppress	CNN
[108]	2022	DL	Unsupervised	OTIDS [32]		✓	Payload value Manipulation	AE
[109]	2023	ML \ DL	Unsupervised	Car Hacking [89], Survival Analysis Dataset [93]		✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Flooding, Malfunction	LSTM, GNB
[110]	2022	DL	Unsupervised	Own		✓	Spoofing, Bus-off, Masquerade, SOME attacks	GAN
CAN Frame-Based Attack Detection								
[111]	2021	DL	Unsupervised	Own	✓	✓	Random CAN payload Values	LSTM
[112]	2021	DL	Unsupervised	Car Hacking [89]	✓	✓	DoS, Fuzzy, RPM, Gear Spoofing	LSTM
[113]	2024	DL	Unsupervised	Car Hacking [89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	RNNs and AEs
[114]	2023	DL	Self-supervised	Survival Analysis Dataset [93]	✓	✓	Malfunction	Transformer
[115]	2021	DL	Unsupervised	ML350 [91]	✓	✓	DoS , Fuzzy	AE, GMM
[116]	2021	ML	Supervised	Simulation, OTIDS [32]	✓	✓	(Gear, RPM) Spoofing	FLXGBoost
[117]	2022	DL	Unsupervised	Car Hacking[89]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	CNNs, LSTMs
[118]	2020	DL	Unsupervised	SynCAN [122]	✓	✓	Flooding, Plateau, Continuous, Suppress, Playback	GRU AE
[119]	2024	DL	Unsupervised	Car Hacking [89], Car Hacking: Attack & Defence Challenge 2020 [94]	✓	✓	(Gear, RPM) Spoofing, DoS, Replay, Fuzzy	LSTM, CNN, AE
[120]	2020	DL	Unsupervised	Recan [130]	✓	✓	Interleave, Discontinuity, Data field anomalies	LSTM -AE
[121]	2023	DL	Unsupervised	Recan [130], car-hacking [54]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Masquerade, Seamless change, Replay	LSTM AE
[122]	2020	DL	Unsupervised	SynCAN [122]	✓	✓	Flooding, Plateau, Continuous, Suppress, Playback	LSTM AE
[123] Defence Challenge 2020 [94]	2022 ✓	DL ✓	Unsupervised Flooding, Spoofing, Replay, Fuzzy	Car Hacking: Attack &			LSTM	
[124]	2023	DL	Unsuper-vised/Super-vised	SynCAN [122], ROAD [90]	✓	✓	13 different attacks	GRU, Latent AE
[125]	2023	DL	Unsupervised	Survival Analysis Dataset [93], Car Hacking: Attack & Defence Challenge 2020 [94]	✓	✓	Spoofing, Replay, Fuzzy	LSTM-AEs
[126]	2024	DL	Unsupervised	Survival Analysis Dataset [93]	✓	✓	Flooding, Fuzzy, Malfunction	Transformer
[127]	2019	DL	Unsupervised	OTIDS [32]	✓	✓	DoS, Fuzzy, Impersonation	ConvLSTM

Table 5
Comparison of resource requirements for existing unknown attack detection methods.

	Reference	Latency (ms)	Model Size	Trainable Parameters
	[100]	1	–	–
	[101]	10/100 ms window	–	–
ID-based	[102]	0.0043	–	–
	[105]	1.7	–	–
	[106]	5.7	682 KB	–
	[107]	0.25	59.62 KB	6064
Payload-based	[110]	0.18	–	–
	[112]	0.023	3655 KB	–
	[113]	0.14 - 0.20	119 -272 KB for each gateway	3921
	[117]	128.73	–	–
	[118]	0.073 to 0.08	443 kB	–
	[119]	0.989	–	–
	[120]	650	Less than 10 MB	–
	[121]	0.07	–	–
	[124]	0.5	94 MB	191,434
	[125]	1.6	3.88 - 3.98 MB	–
CAN frame-based	[126]	0.26	–	–

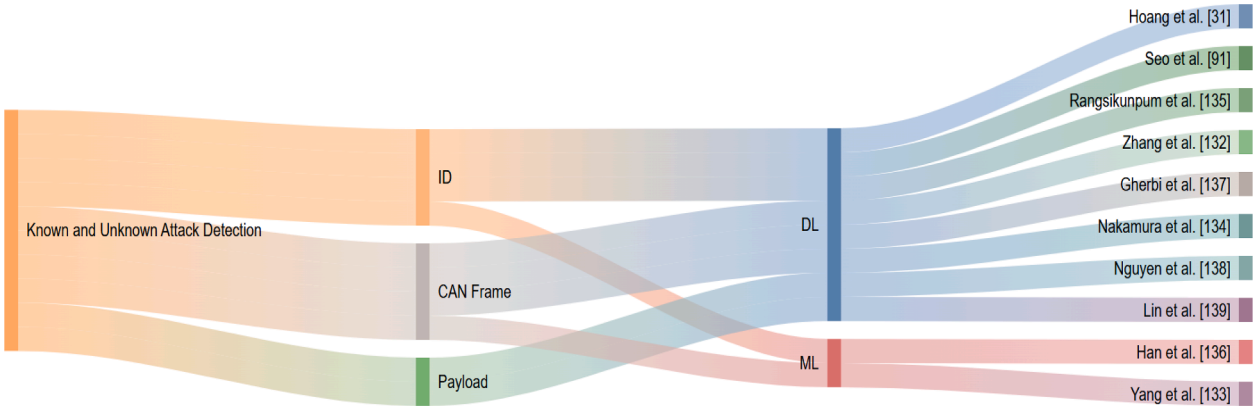


Fig. 8. Related work on known and unknown attack detection.

ble of identifying both known and unknown attacks [31,89,132–139], demonstrating significant advancements in this critical area of cybersecurity. This section reviews state-of-the-art studies and their limitations. This section comprises four subsections: three examine the features used in model construction, namely ID based detection, payload based detection, and CAN frame based detection, while the last subsection presents a comparative discussion. Fig. 8 illustrates exciting work on detecting both known and unknown attacks.

5.5.1. ID-based detection

This section reviews research where authors used only CAN IDs as the input feature to develop IDSs for detecting both known and new, unknown attacks.

Hoang et al. [31] and Seo et al. [89] showed that their proposed IDSs can effectively detect both previously known and new types of attacks. However, as their approaches rely primarily on the CAN ID as the sole feature, their effectiveness in detecting payload-based attacks is limited [16].

Hoang et al. [31] introduced a lightweight, semi-supervised learning-based IDS to detect attacks targeting in-vehicle networks. The IDS utilises a combination of autoencoders and generative adversarial networks (GANs). It was initially trained on unlabelled data to learn the patterns of both normal and malicious traffic, with a small number of labelled samples used later for supervised refinement. Even though they use only the CAN ID as the input feature, the number of trainable parameters is 2.15 million for the two models.

Seo et al. [89] introduced a GAN-based IDS (GIDS) aimed at enhancing security in-vehicle networks. GIDS was trained exclusively on patterns of CAN IDs extracted from CAN data, which were then transformed into simple image representations. It employs two discriminative models to identify both seen and unseen attack data. The first discriminator is specifically trained to identify known attacks, while the second is trained adversarially alongside the generator. As the generator produces modified images, the second discriminator is responsible for distinguishing these generated images from genuine CAN images.

Rangsikunpum et al. [135] developed a Binarized Neural Network (BNN)-based IDS, called BIDS, aimed at classifying known attacks and detecting unknown ones. The model adopts a hierarchical two-stage structure, with the first stage dedicated to detecting attacks and the second to classifying known attack types. To capture the sequential characteristics of CAN IDs, successive IDs are one-hot encoded and organised into a 48 × 48 two-dimensional grid. The proposed IDS has low computational overhead, making it suitable for deployment on cost-effective FPGA platforms.

Han et al. [136] developed an IDS for detecting and identifying abnormalities based on the periodic event-triggered intervals of CAN messages. Statistical features of the event-triggered intervals for each CAN ID were calculated. These features were then used to train ML models, including DT, RF, and XGBoost to classify attack types. This framework emphasises the event-triggered characteristics of CAN IDs and the statistical moments associated with intervals within a defined time window.

Although using the CAN ID as the only feature reduces the input features and results in a lightweight model, it limits the detection capability of payload manipulation attacks.

5.5.2. Payload-based detection

This section discusses IDSs that use the 8-byte CAN payload as an input feature to identify both known and new, unknown attacks.

Zhang et al. [132] introduced a DNN-based IDS capable of automatically extracting features from vehicle data packets. The model uses gradient descent with momentum (GDM) and an enhanced variant incorporating adaptive gain (GDM/AG). The results show strong performance in detecting replay attacks. However, a major limitation of the IDS is its dependence on access to the DBC file or detailed knowledge of the CAN payload, both of which are typically confidential and proprietary to the vehicle manufacturer [21].

Gherbi et al. [137] introduced a multivariate time series representation matrix to structure CAN data by integrating flow and payload information. They utilised autoencoder-based DL models such as Fully-Connected Networks (FCNs), CNNs, LSTMs, and Temporal Convolutional Networks (TCNs) to extract hierarchical representation vectors from the CAN matrix for anomaly detection. These vectors are derived either from the bottleneck layer in unsupervised tasks or the final layer in supervised tasks. The findings indicate that TCNs and LSTMs achieve strong performance, demonstrating their ability to effectively capture information from the representation matrix during training.

These payload-based detection methods overlook attacks targeting the CAN ID field and rely solely on binary classification, limiting their usefulness in selecting appropriate countermeasures or identifying specific attack types [131].

5.5.3. CAN frame-based detection

This section reviews IDSs that use the CAN frame (CAN IDs and payload) as input features to detect both known and new, unknown attacks.

Nguyen et al. [138] introduced a semi-supervised learning-based IDS that combines a variational autoencoder (VAE) with adversarial environment reinforcement learning (AERL) for multiclass classification. The proposed IDS is able to detect both known and unknown attacks. The objective of this approach is to improve training efficiency by minimising the reliance on labelled data.

Nakamura et al. [134] introduced a hybrid approach that integrates a supervised model based on LightGBM with an unsupervised model using an autoencoder to address the challenge of transferring knowledge across multiple car models for detecting and classifying attacks. Time differences between consecutive CAN IDs, along with CAN ID and payload values, were used as input features. The experimental results indicated that the hybrid model achieved superior performance compared to the pre-trained LightGBM model.

Lin et al. [139] introduced a two-stage IDS that combines incremental learning (IL) and a DNN, referred to as IL-DNN, to address changes in driving environments and behaviours. In the offline training stage, the DNN was applied to actual CAN data to develop a basic classification model. These predicted class labels were then used in the second stage. In the online detection and updating stage, the DNN model was updated using the IL approach with new, unlabeled data, while simultaneously performing intrusion detection. However, this approach risks degrading model performance if the original model's predictions are incorrect. However, both proposed IDSs in [134] and [139] are limited to binary classification, do not consider multi-class classification for known attacks, and do not account for the model size.

The majority of the previously discussed studies rely on either supervised or unsupervised learning approaches. To combine the advantages of both, Yang et al. [133] introduced MTH-IDS, a multi-tiered IDS, aimed at securing both in-vehicle and external networks against cyber-attacks. MTH-IDS uses ML algorithms and combines supervised and unsupervised models. The proposed MTH-IDS includes two traditional ML stages: data pre-processing and feature engineering. In the first tier, four tree-based supervised models, DT, RF, ET, and XGBoost, are used to detect known attacks. The second tier incorporates a stacking ensemble model alongside Bayesian optimization using the tree Parzen estimator (BO-TPE) to enhance the accuracy of the base learners. For unknown attack detection, the third tier introduces a novel unsupervised CL-k-means model. Lastly, the fourth tier applies Bayesian optimization with a Gaussian process (BO-GP) and two biased classifiers to refine the performance of the unsupervised learners. Despite achieving good results and a small model size of 2.61 MB, the proposed IDS presents some limitations. In the unsupervised model, the authors add an additional tier with two biased classifiers to improve the results. However, training these biased classifiers on FPs and FNs can degrade performance on unseen data. Furthermore, incorporating this layer compromises the model's unsupervised nature by introducing reliance on labelled data, which is often impractical in real-world applications. Moreover, the authors used only four features–CAN ID, and selected three features from the payload field which are DATA[5], DATA[3], and DATA[1] to train the model after feature extraction. Although feature selection approaches may lead to more efficient models, they create the risk that attackers could manipulate features not considered during the model's training process [140]. This presents a critical limitation in CAN bus data for two reasons. First, prioritising a subset of payload features while overlooking others may give attackers the opportunity to target the ignored features to bypass detection [141,142]. Second, the continuous evolution of attack methods implies that features effective for detecting one type of attack may no longer be suitable for identifying emerging or unknown attacks [140].

Yang et al. [133] employed conventional ML models in their proposed IDS due to their lower computational cost compared to DL algorithms. However, DL has shown superior performance in processing large volumes of data efficiently and at a faster rate [143]. Considering that modern vehicle ECUs produce around 2000 CAN frames per second [89], this capability is essential to handle the extensive data of the CAN bus. Moreover, multiple studies have highlighted the superior performance of DL-based IDSs over traditional ML-based IDSs in automotive applications [144]. Several factors contribute to this superiority: DL methods are more adaptive and continuously updated with incoming data, making them particularly well-suited to the dynamic nature of CAN bus data [132]. Moreover, traditional ML approaches often depend on manual feature engineering, such as correlation-based feature selection, which can be time-consuming [45]. In contrast, DL methods automatically learn and extract features, allowing models to identify optimal representations directly from raw data [46]. Furthermore, DL-based IDSs are particularly effective at identifying previously unseen attacks and are better suited to scaling with the complexity of in-vehicle network data without compromising performance [46].

To address these limitations, Althunayyan et al. [5] developed a multi-stage IDS designed to identify seen and previously unseen attacks, considering that some attacks may evade detection and be misclassified as benign. In the first stage, a supervised ANN is used to identify and categorise known attacks, whereas the subsequent stage applies an unsupervised LSTM autoencoder to identify unknown attacks that are not captured by the initial model. If the first model fails to identify malicious traffic and classifies it as benign, the anomaly detection model detects deviations from learned patterns and flags them as unseen attacks. Table 6 summarises the details of known and unknown attack detection studies.

A common limitation of most proposed CAN frame-based detection approaches lies in their deployment strategy. The majority adopt a traditional centralised learning model, requiring the transmission of large volumes of data to the cloud for both training and testing. This raises concerns related to privacy, high communication overhead, and increased response times [49].

5.5.4. Comparative discussion

For ID-based approaches, several studies evaluate their methods on the same benchmark dataset, including Hoang et al. [31], Seo et al. [89], and Rangsikunpum et al. [135], enabling direct comparison. Hoang et al. [31] report the highest performance, achieving an F1-score of 0.9984 and a low error rate of 0.1% using limited labelled data. Similarly, the GAN-based IDS proposed by Seo et al. [89] achieves an average accuracy of 100% for the first discriminator and 98% for the second; how-

Table 6
Summary of related work on known and unknown attack detection methods.

Reference	Year	ML \ DL	Category	Dataset	Algorithm	M-C	ID	Payload	FL[a]
ID-Based Attack Detection									
[31]	2022	DL	Semi-supervised	Car-Hacking [89]	AE, GAN		✓		
[89]	2018	DL	Unsupervised	Car-Hacking [89]	GAN		✓		
[135]	2024	DL[a]	Semi-supervised	Car Hacking [89], Survival Analysis Dataset [93]	BNN, GAN	✓	✓		
[136]	2021	ML[a]	Unsupervised	Own	DT, RF, XGBoost	✓	✓		
Payload-Based Attack Detection									
[132]	2019	DL[a]	Supervised	Simulation	DNN			✓	
[137]	2020	DL	Supervised/Unsupervised	SynCAN [122]	FCN, CNN, TCN, LSTM, AE			✓	
CAN Frame-Based Attack Detection									
[134]	2021	DL[a]	Unsupervised	Survival Analysis					
Dataset [93]	LightGBM, AE		✓	✓					
[138]	2024	DL	Semi-supervised	car-hacking [54], ROAD [90]	VAE and AERL	✓	✓	✓	
[139]	2021	DL	Supervised/Semi-supervised	car-hacking [54]	DNN and IL		✓	✓	
[133]	2022	ML[a]	Hybrid	Car-Hacking [89], CICIDS2017 [145]	DT, RF, ET, XGBoost, CL-k-means	✓	✓	✓	
[5]	2024	DL[a]	Hybrid	Car-Hacking [89]	ANN-LSTM AE	✓	✓	✓	✓

[a] **DL:** Deap Learning, **FL:** Federated Learning, **M-C:** Multi-class classification.

Table 7
Comparison of resource requirements for existing known and unknown attack detection methods.

Latency (ms)	Model Size	Trainable Parameters	
	[31]	0.63-0.69	–	2.15 million
	[89]	0.092	–	–
ID-based	[135]	0.170	4.07 Mb	–
	[132]	2-4	–	–
Payload-based	[137]	–	0.01 - 0.3MB	2920 - 75,238
	[138]	3.21	2,542 KB	–
	[139]	0.015	–	–
	[133]	0.6	2.61 MB	–
CAN frame-based	[5]	–	2.98 MB	253,582

ever, accuracy alone may not be a reliable metric for comprehensive evaluation. The BIDS CH model proposed by Rangsikunpum et al. [135] achieves 99.72% accuracy and F1-scores above 99% for each class of known attacks. For unknown attack classification, it attains an F1-score of 98.63%, with gear and rpm attacks exceeding 0.99. In contrast, recall for DoS and fuzzy attacks decreases to 0.98 and 0.84, respectively, highlighting a trade off between detecting known and unknown attacks. For payload-based approaches, Zhang et al. [132] propose a DNN-based IDS that achieves approximately 98% accuracy, with a true positive rate of around 98% and a low false positive rate of 1-2%. In contrast, the unsupervised approach proposed by Gherbi et al. [137] shows that, among various algorithms, TCNs perform best across most attack scenarios, achieving F1-scores in the range of 97-99%. For CAN frame-based approaches, the IDS proposed by Nguyen et al. [138] achieves high performance across all metrics on two datasets for known attack detection, outperforming baseline models. The proposed system also detects unknown attacks with high F1-scores ranging from 0.88 to 0.99 across different attack types. The MTH-IDS proposed by Yang et al. [133] proves effective against spoofing and DoS attacks, achieving optimal scores, but shows reduced performance for fuzzy attacks, with an F1-score of 0.8439. Althunayyan et al. [5] report F1-scores exceeding 0.99 for known attacks and around 0.95 for all unknown attacks, alongside a detection rate of 99.99% and a low false alarm rate of 0.016%. Latency, model size, and the number of trainable parameters are also key evaluation metrics and are shown in Table 7. Hoang et al. [31] report a large number of trainable parameters, which may lead to an increased model size. Zhang et al. [132] and Nguyen et al. [138] achieve a latency of 2 to 4 ms, which is relatively high compared to other methods. In addition, MTH IDS [133] and Althunayyan et al. [5] adopt similar strategies by using CAN frames to detect both known and unknown attacks. While the former relies on ML and the latter on DL, both approaches exhibit comparable model sizes, indicating that ML and DL techniques can achieve similar levels of model sizes. Notably, despite incorporating two models, the approaches remain lightweight and practical for deployment.

5.6. Evaluation metrics

In this section, we review all the evaluation metrics applied to assess the effectiveness of the IDS approaches in previously reviewed papers. The aim is to emphasise the importance of considering these metrics when designing models, rather than focusing on a few while ignoring others, to develop more deployable solutions. Based on the reviewed papers, we categorise the evaluation metrics into performance metrics, time complexity metrics, memory requirement metrics, and other metrics.

Performance metrics assess a model's effectiveness, including accuracy, F1-score, precision, recall (also known as Detection Rate (DR)), Error Rate (ER), confusion matrix, False Negative Rate (FNR), True Positive Rate (TPR), False Positive Rate (FPR), and True Negative Rate (TNR), also known as specificity. Additionally, False Alarm Rate (FAR), Receiver Operating Characteristic (ROC) Curve, Area Under the ROC Curve (AUC-ROC), and Area Under the Precision-Recall Curve (AUPR) are commonly used. These metrics are computed using True Positives

(TP), False Positives (FP), False Negatives (FN), and True Negatives (TN). Furthermore, the G-mean score and Matthews Correlation Coefficient (MCC) are valuable for evaluating model performance, particularly in cases of significant class imbalance [103,107]. Other relevant metrics include kappa and loss. Table 8 presents the performance metrics used in the reviewed papers. Accuracy, F1-score, precision, and recall are the most commonly used metrics, while AUPR, MCC, TP, FP, FN, TN, G-mean score, FAR, Kappa, and loss are rarely used.

For time complexity, several measures are commonly used, including training time, detection (inference) time, and latency. Regarding memory requirement metrics for evaluating model size, key metrics include the number of trainable parameters (which reflects memory usage), the model size in megabytes or kilobytes, and the number of Floating Point Operations (FLOPs). Other metrics, which are less commonly used in the reviewed papers, include resource allocation, power consumption, and Multiply-Accumulate (MAC) operations, which measure the speed of DL models [63]. Table 9 shows the time, memory, and other metrics used in the reviewed papers. Most papers evaluate detection latency or inference time as measures of time complexity, while only a few have measured memory footprint and parameter count.

Most studies have focused on some performance metrics while giving less consideration to time and memory requirements. Considering all these metrics (performance, time, and memory) makes the proposed models more deployable and easier to compare with other works.

6. Federated learning for in-vehicle networks

This section starts with an overview of the FL approach, followed by a review of existing FL-based in-vehicle IDSs, and concludes with their limitations.

6.1. Overview of federated learning

FL is a privacy-preserving decentralised learning technique that trains models locally without transferring row data to a centralised server [146]. Instead, it transfers model parameters to a centralised server, which aggregates the clients' models to build a shared global model [147]. The incorporation of FL into IDSs enhances security and privacy, addressing the growing challenges of protecting data in an increasingly interconnected world. While ML and DL have made notable progress in-vehicle IDSs, it is crucial to recognise their limitations, particularly regarding data privacy and communication efficiency. FL mitigates these challenges by enabling local model training while preserving the privacy of raw data [148]. FL is well-suited for in-vehicle IDSs for the following reasons:

- FL approach maintains data privacy by periodically transmitting learned model parameters to the cloud server instead of sharing raw data. This aligns with various data protection regulations, such as GDPR (Europe), CCPA (California), PIPEDA (Canada), and LGPD (Brazil), which are designed to prevent the unauthorised transfer of sensitive information.
- FL facilitates the efficient creation of a robust global model by multiple participants while ensuring the privacy of individual user data. It enables real-time model updates and data access without the need to communicate with a central server.
- FL minimises latency by eliminating the need to transmit raw data to a central server [147].
- Referring to the 2020 guidelines of the International Telecommunication Union [149] for IDS in vehicular networks, an in-vehicle IDS should be capable of updating its rule set regularly.
- FL improves the adaptability of IDS to new, previously unseen attacks by incorporating local models updated with those trained on newly detected attacks. This enables the continuous updating of models as new data becomes available, ensuring effective response to evolving threats in real-time.
- FL enables the development of a universal model that covers diverse driving scenarios, vehicle states, and driving behaviors [150].

As depicted in Fig. 9, the standard cloud-based FL architecture consists of a cloud server and multiple N clients (vehicles). Selected clients download the global model from the server, perform several rounds of local training using their own private data, and subsequently return the updated model weights to the server for aggregation. This iterative process continues until the model reaches the desired level of accuracy.

6.2. Federated learning for intrusion detection systems for in-vehicle networks

Driss et al. [151] proposed an FL-based framework to safeguard vehicular sensor networks against cyber attacks. The authors highlighted the importance of lightweight security solutions due to the limited resources of smart sensing devices in such networks. To tackle this challenge, they employed a combination of Gated Recurrent Units (GRU) and an ensemble method using RF to aggregate the global ML models. The dataset was evenly distributed among the clients.

Shibly et al. [152] designed a personalised FL-based IDS without requiring any data sharing. The authors explored both supervised and unsupervised methods within the FL framework, including CNN, XGBoost, MLP, and AE. Although their results were promising for both binary and multiclass classification, they overlooked the presence of non-IID data distributions.

Yu et al. [153] proposed an FL-based IDS employing LSTM, which takes advantage of the periodic nature of CAN communications to predict the arbitration IDs of incoming messages. Each 11-bit arbitration ID is converted into a one-hot encoded vector and input to the LSTM for next-ID prediction. The dataset is evenly partitioned among clients, with each assigned 1000 training instances and 200 testing instances. A comparison with the centralised IDS showed a 0.071 accuracy reduction for the FL-based IDS. However, the authors proposed that this reduction could be addressed with a cumulative error scheme.

Zhang et al. [154] designed an IDS using a graph neural network to detect anomalies on the CAN bus in just 3 milliseconds. The IDS utilises a two-stage cascade of classifiers: one focuses on detecting anomalies within a single class, while the other categorises detected attacks into multiple classes. The multi-class classifier is enhanced with an OpenMax layer to enable the detection of novel anomalies from classes not encountered during training.

Yang et al. [155] proposed an IDS using a federated DL framework that leverages recurring patterns in network messages and employs the ConvLSTM architecture. Clients were assigned differing amounts of data, between 50 and 3500 samples, to simulate a non-IID setting; however, the precise details of data allocation across clients and classes were not specified.

Taslimasa et al. [156] introduced ImageFed, a privacy-preserving IDS that employs federated CNNs. To create a non-IID environment, data were allocated among clients according to Dirichlet(μ) distribution, with (μ) values ranging between 0.1 and 0.7. The authors investigated two scenarios that could negatively impact FL performance: non-IID clients and restricted access to training data.

Longari et al. [121] deployed their proposed IDS, CANdito, presented in Section 5.4, in an FL setting to assess the detection effectiveness and communication cost in comparison with a centralised IDS. The experimental results indicate that FL offers a viable solution in practical scenarios where data privacy and security are critical. Although the federated model exhibits a slight reduction in detection performance compared to the centralised version, it still delivers robust results.

To overcome the challenge of DL models requiring large amounts of data to reach optimal performance, particularly in the case of CAN bus IDS, Hoang et al. [157] introduced CANPerFL, an IDS that employs a personalised FL approach to combine data from various vehicle models. Their approach builds a global model using limited training data from

Table 8
Performance metrics used in existing works.

Performance Metrics																				
Article	Accuracy	F1	Precision	Recall	Conf. Matrix	TPR	FPR	FNR	TNR	ER	ROC	AUC-ROC	AUPR	MCC	TP, FP	FN, TN	G-mean Score	FAR	Kappa	Loss
[55]	✓	✓	✓	✓								✓								
[65]	✓	✓	✓	✓		✓	✓					✓								
[113]	✓	✓	✓	✓																
[111]	✓	✓	✓	✓							✓									
[6]	✓	✓	✓	✓		✓	✓	✓	✓			✓								
[112]	✓	✓	✓	✓								✓							✓	
[133]	✓	✓		✓	✓													✓		
[138]	✓	✓	✓	✓			✓	✓												
[134]	✓	✓	✓	✓	✓			✓												
[135]	✓	✓	✓	✓	✓															
[118]	✓						✓													
[109]	✓	✓	✓	✓																
[123]	✓	✓	✓	✓								✓								
[121]	✓	✓	✓	✓		✓	✓							✓						
[88]	✓	✓	✓	✓	✓					✓	✓									
[80]	✓	✓	✓	✓																
[125]	✓	✓	✓	✓								✓								
[77]	✓	✓	✓	✓								✓								
[85]	✓	✓	✓	✓	✓		✓	✓				✓								
[56]	✓														✓					
[57]	✓	✓	✓	✓	✓															
[73]	✓	✓	✓	✓	✓						✓									
[58]	✓	✓	✓	✓			✓	✓												
[61]	✓		✓			✓														
[62]	✓	✓	✓		✓		✓	✓	✓			✓								
[37]	✓	✓	✓	✓	✓	✓	✓					✓								
[34]	✓	✓	✓	✓			✓	✓												
[6]	✓	✓																		
[76]	✓	✓	✓	✓							✓									✓
[64]	✓	✓				✓	✓													
[136]	✓	✓										✓								
[68]	✓	✓	✓	✓	✓															
[74]	✓	✓	✓	✓				✓		✓		✓								
[119]	✓	✓	✓	✓			✓	✓												
[89]	✓		✓	✓								✓								
[71]	✓	✓				✓	✓													
[78]	✓	✓	✓	✓																✓
[79]	✓	✓	✓	✓	✓															
[81]	✓	✓	✓	✓	✓															
[83]	✓	✓	✓	✓	✓															
[139]	✓	✓																		
[114]	✓	✓	✓	✓																
[115]	✓	✓	✓	✓																
[116]	✓	✓	✓	✓								✓								
[105]	✓	✓									✓									
[70]	✓	✓		✓			✓													
[84]	✓	✓	✓	✓								✓								
[107]	✓							✓				✓		✓						
[5]	✓	✓	✓	✓	✓													✓		
[110]	✓		✓	✓			✓	✓												
[126]		✓	✓	✓							✓									
[117]		✓	✓	✓																
[120]		✓	✓	✓								✓								
[127]		✓	✓	✓							✓									
[86]		✓	✓	✓	✓															
[137]		✓	✓	✓																
[124]		✓				✓	✓	✓	✓											
[31]		✓	✓	✓						✓										
[87]		✓	✓	✓	✓				✓			✓								
[106]		✓	✓	✓						✓										
[75]		✓	✓	✓	✓		✓					✓								
[54]		✓	✓	✓	✓			✓		✓										
[60]		✓	✓	✓			✓				✓									
[69]		✓	✓	✓				✓												
[102]		✓	✓	✓	✓		✓	✓												
[63]		✓	✓	✓	✓			✓												
[72]		✓	✓	✓																
[82]		✓	✓	✓																
[101]		✓					✓	✓												
[108]			✓									✓	✓							
[100]					✓						✓				✓	✓				
[66]					✓						✓									
[132]					✓	✓	✓				✓									
[122]						✓			✓			✓								
[59]							✓				✓									
[67]												✓								
[103]																	✓			
[104]											✓	✓								

Table 9
Time complexity, memory requirements, and other metrics.

Article	Time Complexity Metrics			Memory Requirements Metrics			Other Metrics		
	Training Time	Detection Latency/Inference Time	Execution Time	Memory Footprint (Size)	Parameters	Training Cost (FLOPS)	Resource Utilization	Power/Energy Consumption	MAC
[54]	✓	✓			✓				
[68]	✓	✓							
[78]	✓								
[112]	✓	✓		✓					
[84]	✓								
[88]	✓	✓				✓			
[100]	✓	✓							
[113]	✓	✓		✓	✓	✓		✓	
[132]	✓	✓	✓						
[137]	✓			✓	✓				
[59]	✓	✓							
[133]	✓	✓		✓					
[57]		✓		✓			✓	✓	
[63]		✓			✓				✓
[124]		✓		✓	✓				
[64]		✓		✓					
[76]		✓							
[121]		✓							
[101]		✓							
[102]		✓					✓	✓	
[105]		✓							
[106]		✓		✓					
[107]		✓		✓	✓				
[110]		✓							
[117]		✓							
[118]		✓		✓			✓		
[119]		✓					✓		
[125]		✓		✓					
[126]		✓							
[31]		✓			✓				
[89]		✓							
[135]		✓		✓					
[138]		✓		✓					
[82]		✓							
[139]		✓							
[61]			✓		✓				
[66]			✓						
[70]			✓						
[71]			✓						
[81]			✓						
[83]			✓						
[136]			✓						
[120]			✓	✓					
[5]				✓	✓				

each automaker, providing shared knowledge that enhances the performance of individual participants. Experimental findings reveal that the proposed model achieves a 4% overall improvement in F1-score compared to baseline models. Its effectiveness is particularly evident in scenarios where participants have access to limited local datasets.

Althunayyan et al. [150] deployed their proposed IDS from [5] within a Hierarchical FL (H-FL) framework. This framework aims to address the limitations of standard FL-based IDSs, which rely on a single central aggregator, leading to system slowdowns and a single point of failure that compromises robustness and scalability. By incorporating several edge aggregators along with the main aggregator, the proposed H-FL mitigates the risk of single-point failures, enhances scalability, and optimises the distribution of computational load. The experiments indicate that incorporating the IDS into the H-FL framework results in an F1-score improvement of up to 10.63%, effectively mitigating the issues of dataset diversity and attack coverage encountered in edge-FL.

Table 10 summarises previous work. If the authors do not explicitly state the aggregation function, as in [151,152], it is assumed that FedAvg was employed.

6.3. Limitations of existing FL-based IDSs

Although previous works have contributed to the field of FL-based in-vehicle IDSs, they exhibit certain limitations. A major challenge in FL is managing non-independent and identically distributed (non-IID) data, where significant differences in client training data lead to varied data distributions across clients [159]. In real-world use cases, data tends to be non-IID due to differences in user behaviour, preferences, and environments [160]. However, most existing studies overlook Non-IID data, often assuming that clients are assigned either an equal number of samples or a balanced representation of all classes (i.e., attack types). This assumption contradicts real-world FL scenarios, which inherently involve Non-IID data distributions [53], resulting in an unrealistic evaluation of FL-based IDS performance [161]. Only a few studies [150,155,156] have explicitly considered Non-IID data distributions. In [155], nine candidate clients are considered, each holding a different number of data samples (ranging from 50 to 3500), although the distribution of samples across classes and between clients remains unclear. In contrast, [156] and [150] implement a Non-IID setting by distributing data to vehicles using a *Dirichlet*(μ) distribution, where the (μ) parame-

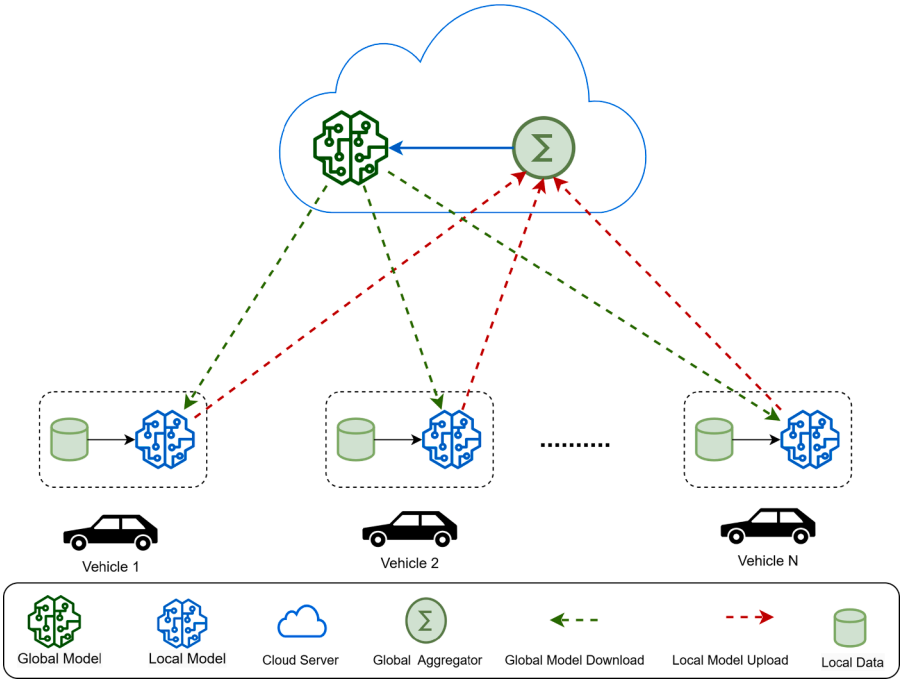


Fig. 9. Federated learning architecture.

Table 10
FL-based IDSs for in-vehicle network.

Reference	FL	Non-IID	Aggregation Function	Dataset	FL Implementation
[156]	Standard	✓	FedAvg	car-hacking [54]	PyTorch
[153]	Standard	x	FedAvg	HCRL CAN Intrusion Detection [32]	N\ A
[152]	Standard	x	FedAvg	car-hacking [54], NAIST CAN attack dataset[37]	Keras, TensorFlow
[154]	Standard	N\ A	FedAvg, FedProx	READ [158]	N\ A
[151]	Standard	x	FedAvg	Car Hacking: Attack & Defence Challenge 2020 [94]	Keras, TensorFlow
[155]	Standard	✓	FedAvg	HCRL CAN Intrusion Detection [32]	N\ A
[121]	Standard	N\ A	FedAvg, FedProx	Recan [130]	N\ A
[157]	Standard	–	FedAvg	Own	Pytorch, Flower
[150]	Hierarchical	✓	FedAvg	car-hacking [54], Car Hacking [89]	Flower

ter is adjusted between 0.1 and 0.7 to control the level of Non-IIDness. Another key limitation in FL-based in-vehicle IDS research is the lack of client selection strategies. Real-world FL scenarios involve clients with varying resources, network stability, and data quality. However, existing studies assume equal participation in every training round, ignoring the dynamic nature of vehicular environments and the need for adaptive selection.

7. Future research directions

This section identifies the limitations of existing approaches and explores potential future research directions to improve the security of in-vehicle networks.

- **Limited Access to Real-World Datasets:** It is a fact that the best ML/DL-based models are derived from high-quality data. Therefore, a key challenge in-vehicle security research is the limited access to real-world datasets that reflect diverse driving behaviours and environments, such as urban, mountainous, and rural terrains. Existing datasets do not fully represent real driving conditions primarily due to privacy and legal constraints [97]. Consequently, most proposed IDSs have been trained and evaluated under restricted conditions, limiting their ability to generalise normal vehicle behaviour across varied scenarios. Moreover, the literature review highlights that publicly available datasets are often less challenging, allowing even simple ML models to achieve high accuracy. However, the effectiveness of these ML/DL-based IDSs in real-world applications may not be guaranteed. Since in-vehicle networks demand high reliability, this could hinder their practical implementation. A promising research direction is the exploration of streaming learning, which enables models to dynamically adapt in real-time as vehicles encounter different driving conditions. This approach could enhance detection accuracy and improve system adaptability across diverse environments.
- **Protecting the in-vehicle IDSs:** In-vehicle IDSs are vulnerable to adversarial attacks, as recent studies [1,162] have highlighted the vulnerabilities of these systems. Adversarial attacks manipulate input data to deceive models into producing incorrect or misclassified outputs [162], thereby threatening the safety and security of CAVs. The literature clearly shows that almost no proposed IDS has considered protecting the system from adversarial attacks, except for the work in [162], where Li et al. [162] developed a defence strategy to protect LSTM-based IDSs from adversarial attacks. Consequently, deploying IDSs without properly evaluating their adversarial robustness not only compromises vehicle security but also increases the risk of malicious manipulation. Thus, training IDSs on adversarial samples to detect these attacks is a possible solution. Moreover, adapting defence strategies from other fields could significantly enhance the resilience of in-vehicle IDSs, ensuring robustness against both known and emerging threats, including adversarial examples. This remains a crucial area for future research.
- **False Positives in Unsupervised Learning:** As with all unsupervised learning methods [16], anomaly detection models usually suf-

fer from FPs. In critical systems, minimising false alarms is essential for maintaining system reliability. Some existing approaches train biased classifiers to reduce FPs and FNs, but this shifts the model away from being purely unsupervised. Future research should focus on finding practical solutions that reduce FPs without compromising the model's unsupervised nature. One potential direction is to leverage eXplainable AI (XAI) techniques to make the behaviour of in-vehicle IDSs more interpretable and transparent. While AI methods have shown great potential in combating cyberattacks, they often generate false alarms and produce decisions that are difficult to interpret, leading to uncertainty and distrust [163]. XAI methods, such as SHapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME), enhance the interpretability of IDSs by providing clearer insights into model decisions, enabling more effective responses to alarms, and fostering greater trust in AI-driven security systems [164]. Further exploration of XAI could significantly improve both the transparency and reliability of AI-based in-vehicle IDSs.

- **Vehicle-Specific Models and Generalisation Challenges:** Another limitation is the assumption that all vehicles in the FL environment share the same make, model, CAN IDs, and payload interpretations. This assumption could necessitate developing separate models for each vehicle make and model, leading to increased complexity. Generalising the IDS to learn across different vehicle types, rather than relying on distinct models for each, remains a significant challenge due to variations in CAN bus data and the lack of access to DBC files, which define signal meanings. While FL has shown promise in enhancing IDS performance by integrating models from diverse driving scenarios and vehicle states, achieving robust model generalisation across all vehicle types is complex. Future research could explore techniques such as domain adaptation or transfer learning to address variations between different vehicle models and develop a system that works across various vehicle types.
- **Client Selection in FL:** Another future direction for improving the efficiency of the FL process is exploring methods for selecting or excluding clients. Given the heterogeneity of in-vehicle network traffic, it is neither practical nor efficient to include all vehicles as federated clients [155]. Investigating effective client selection strategies is crucial to optimising model accuracy while minimising computational and communication overhead. Based on the reviewed papers on FL-based in-vehicle IDS, no work has been done on client selection using in-vehicle traffic data. Potential strategies could involve selecting clients based on similarities in CAN bus patterns, driving behaviour, or geographic location to ensure that the FL process remains efficient.
- **Evaluation Metrics:** Most of the reviewed studies evaluated their proposed IDSs using performance metrics such as accuracy, F1-score, precision, and recall. However, many existing IDSs either fail to consider memory constraints and real-time requirements when designing in-vehicle IDSs [16], making many proposed IDSs impractical for real-world applications. Given the constrained memory resources of ECUs and the real-time requirements in-vehicle networks [113], an efficient IDS must be lightweight, have a small memory footprint [64,118], and satisfy real-time performance requirements. Designing in-vehicle IDS solutions requires careful consideration of deployment constraints [21]. The limited memory, processing power, and bandwidth of ECUs in-vehicle networks directly affect the feasibility and effectiveness of IDS development and deployment [16]. Moreover, since CAN is a time-critical system, inference time and detection latency are essential safety-related metrics for in-vehicle IDS to ensure real-time performance. Inference time is the time it takes for a trained model to generate predictions on a new data batch [63]. Latency, on the other hand, is the time a packet requires to be transmitted from its origin to its target destination [133]. According to the United States Department of Transportation, critical vehicle safety functions, including collision and attack alerts, are required to operate within a latency range of 10 to 100 ms [165]. Meanwhile, Vehicle-to-everything (V2X)-based autonomous and cooperative driving applications require even stricter latency, typically between 10 and 20 ms [166]. Therefore, to comply with real-time requirements, a vehicle-level IDS must process each network packet in under 10 ms.

8. Conclusion

CAVs improve transportation efficiency but are vulnerable to cybersecurity threats, particularly due to the insecurity of the CAN bus protocol. These cyberattacks can have severe consequences, such as compromising control over essential systems, necessitating robust and reliable security measures. ML-based in-vehicle IDSs offer an effective solution by detecting malicious activities in real time. The main contribution of this paper is a detailed survey of current ML and DL approaches for building in-vehicle IDSs, focusing on detecting known attacks (38 papers), unknown attacks (28 papers), and combined known and unknown attacks (11 papers). Moreover, we reviewed the evaluation metrics used by researchers to build their IDSs and categorised them into performance metrics, time complexity metrics, memory requirement metrics, and other metrics, emphasizing the importance of considering all these metrics to achieve more deployable solutions. Additionally, we reviewed research on FL-based IDSs (9 papers) applied to in-vehicle networks. The total number of reviewed papers in this survey is 86. Lastly, we present future directions that can help enhance the security and privacy of in-vehicle IDSs.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT-4 in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CRediT authorship contribution statement

Muzun Althunayyan: Writing – original draft, Visualization, Resources, Methodology, Formal analysis, Conceptualization; **Amir Javed:** Writing – review & editing, Supervision; **Omer Rana:** Supervision.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Aloraini, A. Javed, O. Rana, Adversarial attacks on intrusion detection systems in in-vehicle networks of connected and autonomous vehicles, Sensors 24 (12) (2024) 3848. https://doi.org/10.3390/s24123848
[2] SMMT Driving the Motor Industry, Connected and Autonomous Vehicles: The Global Race to Market, Technical Report, The Society of Motor Manufacturers and Traders Limited, 2019.
[3] J. Pickford, R. Attale, S. Shaikh, H.N. Nguyen, L. Harrison, Systematic risk characterisation of hardware threats to automotive systems, J. Auton. Transp. Syst. 1 (4) (2024) 1–36.
[4] O.Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, A. Mouzakitis, Intrusion detection systems for intra-vehicle networks: a review, IEEE Access 7 (2019) 21266–21289.
[5] M. Althunayyan, A. Javed, O. Rana, A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning, Veh. Commun. 49 (2024) 100837.

[6] A. Paul, M.R. Islam, An artificial neural network based anomaly detection method in can bus messages in vehicles, in: 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI), IEEE, 2021, pp. 1–5. https://doi.org/10.1109/ACMI53878.2021.9528201

[7] E. Aliwa, O. Rana, C. Perera, P. Burnap, Cyberattacks and countermeasures for in-vehicle networks, ACM Comput. Surv. (CSUR) 54 (1) (2021) 1–37. https://doi.org/10.1145/3431233

[8] L. Upstream Security, Upstream's 2024 global automotive cybersecurity report, 2024, Accessed: 2025-01-18, (https://upstream.auto/reports/global-automotive-cybersecurity-report/#).

[9] C. Young, J. Zambreno, H. Olufowobi, G. Bloom, Survey of automotive controller area network intrusion detection systems, IEEE Des. Test 36 (6) (2019) 48–55. https://doi.org/MDAT.2019.2899062

[10] K. Tindell, CAN injection: keyless car theft, 2023. https://kentindell.github.io/2023/04/03/can-injection/.

[11] J. Golson, Jeep hackers at it again, this time taking control of steering and braking systems, 2016, (accessed 1 April 2023), (https://www.theverge.com/2016/8/2/12353186/car-hack-jeep-cherokee-vulnerability-miller-valasek).

[12] T.K.S. Lab, New vehicle security research by KeenLab: experimental security assessment of BMW cars, 2018, (accessed 10 April 2023), (https://keenlab.tencent.com/en/2018/05/22/New-CarHacking-Research-by-KeenLab-Experimental-Security-Assessment-of-BMW-Cars/).

[13] T.K.S. Lab, Experimental security assessment on lexus cars, 2020, (accessed 10 April 2023), (https://keenlab.tencent.com/en/2020/03/30/Tencent-Keen-Security-Lab-Experimental-Security-Assessment-on-Lexus-Cars/).

[14] M. Bertoncello, C. Martens, T. Möller, T. Schneiderbauer, Unlocking the full life-cycle value from connected-car data, 2021, (accessed 6 April 2023), (https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/unlocking-the-full-life-cycle-value-from-connected-car-data).

[15] T. Hoppe, S. Kiltz, J. Dittmann, Applying intrusion detection to automotive it-early insights and remaining challenges, J. Inform. Assur. Secur. (JIAS) 4 (6) (2009) 226–235.

[16] S. Rajapaksha, H. Kalutarage, M.O. Al-Kadri, A. Petrovski, G. Madzudzo, M. Cheah, AI-based intrusion detection systems for in-vehicle networks: a survey, ACM Comput. Surv. 55 (11) (2023) 1–40. https://doi.org/10.1145/3570954

[17] L. Liu, J. Zhang, S.H. Song, K.B. Letaief, Client-edge-cloud hierarchical federated learning, in: ICC 2020-2020 IEEE International Conference on Communications (ICC), IEEE, 2020, pp. 1–6. https://doi.org/10.1109/ICC40277.2020.9148862

[18] U. Ahmad, M. Han, A. Jolfaei, S. Jabbar, M. Ibrar, A. Erbad, H.H. Song, Y. Alkhrijah, A comprehensive survey and tutorial on smart vehicles: emerging technologies, security issues, and solutions using machine learning, IEEE Trans. Intell. Transp. Syst. 25 (11) (2024) 15314–15341.

[19] S. Boumiza, R. Braham, Intrusion threats and security solutions for autonomous vehicle networks, in: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), IEEE, 2017, pp. 120–127. https://doi.org/10.1109/AICCSA.2017.42

[20] B.V. Kumar, J. Ramesh, Automotive in vehicle network protocols, in: 2014 International Conference on Computer Communication and Informatics, IEEE, 2014, pp. 1–5. https://doi.org/10.1109/ICCCI.2014.6921836

[21] S.-F. Lokman, A.T. Othman, M.-H. Abu-Bakar, Intrusion detection system for automotive controller area network (CAN) bus system: a review, EURASIP J. Wireless Commun. Netw. 2019 (2019) 1–17. https://doi.org/10.1186/s13638-019-1484-3

[22] P. Carsten, T.R. Andel, M. Yampolskiy, J.T. McDonald, In-vehicle networks: attacks, vulnerabilities, and proposed solutions, in: Proceedings of the 10th Annual Cyber and Information Security Research Conference, 2015, pp. 1–8.

[23] J. Liu, S. Zhang, W. Sun, Y. Shi, In-vehicle network attacks and countermeasures: challenges and future directions, IEEE Netw. 31 (5) (2017) 50–58.

[24] G. Dupont, J. den Hartog, S. Etalle, A. Lekidis, A survey of network intrusion detection systems for controller area network, in: 2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES), IEEE, 2019, pp. 1–6.

[25] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, K. Li, A survey of intrusion detection for in-vehicle networks, IEEE Trans. Intell. Transp. Syst. 21 (3) (2019) 919–933.

[26] T. Limbasiya, K.Z. Teng, S. Chattopadhyay, J. Zhou, A systematic survey of attack detection and prevention in connected and autonomous vehicles, Veh. Commun. 37 (2022) 100515.

[27] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, Comprehensive experimental analyses of automotive attack surfaces, in: 20th USENIX Security Symposium (USENIX Security 11), USENIX Association, San Francisco, CA, 2011.

[28] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al., Experimental security analysis of a modern automobile, in: 2010 IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 447–462.

[29] V. Chockalingam, I. Larson, D. Lin, S. Nofzinger, Detecting attacks on the CAN protocol with machine learning, Annu. EECS 558 (7) (2016).

[30] S. Woo, H.J. Jo, D.H. Lee, A practical wireless attack on the connected car and security protocol for in-vehicle CAN, IEEE Trans. Intell. Transp. Syst. 16 (2) (2014) 993–1006.

[31] T.-N. Hoang, D. Kim, Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders, Veh. Commun. 38 (2022). https://doi.org/10.1016/j.vehcom.2022.100520

[32] H. Lee, S.H. Jeong, H.K. Kim, OTIDS: a novel intrusion detection system for in-vehicle network by using remote frame, in: 2017 15th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2017, pp. 57–5709. https://doi.org/10.1109/PST.2017.00017

[33] K.-T. Cho, K.G. Shin, Error handling of in-vehicle networks makes them vulnerable, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1044–1055.

[34] M.D. Hossain, H. Inoue, H. Ochiai, D. Fall, Y. Kadobayashi, An effective in-vehicle CAN bus intrusion detection system using CNN deep learning approach, in: GLOBECOM 2020-2020 IEEE Global Communications Conference, IEEE, 2020, pp. 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9322395

[35] D.S. Fowler, J. Bryans, S.A. Shaikh, P. Wooderson, Fuzz testing for automotive cyber-security, in: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), IEEE, 2018, pp. 239–246.

[36] K. Iehira, H. Inoue, K. Ishida, Spoofing attack using bus-off attacks against a specific ECU of the CAN bus, in: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2018, pp. 1–4.

[37] M.D. Hossain, H. Inoue, H. Ochiai, D. Fall, Y. Kadobayashi, LSTM-based intrusion detection system for in-vehicle can bus communications, IEEE Access 8 (2020) 185489–185502. https://doi.org/10.1109/ACCESS.2020.3029307

[38] H.J. Jo, W. Choi, A survey of attacks on controller area networks and corresponding countermeasures, IEEE Trans. Intell. Transp. Syst. 23 (7) (2021) 6123–6141.

[39] G. Karopoulos, G. Kambourakis, E. Chatzoglou, J.L. Hernández-Ramos, V. Kouliaridis, Demystifying in-vehicle intrusion detection systems: a survey of surveys and a meta-taxonomy, Electronics 11 (7) (2022) 1072.

[40] A. Tomlinson, J. Bryans, S.A. Shaikh, Towards viable intrusion detection methods for the automotive controller area network, in: 2nd ACM Computer Science in Cars Symposium, 2018, pp. 1–9.

[41] G.K. Rajbahadur, A.J. Malton, A. Walenstein, A.E. Hassan, A survey of anomaly detection for connected vehicle cybersecurity and safety, in: 2018 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2018, pp. 421–426.

[42] G. Loukas, E. Karapistoli, E. Panaousis, P. Sarigiannidis, A. Bezemskij, T. Vuong, A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles, Ad Hoc Netw. 84 (2019) 124–147.

[43] N. Quadar, A. Chehri, B. Debaque, I. Ahmed, G. Jeon, Intrusion detection systems in automotive ethernet networks: challenges, opportunities and future research trends, IEEE Internet Things Mag. 7 (2) (2024) 62–68.

[44] B. Lampe, W. Meng, Intrusion detection in the automotive domain: a comprehensive review, IEEE Commun. Surv. Tutorials 25 (4) (2023) 2356–2426.

[45] J. Nagarajan, P. Mansourian, M.A. Shahid, A. Jaekel, I. Saini, N. Zhang, M. Kneppers, Machine learning based intrusion detection systems for connected autonomous vehicles: a survey, Peer-to-Peer Netw. Appl. (2023) 1–33. https://doi.org/10.1007/s12083-023-01508-7

[46] B. Lampe, W. Meng, A survey of deep learning-based intrusion detection in automotive applications, Expert Syst. Appl. 221 (2023) 119771. https://doi.org/10.1016/j.eswa.2023.119771

[47] M. Almehdhar, A. Albaseer, M.A. Khan, M. Abdallah, H. Menouar, S. Al-Kuwari, A. Al-Fuqaha, Deep learning in the fast lane: a survey on advanced intrusion detection systems for intelligent vehicle networks, IEEE Open J. Veh. Technol. 5 869–906 (2024).

[48] H. Taslimasa, S. Dadkhah, E.C.P. Neto, P. Xiong, S. Ray, A.A. Ghorbani, Security issues in internet of vehicles (IoV): a comprehensive survey, Internet Things 22 (2023) 100809.

[49] V.P. Chellapandi, L. Yuan, Ż. Stanislaw H., Z. Wang, A survey of federated learning for connected and automated vehicles, in: 2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC), IEEE, 2023, pp. 2485–2492. https://doi.org/10.48550/arXiv.2303.10677

[50] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Inf. Software Technol. 55 (12) (2013) 2049–2075.

[51] N. Alhirabi, O. Rana, C. Perera, Security and privacy requirements for the internet of things: a survey, ACM Trans. Internet Things 2 (1) (2021) 1–37.

[52] C. Wohlin, 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, 2014, pp. 1–10.

[53] J.L. Hernandez-Ramos, G. Karopoulos, E. Chatzoglou, V. Kouliaridis, E. Marmol, A. Gonzalez-Vidal, G. Kambourakis, Intrusion detection based on federated learning: a systematic review, arXiv preprint arXiv:2308.09522 (2023). https://doi.org/10.48550/arXiv.2308.09522

[54] H.M. Song, J. Woo, H.K. Kim, In-vehicle network intrusion detection using deep convolutional neural network, Veh. Commun. 21 (2020) 100198. https://doi.org/10.1016/j.vehcom.2019.100198

[55] R.U.D. Refat, A.A. Elkhail, A. Hafeez, H. Malik, Detecting can bus intrusion by applying machine learning method to graph based features, in: Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3, Springer, 2022, pp. 730–748.

[56] S.R. Nandam, A. Vamshi, I. Sucharitha, CAN intrusion detection using long short-term memory (LSTM), in: Computer Communication, Networking and IoT: Proceedings of 5th ICICC 2021, Volume 2, Springer, 2022, pp. 295–302.

[57] A. Rangsikunpum, S. Amiri, L. Ost, An FPGA-based intrusion detection system using binarised neural network for CAN bus systems, in: 2024 IEEE International Conference on Industrial Technology (ICIT), IEEE, 2024, pp. 1–6.

[58] Y. Wu, X. Tao, Network traffic anomaly detection in CAN bus based on ensemble learning, in: 2024 4th International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), IEEE, 2024, pp. 240–245.

[59] M.-J. Kang, J.-W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, PLoS ONE 11 (6) (2016) e0155781.

[60] F. Martinelli, F. Mercaldo, V. Nardone, A. Santone, Car hacking identification through fuzzy logic algorithms, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–7.

[61] F. Fenzl, R. Rieke, A. Dominik, In-vehicle detection of targeted CAN bus attacks, in: Proceedings of the 16th International Conference on Availability, Reliability and Security, 2021, pp. 1–7.

[62] S.B.H. Samir, M. Raissa, H. Touati, M. Hadded, H. Ghazzai, Machine learning-based intrusion detection for securing in-vehicle CAN bus communication, SN Comput. Sci. 5 (8) (2024) 1082.

[63] T.-D. Le, H.B.H. Truong, D. Kim, et al., Multi-classification in-vehicle intrusion detection system using packet-and sequence-level characteristics from time-embedded transformer with autoencoder, Knowl. Based Syst. 299 (2024) 112091.

[64] L. Zhang, X. Yan, D. Ma, Efficient and effective in-vehicle intrusion detection system using binarized convolutional neural network, in: IEEE INFOCOM 2024-IEEE Conference on Computer Communications, IEEE, 2024, pp. 2299–2307.

[65] M. Sami, M. Ibarra, A.C. Esparza, S. Al-Jufout, M. Aliasgari, M. Mozumdar, Rapid, multi-vehicle and feed-forward neural network based intrusion detection system for controller area network bus, in: 2020 IEEE Green Energy and Smart Systems Conference (IGESSC), IEEE, 2020, pp. 1–6.

[66] D. Aksu, M.A. Aydin, MGA-IDS: optimal feature subset selection for anomaly detection framework on in-vehicle networks-CAN bus based on genetic algorithm and intrusion detection approach, Comput. Secur. 118 (2022) 102717.

[67] S. Boumiza, R. Braham, An anomaly detector for CAN bus networks in autonomous cars based on neural networks, in: 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2019, pp. 1–6.

[68] S. Park, J.-Y. Choi, Hierarchical anomaly detection model for in-vehicle networks using machine learning algorithms, Sensors 20 (14) (2020) 3934.

[69] X. Zhang, X. Cui, K. Cheng, L. Zhang, A convolutional encoder network for intrusion detection in controller area networks, in: 2020 16th International Conference on Computational Intelligence and Security (CIS), IEEE, 2020, pp. 366–369.

[70] O. Minawi, J. Whelan, A. Almehmadi, K. El-Khatib, 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, 2020, pp. 41–47.

[71] A. Alfardus, D.B. Rawat, Intrusion detection system for can bus in-vehicle network based on machine learning algorithms, in: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE, 2021, pp. 0944–0949. https://doi.org/10.1109/UEMCON53757.2021.9666745

[72] A. NasrEldin, A.M. Bahaa-Eldin, M.A. Sobh, In-vehicle intrusion detection based on deep learning attention technique, in: 2021 16th International Conference on Computer Engineering and Systems (ICCES), IEEE, 2021, pp. 1–7.

[73] E. Alalwany, I. Mahgoub, Classification of normal and malicious traffic based on an ensemble of machine learning for a vehicle can-network, Sensors 22 (23) (2022) 9195.

[74] D. Ding, L. Zhu, J. Xie, J. Lin, In-vehicle network intrusion detection system based on Bi-LSTM, in: 2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP), IEEE, 2022, pp. 580–583.

[75] C.R. Kishore, D.C. Rao, J. Nayak, H.S. Behera, Intelligent intrusion detection framework for anomaly-based CAN bus network using bidirectional long short-term memory, J. Inst. Eng.: Ser. B 105 (2024) 1–24.

[76] A. Chougule, I. Kulkarni, T. Alladi, V. Chamola, F.R. Yu, HybridSecNet: in-vehicle security on controller area networks through a hybrid two-step LSTM-CNN model, IEEE Trans. Veh. Technol. 73 14580–14591 (2024).

[77] E. Alalwany, I. Mahgoub, An effective ensemble learning-based real-time intrusion detection scheme for an in-vehicle network, Electronics 13 (5) (2024) 919.

[78] D. Basavaraj, S. Tayeb, Towards a lightweight intrusion detection framework for in-vehicle networks, J. Sens. Actuator Netw. 11 (1) (2022) 6.

[79] K. Gao, H. Huang, L. Liu, R. Du, J. Zhang, A multi-attention based CNN-BiLSTM intrusion detection model for in-vehicle networks, in: 2023 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), IEEE, 2023, pp. 809–816.

[80] S.C. Kalkan, O.K. Sahingoz, In-vehicle intrusion detection system on controller area network with machine learning models, in: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), IEEE, 2020, pp. 1–6.

[81] W. Gou, H. Zhang, R. Zhang, Multi-classification and tree-based ensemble network for the intrusion detection system in the internet of vehicles, Sensors 23 (21) (2023) 8788.

[82] H. Ma, J. Cao, B. Mi, D. Huang, Y. Liu, S. Li, A GRU-based lightweight system for CAN intrusion detection in real time, Secur. Commun. Netw. 2022 (1) (2022) 5827056.

[83] M.H. Khan, A.R. Javed, Z. Iqbal, M. Asim, A.I. Awad, DivaCAN: detecting in-vehicle intrusion attacks on a controller area network using ensemble learning, Comput. Secur. 139 (2024) 103712.

[84] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, J.-H. Chen, Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks, Electronics 11 (14) (2022) 2180.

[85] M.D. Hossain, H. Inoue, H. Ochiai, D. Fall, Y. Kadobayashi, Long short-term memory-based intrusion detection system for in-vehicle controller area network bus, in: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), IEEE, 2020, pp. 10–17.

[86] M. Casillo, S. Coppola, M. De Santo, F. Pascale, E. Santonicola, Embedded intrusion detection system for detecting attacks over CAN-BUS, in: 2019 4th International Conference on System Reliability and Safety (ICSRS), IEEE, 2019, pp. 136–141.

[87] M. Nazeer, A. Alasiry, M. Qayyum, V.K. Madhan, G. Patil, P. Srilatha, Enhancing cyber security in autonomous vehicles: a hybrid XG boost-deep learning approach for intrusion detection in the CAN bus, J. Européen des Systèmes Automatisés 57 (5) (2024).

[88] T.P. Nguyen, H. Nam, D. Kim, Transformer-based attention network for in-vehicle intrusion detection, IEEE Access 11 (2023) 55389–55403.

[89] E. Seo, H.M. Song, H.K. Kim, GIDS: GAN based intrusion detection system for in-vehicle network, in: 2018 16th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2018, pp. 1–6. https://doi.org/10.1109/PST.2018.8514157

[90] M.E. Verma, M.D. Iannacone, R.A. Bridges, S.C. Hollifield, B. Kay, F.L. Combs, Road: the real ornl automotive dynamometer controller area network intrusion detection dataset (with a comprehensive can ids dataset survey & guide), arXiv preprint arXiv:2012.14600 (2020).

[91] M. Sami, Intrusion detection in CAN bus, 2019. https://dx.doi.org/10.21227/24m9-a446. https://doi.org/10.21227/24m9-a446

[92] A. Taylor, S. Leblanc, N. Japkowicz, Probing the limits of anomaly detectors for automobiles with a cyberattack framework, IEEE Intell. Syst. 33 (2) (2018) 54–62.

[93] M.L. Han, B.I. Kwak, H.K. Kim, Anomaly intrusion detection method for vehicular networks based on survival analysis, Veh. Commun. 14 (2018) 52–63.

[94] H. Kang, B.I. Kwak, Y.H. Lee, H. Lee, H. Lee, H.K. Kim, Car hacking and defense competition on in-vehicle network, in: Workshop on Automotive and Autonomous Vehicle Security (AutoSec), 2021, 2021, p. 25. https://doi.org/10.14722/autosec.2021.23035

[95] G. Dupont, A. Lekidis, J.J. den Hartog, S.S. Etalle, Automotive controller area network (CAN) bus intrusion dataset v2, 2019. https://data.4tu.nl/articles/_/12696950/2. https://doi.org/10.4121/UUID:B74B4928-C377-4585-9432-2004DFA20A5D

[96] Hacking, C.R.L. (HCRL), In-vehicle network intrusion detection challenge, 2019. https://sites.google.com/hksecurity.net/hcrl/Datasets/datachallenge2019/car. https://doi.org/10.4121/UUID:B74B4928-C377-4585-9432-2004DFA20A5D

[97] M. Said Elsayed, N.-A. Le-Khac, S. Dev, A.D. Jurcut, Network anomaly detection using LSTM based autoencoder, in: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 37–45.

[98] A. Vikram, et al., Anomaly detection in network traffic using unsupervised machine learning approach, in: 2020 5th International Conference on Communication and Electronics Systems (ICCES), IEEE, 2020, pp. 476–479. https://doi.org/10.1109/ICCES48766.2020.9137987

[99] B.A. Pratomo, P. Burnap, G. Theodorakopoulos, Unsupervised approach for detecting low rate attacks on network traffic with autoencoder, in: 2018 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), IEEE, 2018, pp. 1–8. https://doi.org/10.1109/CyberSecPODS.2018.8560678

[100] O. Avatefipour, A.S. Al-Sumaiti, A.M. El-Sherbeeny, E.M. Awwad, M.A. Elmeligy, M.A. Mohamed, H. Malik, An intelligent secured framework for cyberattack detection in electric vehicles' CAN bus using machine learning, IEEE Access 7 (2019) 127580–127592.

[101] S. Rajapaksha, H. Kalutarage, M.O. Al-Kadri, G. Madzudzo, A.V. Petrovski, Keep the moving vehicle secure: context-aware intrusion detection system for in-vehicle CAN bus security, in: 2022 14th International Conference on Cyber Conflict: Keep Moving!(CyCon), 700, IEEE, 2022, pp. 309–330.

[102] S. Khandelwal, S. Shreejith, Real-time zero-day intrusion detection system for automotive controller area network on fpgas, in: 2023 IEEE 34th International Conference on Application-specific Systems, Architectures and Processors (ASAP), IEEE, 2023, pp. 139–146.

[103] J. Guidry, F. Sohrab, R. Gottumukkala, S. Katragadda, M. Gabbouj, One-class classification for intrusion detection on vehicular networks, in: 2023 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2023, pp. 1176–1182.

[104] S. Sharmin, H. Mansor, Intrusion detection on the in-vehicle network using machine learning, in: 2021 3rd International Cyber Resilience Conference (CRC), IEEE, 2021, pp. 1–6.

[105] P. Balaji, M. Ghaderi, NeuroCAN: contextual anomaly detection in controller area networks, in: 2021 IEEE International Smart Cities Conference (ISC2), IEEE, 2021, pp. 1–7.

[106] H. Sun, M. Chen, J. Weng, Z. Liu, G. Geng, Anomaly detection for in-vehicle network using CNN-LSTM with attention mechanism, IEEE Trans. Veh. Technol. 70 (10) (2021) 10880–10893.

[107] S.V. Thiruloga, V.K. Kukkala, S. Pasricha, TENET: temporal CNN with attention for anomaly detection in automotive cyber-physical systems, in: 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2022, pp. 326–331.

[108] P. Wei, B. Wang, X. Dai, L. Li, F. He, A novel intrusion detection model for the CAN bus packet of in-vehicle network based on attention mechanism and autoencoder, Digit. Commun. Netw. (2022). https://doi.org/10.1016/j.dcan.2022.04.021

[109] P. Mansourian, N. Zhang, A. Jaekel, M. Zamanirafe, M. Kneppers, Anomaly detection for connected autonomous vehicles using LSTM and Gaussian naïve Bayes, in: International Conference on Wireless and Satellite Systems, Springer, 2023, pp. 31–43.

[110] Y. Zhao, Y. Xun, J. Liu, S. Ma, GVIDS: a reliable vehicle intrusion detection system based on generative adversarial network, in: GLOBECOM 2022-2022 IEEE Global Communications Conference, IEEE, 2022, pp. 4310–4315.

[111] H. Qin, M. Yan, H. Ji, Application of controller area network (CAN) bus anomaly detection based on time series prediction, Veh. Commun. 27 (2021) 100291.

[112] I.A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, A. Jolfaei, An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles, IEEE Trans. Intell. Transp. Syst. 23 (12) (2021) 25469–25478.

[113] E. Kristianto, P.-C. Lin, R.-H. Hwang, Sustainable and lightweight domain-based intrusion detection system for in-vehicle network, Sustain. Comput. Inform. Syst. 41 (2024) 100936.

[114] V. Cobilean, H.S. Mavikumbure, C.S. Wickramasinghe, B.J. Varghese, T. Pennington, M. Manic, Anomaly detection for in-vehicle communication using transformers, in: IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2023, pp. 1–6.

[115] H. Narasimhan, V. Ravi, N. Mohammad, Unsupervised deep learning approach for in-vehicle intrusion detection system, IEEE Consum. Electron. Mag. 12 (1) (2021) 103–108.
[116] J. Wang, X. Mo, A CAN bus anomaly detection based on FLXGBoost algorithm, in: 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys), IEEE, 2021, pp. 1558–1564.
[117] K. Agrawal, T. Alladi, A. Agrawal, V. Chamola, A. Benslimane, NovelADS: a novel anomaly detection system for intra-vehicular networks, IEEE Trans. Intell. Transp. Syst. 23 (11) (2022) 22596–22606.
[118] V.K. Kukkala, S.V. Thiruloga, S. Pasricha, INDRA: intrusion detection using recurrent autoencoders in automotive embedded systems, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 39 (11) (2020) 3698–3710. https://doi.org/10.1109/TCAD.2020.3012749
[119] J. Shi, Z. Xie, L. Dong, X. Jiang, X. Jin, IDS-DEC: a novel intrusion detection for CAN bus traffic based on deep embedded clustering, Veh. Commun. 49 (2024) 100830.
[120] S. Longari, D.H.N. Valcarcel, M. Zago, M. Carminati, S. Zanero, CANnolo: an anomaly detection system based on LSTM autoencoders for controller area network, IEEE Trans. Netw. Serv. Manage. 18 (2) (2020) 1913–1924.
[121] S. Longari, C.A. Pozzoli, A. Nichelini, M. Carminati, S. Zanero, Candito: improving payload-based detection of attacks on controller area networks, in: International Symposium on Cyber Security, Cryptology, and Machine Learning, Springer, 2023, pp. 135–150.
[122] M. Hanselmann, T. Strauss, K. Dormann, H. Ulmer, CANet: an unsupervised intrusion detection system for high dimensional CAN bus data, IEEE Access 8 (2020) 58194–58205.
[123] C.R. Kishore, D.C. Rao, H.S. Behera, Deep learning approach for anamoly detection in CAN bus network: an intelligent LSTM-based intrusion detection system, in: International Conference on Computational Intelligence in Pattern Recognition, Springer, 2022, pp. 531–544.
[124] S. Rajapaksha, H. Kalutarage, M.O. Al-Kadri, A. Petrovski, G. Madzudzo, Beyond vanilla: improved autoencoder-based ensemble in-vehicle intrusion detection system, J. Inform. Secur. Appl. 77 (2023) 103570.
[125] T. Kim, J. Kim, I. You, An anomaly detection method based on multiple LSTM-autoencoder models for in-vehicle network, Electronics 12 (17) (2023) 3543.
[126] H. Jo, D.-H. Kim, Intrusion detection using transformer in con-troller area network, IEEE Access (2024).
[127] J. Xiao, H. Wu, X. Li, Robust and self-evolving IDS for in-vehicle network by enabling spatiotemporal information, in: 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), IEEE, 2019, pp. 1390–1397.
[128] C.R.R. Consortium, Dodge CAN messages. (accessed 10 October 2024), https://www.engr.colostate.edu/~jdaily/tucrrc/DodgeCAN.html.
[129] H.M. Song, H.K. Kim, Discovering can specification using on-board diagnostics, IEEE Des. Test 38 (3) (2020) 93–103.
[130] M. Zago, S. Longari, A. Tricarico, M. Carminati, M.G. Pérez, G.M. Pérez, S. Zanero, Recan–dataset for reverse engineering of controller area networks, Data Brief 29 (2020) 105149.
[131] Q. Zhao, M. Chen, Z. Gu, S. Luan, H. Zeng, S. Chakrabory, CAN bus intrusion detection based on auxiliary classifier GAN and out-of-distribution detection, ACM Trans. Embedded Comput. Syst. (TECS) 21 (4) (2022) 1–30. https://doi.org/10.1145/3540198
[132] J. Zhang, F. Li, H. Zhang, R. Li, Y. Li, Intrusion detection system using deep learning for in-vehicle security, Ad Hoc Netw. 95 (2019) 101974. https://doi.org/10.1016/j.adhoc.2019.101974
[133] L. Yang, A. Moubayed, A. Shami, MTH-IDS: a multitiered hybrid intrusion detection system for internet of vehicles, IEEE Internet Things J. 9 (1) (2022) 616–632. https://doi.org/10.1109/JIOT.2021.3084796
[134] S. Nakamura, K. Takeuchi, H. Kashima, T. Kishikawa, T. Ushio, T. Haga, T. Sasaki, In-vehicle network attack detection across vehicle models: a supervised-unsupervised hybrid approach, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE, 2021, pp. 1286–1291.
[135] A. Rangsikunpum, S. Amiri, L. Ost, BIDS: an efficient intrusion detection system for in-vehicle networks using a two-stage binarised neural network on low-cost FPGA, J. Syst. Archit. 156 (2024) 103285.
[136] M.L. Han, B.I. Kwak, H.K. Kim, Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network, IEEE Trans. Inf. Forensics Secur. 16 (2021) 2941–2956.
[137] E. Gherbi, B. Hanczar, J.-C. Janodet, W. Klaudel, Deep learning for in-vehicle intrusion detection system, in: Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 18–22, 2020, Proceedings, Part IV 27, Springer, 2020, pp. 50–58.
[138] T.-P. Nguyen, J. Cho, D. Kim, Semi-supervised intrusion detection system for in-vehicle networks based on variational autoencoder and adversarial reinforcement learning, Knowl. Based Syst. 304 (2024) 112563.
[139] J. Lin, Y. Wei, W. Li, J. Long, Intrusion detection system based on deep neural network and incremental learning for in-vehicle CAN networks, in: Inernational Conference on Ubiquitous Security, Springer, 2021, pp. 255–267.
[140] G. Kocher, G. Kumar, Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges, Soft Comput. 25 (15) (2021) 9731–9763. https://doi.org/10.1007/s00500-021-05893-0
[141] B. Li, Y. Vorobeychik, Feature cross-substitution in adversarial classification, Adv. Neural Inf. Process. Syst. 27 (2014).
[142] F. Zhang, P.P.K. Chan, B. Biggio, D.S. Yeung, F. Roli, Adversarial feature selection against evasion attacks, IEEE Trans. Cybern. 46 (3) (2015) 766–777. https://doi.org/10.1109/TCYB.2015.2415032
[143] B. Jan, H. Farman, M. Khan, M. Imran, I.U. Islam, A. Ahmad, S. Ali, G. Jeon, Deep learning in big data analytics: a comparative study, Comput. Electr. Eng. 75 (2019) 275–287.
[144] S.T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, Deep transfer learning based intrusion detection system for electric vehicular networks, Sensors 21 (14) (2021) 4736. https://doi.org/10.3390/s21144736
[145] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.
[146] L. Li, Y. Fan, M. Tse, K.-Y. Lin, A review of applications in federated learning, Comput. Ind. Eng. 149 (2020) 106854. https://doi.org/10.1016/j.cie.2020.106854
[147] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P.K.R. Maddikunta, T.R. Gadekallu, Federated learning for intrusion detection system: concepts, challenges and future directions, Comput. Commun. 195 (2022) 346–361. https://doi.org/10.1016/j.comcom.2022.09.012
[148] J. Alsamiri, K. Alsubhi, Federated learning for intrusion detection systems in internet of vehicles: a general taxonomy, applications, and future directions, Future Internet 15 (12) (2023) 403. https://doi.org/10.3390/fi15120403
[149] X. 1375 Working Group, Guidelines for an Intrusion Detection System for in-Vehicle Networks, Technical Report X.1375, International Telecommunication Union, 2020.
[150] M. Althunayyan, A. Javed, O. Rana, T. Spyridopoulos, Hierarchical federated learning-based intrusion detection for in-vehicle networks, Future Internet 16 (12) (2024) 451.
[151] M. Driss, I. Almomani, Z. e Huma, J. Ahmad, A federated learning framework for cyberattack detection in vehicular sensor networks, Complex Intell. Syst. 8 (5) (2022) 4221–4235. https://doi.org/10.1007/s40747-022-00705-w
[152] K.H. Shibly, M.D. Hossain, H. Inoue, Y. Taenaka, Y. Kadobayashi, Personalized federated learning for automotive intrusion detection systems, in: 2022 IEEE Future Networks World Forum (FNWF), IEEE, 2022, pp. 544–549. https://doi.org/10.1109/FNWF55208.2022.00101
[153] T. Yu, G. Hua, H. Wang, J. Yang, J. Hu, Federated-LSTM based network intrusion detection method for intelligent connected vehicles, IEEE Int. Conf. Commun. (ICC) (2022). https://doi.org/10.1109/ICC45855.2022.9838655
[154] H. Zhang, K. Zeng, S. Lin, Federated graph neural network for fast anomaly detection in controller area networks, IEEE Trans. Inf. Forensics Secur. 18 (2023) 1566–1579. https://doi.org/10.1109/TIFS.2023.3240291
[155] J. Yang, J. Hu, T. Yu, Federated AI-enabled in-vehicle network intrusion detection for internet of vehicles, Electronics (2022). https://doi.org/electronics11223658
[156] H. Taslimasa, S. Dadkhah, E.P. Neto, P. Xiong, S. Iqbal, S. Ray, A. Ghorbani, ImageFed: practical privacy preserving intrusion detection system for in-vehicle CAN bus protocol, IEEE BigDataSecurity (2023). https://doi.org/10.1109/BigDataSecurity-HPSC-IDS58521.2023.00031
[157] T.-N. Hoang, M.R. Islam, K. Yim, D. Kim, CANPerFL: improve in-vehicle intrusion detection performance by sharing knowledge, Appl. Sci. 13 (11) (2023) 6369.
[158] M. Marchetti, D. Stabili, READ: reverse engineering of automotive data frames, IEEE Trans. Inf. Forensics Secur. 14 (4) (2018) 1083–1097. https://doi.org/10.1109/TIFS.2018.2870826
[159] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282. https://doi.org/10.48550/arXiv.1602.05629
[160] Q. Li, Y. Diao, Q. Chen, B. He, Federated learning on non-iid data silos: An experimental study, in: 2022 IEEE 38th International Conference on Data Engineering (ICDE), IEEE, 2022, pp. 965–978. https://doi.org/10.48550/arXiv.2102.02079
[161] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, V. Chandra, Federated learning with non-iid data, arXiv preprint arXiv:1806.00582 (2018). https://doi.org/10.48550/arXiv.1806.00582
[162] Y. Li, J. Lin, K. Xiong, An adversarial attack defending system for securing in-vehicle networks, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2021, pp. 1–6.
[163] S. Axelsson, D. Sands, Understanding Intrusion Detection Through Visualization, 24, Springer Science & Business Media, 2006.
[164] H. Lundberg, N.I. Mowla, S.F. Abedin, K. Thar, A. Mahmood, M. Gidlund, S. Raza, Experimental analysis of trustworthy in-vehicle intrusion detection system using explainable artificial intelligence (XAI), IEEE Access 10 (2022) 102831–102841.
[165] M.Y. Abualhoul, O. Shagdar, F. Nashashibi, Visible light inter-vehicle communication for platooning of autonomous vehicles, in: 2016 IEEE Intelligent Vehicles Symposium (IV), IEEE, 2016, pp. 508–513. https://doi.org/10.1109/IVS.2016.7535434
[166] A. Moubayed, A. Shami, P. Heidari, A. Larabi, R. Brunner, Edge-enabled V2X service placement for intelligent transportation systems, IEEE Trans. Mob. Comput. 20 (4) (2020) 1380–1392.