


A SURVEY OF LEARNING-BASED INTRUSION DETECTION SYSTEMS FOR IN-VEHICLE NETWORK


A PREPRINT

 **Muzun Althunayyan**

School of Computer Science & Informatics
Cardiff University
Cardiff, United Kingdom
AlthunayyanMS@cardiff.ac.uk

 **Amir Javed**

School of Computer Science & Informatics
Cardiff University
Cardiff, United Kingdom
javeda7@cardiff.ac.uk

 **Omer Rana**

School of Computer Science & Informatics
Cardiff University
Cardiff, United Kingdom
ranaof@cardiff.ac.uk

May 20, 2025

ABSTRACT

Connected and Autonomous Vehicles (CAVs) enhance mobility but face 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.

Keywords CAN bus · Cyberattack · Intrusion Detection System · Anomaly Detection · Machine Learning · In-vehicle Network.

1 Introduction

Connected and Autonomous Vehicles (CAVs) are expected to become the backbone of future transportation systems [Aloraini et al.(2024)Aloraini, Javed, and Rana], offering the potential to not only revolutionise mobility, but also deliver significant economic benefits. For example, the Society of Motor Manufacturers and Traders (SMMT) estimates that this technological shift could provide the United Kingdom with an annual economic boost of £62 billion by 2030 [SMMT Driving the Motor Industry(2019)]. However, these advancements also present significant security challenges for CAVs, making them attractive targets for emerging cyber threats [Pickford et al.(2024)Pickford, Attale, Shaikh, Nguyen, and Harrison].

CAVs rely on Electronic Control Units (ECUs) to manage and control various functions. These ECUs communicate through standardised in-vehicle communication protocols, such as the Controller Area Network (CAN), FlexRay, Local Interconnect Network (LIN) and Media Oriented System Transport (MOST). Among

these, the CAN bus is the protocol that is the most widely used, valued for its high speed, reliability, and ease of use [Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis]. Although originally designed for industrial applications, the CAN bus has become the de facto standard for in-vehicle communication [Althunayyan et al.(2024a)Althunayyan, Javed, and Rana]. Despite its advantages, the CAN protocol was not designed with security in mind and lacks essential features such as sender authentication and encryption [Paul and Islam(2021)].

The increasing interconnectivity of CAVs exposes them to a range of cyberattacks. The attack surfaces in modern vehicles can be accessed either physically, via ports such as the USB or the onboard diagnostic (OBD)-II port, or remotely through wireless technologies such as Bluetooth, Wi-Fi, and LTE [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap]. 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 [Ltd.(2024)]. These vulnerabilities make vehicles susceptible to attacks that could have devastating consequences, including loss of control over critical systems like braking, steering, and acceleration [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom]. A recent incident [Tindell(2023)] 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. This manipulation enabled the attackers to unlock the doors and start the engine, ultimately stealing the vehicle without the need for a key. Moreover, a notorious example is the Jeep hack, where attackers remotely gained control over the vehicle's braking and steering systems, resulting in dangerous driving conditions [Golson(2016)]. Similarly, vulnerabilities in BMW and Toyota Lexus models have been exploited, demonstrating the persistent threat to vehicle security [Lab(2018), Lab(2020)]. Such incidents underscore the need for robust security measures to protect against both information theft and direct physical harm.

Given the severity of these threats, the security of the CAN bus has become a major area of research. According to McKinsey's analysis, by 2030, almost 95% of the new vehicles will be connected to external networks, further highlighting the need for effective security solutions [Bertoncello et al.(2021)Bertoncello, Martens, Möller, and Schneiderbauer]. 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 [Hoppe et al.(2009)Hoppe, Kiltz, and Dittmann]. Anomaly-based IDSs are further categorised into statistical, Machine Learning (ML), rule-based, and physical fingerprinting methods [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah].

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 ensure the safety and reliability 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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. 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 existing work.

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 present a systematic categorization of CAN protocol vulnerabilities by conducting an in-depth analysis of the CAN protocol, identifying its weaknesses, entry points, and potential attack scenarios.

- 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 in connected vehicle environments.
- We outline key open challenges and propose future research directions to enhance the development of more effective, efficient, and resilient in-vehicle IDS solutions.

The remainder of this paper is organised as follows. Section 2 provides context 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 discusses future 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 is the internal network that facilitates communication between multiple ECUs within a vehicle [Liu et al.(2020)Liu, Zhang, Song, and Letaief]. These ECUs are interconnected embedded devices responsible for controlling various vehicle functions, 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 [Ahmad et al.(2024)Ahmad, Han, Jolfaei, Jabbar, Ibrar, Erbad, Song, and Alkhrijah]. These ECUs, along with sensors, actuators, cameras, radars, and communication devices, collaborate to enhance vehicle performance, efficiency, intelligent services, and safety by collecting and interpreting various data [Boumiza and Braham(2017)]. ECUs communicate using standard protocols such as CAN, FlexRay, LIN, and MOST [Kumar and Ramesh(2014)]. Among these, CAN is considered the de facto protocol for in-vehicle communication [Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis].

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 [Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis]. 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 CAN frame follows a specific message structure defined in a database-like file known as the DataBase CAN (DBC) file. This file is confidential and proprietary to the vehicle manufacturer, containing all the essential information about the representation of the CAN bus data [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar]. The CAN data frame consists of seven fields that facilitate data transmission between ECUs. Figure 1 illustrates the standard CAN frame format, which consists of the following fields:

- **Start of Frame (SOF):** The purpose of this field is to synchronise the transmission of the CAN message with all nodes and to signal the initiation of its transmission.
- **Arbitration Field (ID):** This field, also known as the CAN ID, is used to specify the destination address of the designated ECU. It also determines the priority of the message, where a lower value generally indicates a higher priority. The ID field is 11 bits in size.
- **Data Length Code (DLC):** This field provides information about the length of the data field.

- **Data Field:** This field, also known as the payload, includes the actual vehicle parameter values, which are interpreted by the received ECU and its size can vary from 0 to 8 bytes (0-64 bits).
- **Cyclic Redundancy Check (CRC):** This field detects errors and maintains data integrity during message transmission with a fixed size of 16 bits.
- **Acknowledge Field (ACK):** This field receives confirmation from the receiving node that the CAN message was received correctly.
- **End of Frame (EOF):** This field signifies the completion of CAN message transmission.

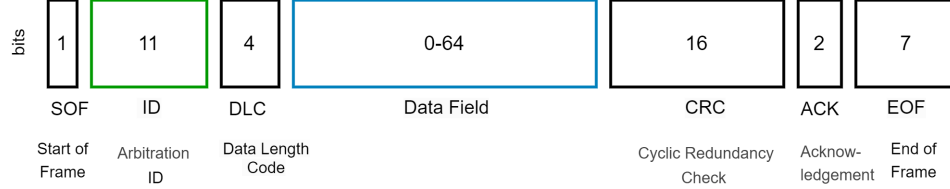


Figure 1: CAN data frame

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 [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap, Carsten et al.(2015)Carsten, Andel, Yampolskiy, and McDonald, Liu et al.(2017)Liu, Zhang, Sun, and Shi], 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 [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom]. 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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah].
- **Lack of encryption:** Due to time constraints, CAN messages are not encrypted [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap], 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 [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom].
- **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 [Ahmad et al.(2024)Ahmad, Han, Jolfaei, Jabbar, Ibrar, Erbad, Song, and Alkhrijah]. If an ECU is compromised, it can intercept and monitor all messages transmitted across the CAN network, enabling an eavesdropping attack [Dupont et al.(2019a)Dupont, den Hartog, Etalle, and Lekidis].
- **ID-based priority:** The CAN network prioritises messages based on their IDs, with lower IDs having higher priority [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. Attackers can exploit this by repeatedly sending frames with low IDs, resulting in a Denial-of-Service (DoS) attack [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar].
- **Unsegmented Network:** All ECUs are connected to a single shared network without segmentation [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom], which is a key reason why CAN was adopted in automotive systems, as it eliminates the need for point-to-point connections. However, this shared network allows components such as infotainment systems to communicate with safety-critical vehicle systems. Although some manufacturers use separate networks for safety-critical systems, there is still cross-communication between critical and non-critical systems [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom].
- **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 [Wu et al.(2019)Wu, Li, Xie, An, Bai, Zhou, and Li, Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah].

2.5 In-vehicle Network Entry Points

Attackers can gain access to the CAN bus or specific ECUs either physically or remotely [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap, Limbasiya et al.(2022)Limbasiya, Teng, Chattopadhyay, and Zhou, Checkoway et al.(2011)Checkoway, McCoy, Kantor, Anderson, Shacham, Savage, Koscher, Czeskis, Roesner, and Kohno, Carsten et al.(2015)Carsten, Anel, Yampolskiy, and McDonald]. These entry points serve as gateways for initiating a range of attacks, exploiting the inherent vulnerabilities described in Section 2.4 in 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 [Checkoway et al.(2011)Checkoway, McCoy, Kantor, Anderson, Shacham, Savage, Koscher, Czeskis, Roesner, and Kohno]. 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 [Koscher et al.(2010)Koscher, Czeskis, Roesner, Patel, Kohno, Checkoway, McCoy, Kantor, Anderson, Shacham, et al.]. 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 [Carsten et al.(2015)Carsten, Anel, Yampolskiy, and McDonald].
- **Aftermarket Components:** Peripheral components such as USB ports, CD players, and third-party add-ons also pose security risks [Liu et al.(2017)Liu, Zhang, Sun, and Shi, Carsten et al.(2015)Carsten, Anel, Yampolskiy, and McDonald]. 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 [Koscher et al.(2010)Koscher, Czeskis, Roesner, Patel, Kohno, Checkoway, McCoy, Kantor, Anderson, Shacham, et al.].

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 inject malicious data or commands into the CAN bus [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap]. Koscher et al. [Chockalingam et al.(2016)Chockalingam, Larson, Lin, and Nofzinger] highlight the feasibility of executing various types of wireless attack injections on in-vehicle network systems. 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) [Chockalingam et al.(2016)Chockalingam, Larson, Lin, and Nofzinger]. Moreover, Woo et al. [Woo et al.(2014)Woo, Jo, and Lee] 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.

2.6 Attack Scenarios

Since an attacker can gain access to 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:** The goal of a DoS attack is to overwhelm the CAN bus bandwidth by transmitting large volumes of messages, leading to system malfunctions and service disruptions

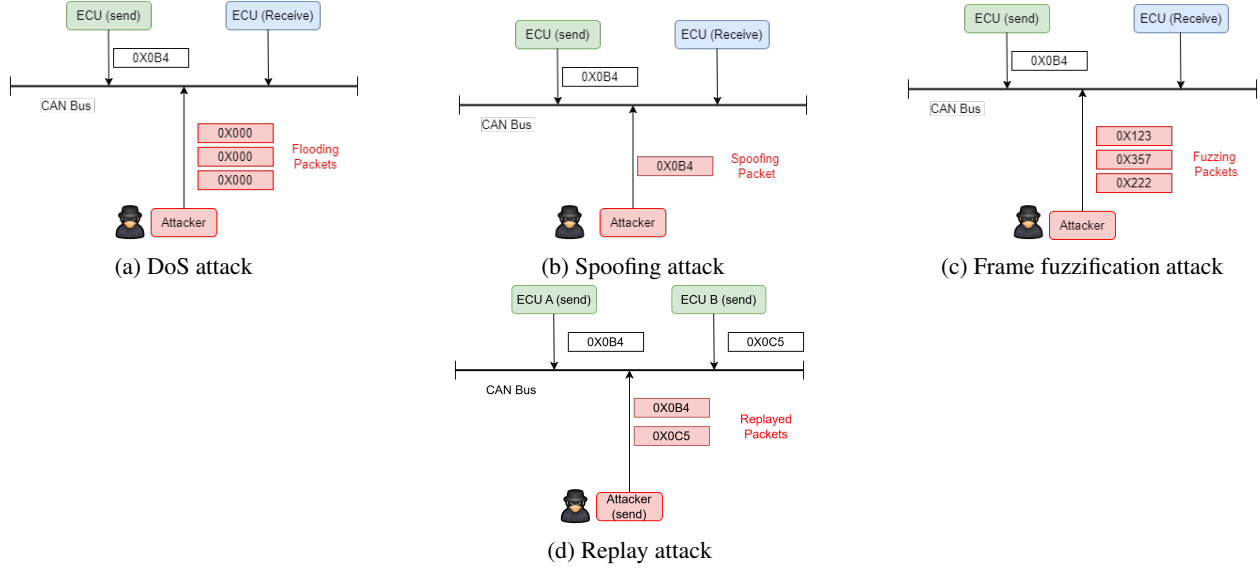


Figure 2: CAN bus attacks

[Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. Koscher et al. [Koscher et al.(2010)Koscher, Czeskis, Roesner, Patel, Kohno, Checkoway, McCoy, Kantor, Anderson, Shacham, et al.] demonstrated that DoS attacks can disable individual CAN bus components.

- Attack Method:** Since message priority is determined by the arbitration field, an attacker can exploit the CAN frame priority arbitration scheme vulnerability by sending numerous messages with low CAN IDs (high priority), such as 0x0000. This flooding of high-priority frames occupies the bus, preventing other ECUs from accessing it [Hoang and Kim(2022)]. Figure 2a illustrates how a high-priority CAN ID 0x0000 delays a lower-priority CAN ID 0x0B4.
- Attack Scenario:** We assume that 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. Leveraging this access, the attacker floods the CAN bus with high-priority messages, such as those with CAN IDs like 0x0000, 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 [Lee et al.(2017)Lee, Jeong, and Kim], but also prevents other ECUs from transmitting frames to the in-vehicle network, significantly impacting network availability [Cho and Shin(2016)]. Such attacks can lead to a complete breakdown of ECU communication and severe disruption of the entire CAN bus network system [Hossain et al.(2020a)Hossain, Inoue, Ochiai, Fall, and Kadobayashi, Liu et al.(2017)Liu, Zhang, Sun, and Shi], posing significant threats to the safety of drivers, passengers, and other road users [Fowler et al.(2018)Fowler, Bryans, Shaikh, and Wooderson].

2.6.2 Spoofing Attack

- Attack Definition:** In a spoofing attack, an unauthorised attacker targets specific existing CAN IDs and injects fabricated messages to control particular functions. Since CAN IDs appear legitimate, distinguishing between real and spoofed messages becomes challenging, leading to system malfunctions [Hoang and Kim(2022)].
- Attack Method:** The attacker may sniff the CAN bus traffic or possess prior knowledge about the ECUs' CAN messages. They can then use this information to inject spoofed messages into the CAN bus. Figure 2b illustrates a spoofing attack where an attacker, using the spoofed CAN ID 0x0B4, targets the legitimate CAN

ID 0x0B4. This enables the attacker to disrupt vehicle functions by generating manipulated messages that appear legitimate.

- **Attack Scenario:** We assume that 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. In this attack, we assume that the attacker has some knowledge of CAN bus traffic by implementing an impersonation attack. 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, with plans to disable it from the bus later. 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. This activity exploits the CAN bus protocol's error-handling mechanisms, as described in [Iehira et al.(2018)Iehira, Inoue, and Ishida]. By transmitting dominant bits whenever the legitimate ECU sends recessive bits, the attacker creates intentional bit conflicts. These conflicts generate repeated error frames, eventually causing the legitimate ECU to exceed its error tolerance threshold and enter a "bus-off" state. In the bus-off state, the legitimate ECU is effectively disconnected from the network, unable to send or receive critical messages. The attacker then injects maliciously crafted messages using the same CAN ID as the target ECU. For instance, they could dangerously slow the vehicle on a highway or increase its speed in restricted areas, thereby creating hazardous conditions.
- **Attack Impact:** Spoofing attacks can cause system malfunctions and disrupt vehicle operations [Hoang and Kim(2022)]. They pose significant threats to personal safety, particularly when targeting critical ECUs responsible for essential functions such as braking or steering [Iehira et al.(2018)Iehira, Inoue, and Ishida].

2.6.3 Frame Fuzzification Attack

- **Attack Definition:** 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 exploit prior knowledge of CAN IDs and payload values obtained through CAN bus sniffing, or they may perform the attack without any prior knowledge of CAN frames, treating it as a black-box attack [Hossain et al.(2020b)Hossain, Inoue, Ochiai, Fall, and Kadobayashi]. In this type of attack, the attacker might alter the CAN ID, the CAN payload, or both simultaneously [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap]. Since the range of valid CAN packets is relatively small, even simple fuzzing of packets can cause significant damage [Koscher et al.(2010)Koscher, Czeskis, Roesner, Patel, Kohno, Checkoway, McCoy, Kantor, Anderson, Shacham, et al.].
- **Attack Method:** The attacker sends arbitrary messages into the CAN bus network, making them appear as legitimate traffic. Figure 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, leading to unintended vehicle behaviours [Lee et al.(2017)Lee, Jeong, and Kim]. For example, Chockalingam et al. [Chockalingam et al.(2016)Chockalingam, Larson, Lin, and Nofzinger] introduced Gaussian noise to create a frame fuzzification attack on CAN data. A frame fuzzification attack can disrupt the entire CAN bus network, leading to severe malfunctions such as the steering wheel shaking uncontrollably, signal lights flickering erratically, or automatic and unintended changes in the gear shift [Hossain et al.(2020a)Hossain, Inoue, Ochiai, Fall, and Kadobayashi].
- **Attack Scenario:** We assume that 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 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 [Lee et al.(2017)Lee, Jeong, and Kim]. 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:** In replay attacks, attackers store a valid message at a certain time and replay it at a different time without any changes [Jo and Choi(2021)]. For example, an attacker can store a speedometer reading and later rebroadcast it to the network [Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis]. In this type of attack, the attacker might alter the ID, payload sequences, or both [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah].
- **Attack Method:** Attackers first observe and capture valid CAN messages while monitoring the vehicle's CAN bus. Later, they replay these captured messages without modification to manipulate the system [Jo and Choi(2021)]. Consequently, the normal CAN packet in the traffic is replaced with a previously captured packet, causing the historical effects [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar]. Figure 2d illustrates a replay attack, where the attacker sniffs the legitimate CAN messages (e.g., 0X0B4 and 0X0C5), and after some time, he reinjects the same messages into the network.
- **Attack Scenario:** We assume that 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. In this attack, we assume that the attacker has no prior knowledge of CAN bus traffic but is able to capture legitimate CAN traffic. One method to achieve this is by connecting a malicious device to sniff the CAN bus traffic, capturing data transmitted across the network. After some time, the attacker gains access to the network and replays the sniffed messages back into the traffic.
- **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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. Koscher et al. [Koscher et al.(2010)Koscher, Czeskis, Roesner, Patel, Kohno, Checkoway, McCoy, Kantor, Anderson, Shacham, et al.] 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 replaced 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 [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar].

3 Related Work

In this section, we review existing surveys and reviews on IDS approaches in in-vehicle networks, highlighting their contributions and how our survey differs. There are several surveys in this field. Most of them review the security of in-vehicle networks in general and include IDS as one of the approaches, alongside ML and DL [Karopoulos et al.(2022)Karopoulos, Kambourakis, Chatzoglou, Hernández-Ramos, and Kouliaridis, Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap, Dupont et al.(2019a)Dupont, den Hartog, Etalle, and Lekidis, Tomlinson et al.(2018)Tomlinson, Bryans, and Shaikh, Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis, Wu et al.(2019)Wu, Li, Xie, An, Bai, Zhou, and Li, Rajbahadur et al.(2018)Rajbahadur, Malton, Walenstein, and Hassan, Jo and Choi(2021), Lokman et al.(2019)Lokman, Othman, and Abu-Bakar, Loukas et al.(2019)Loukas, Karapistoli, Panaousis, Sarigian, Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom, Quadar et al.(2024)Quadar, Chehri, Debaque, Ahmed, and Jeon, Lampe and Meng(2023a), Limbasiya et al.(2022)Limbasiya, Teng, Chattopadhyay, and Zhou]. Others focus specifically on reviewing ML and DL techniques [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah, Nagarajan et al.(2023)Nagarajan, Mansourian, Shahid, Jaekel, Saini, Zhang, and Kneppers, Lampe and Meng(2023b), Almehdhar et al.(2024)Almehdhar, Albaseer, Khan, Abdallah, Menouar, Al-Kuwari, and Al-Fuqaha, Taslimasa et al.(2023a)Taslimasa, Dadkhah, Neto, Xiong, Ray, and Ghorbani].

Rajapaksha et al.[Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah] reviewed and classified state-of-the-art AI-based in-vehicle IDSs, collecting papers using the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) protocol. They proposed an AI-based IDS taxonomy, reviewed benchmark datasets, and outlined the steps for developing AI-based attack detection in the CAN bus. Additionally, they identified and discussed the limitations of current approaches for securing in-vehicle networks and suggested possible future research directions.

Karopoulos et al.[Karopoulos et al.(2022)Karopoulos, Kambourakis, Chatzoglou, Hernández-Ramos, and Kouliaridis] 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. They reviewed available datasets for training and testing in-vehicle IDSs, along with simulators used for dataset generation or performance evaluation. Additionally, they highlighted the main challenges and future directions in this rapidly advancing field.

Aliwa et al.[Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap] reviewed cryptographic and IDS solutions to protect vehicular data and discussed their challenges.

Dupont et al.[Dupont et al.(2019a)Dupont, den Hartog, Etalle, and Lekidis] 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.[Tomlinson et al.(2018)Tomlinson, Bryans, and Shaikh] reviewed methods for CAN IDSs and categorised them into signature detection and anomaly detection. The anomaly detection category was further divided into statistical, knowledge-based, and ML approaches, highlighting their implications.

Al-Jarrah et al.[Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis] provide an overview of intra-vehicle IDSs, categorizing them into flow-based, payload-based, and hybrid types. They discuss and identify the challenges and current gaps in the landscape of intra-vehicle IDS research. Out of the 42 reviewed papers, only 23 are ML-based IDSs.

Wu et al.[Wu et al.(2019)Wu, Li, Xie, An, Bai, Zhou, and Li] categorised in-vehicle IDSs into four types: fingerprint-based, parameter monitoring-based, information theory-based, and ML-based. Moreover, they discussed the drawbacks and emerging research directions. However, out of the 20 reviewed papers, only 9 focused on ML-based IDSs.

Rajbahadur et al.[Rajbahadur et al.(2018)Rajbahadur, Malton, Walenstein, and Hassan] conducted a survey on anomaly detection for securing CAVs. They introduced a taxonomy with three main categories and nine subcategories. In addition, they classified the surveyed papers into 38 dimensions. While the study provided valuable insights, it lacks individual paper summaries and practical implementation strategies.

Jo et al.[Jo and Choi(2021)] classified in-vehicle countermeasures into four categories: preventative protection, IDSs, authentication, and post-protection. They further divided IDS techniques into CAN packet-based and ECU hardware characteristic-based approaches. Although this survey comprehensively examines CAN attack surfaces and corresponding protection mechanisms, it does not focus on ML-based IDSs.

Lokman et al.[Lokman et al.(2019)Lokman, Othman, and Abu-Bakar] introduced a taxonomy for classifying research papers according to four aspects: deployment strategies, attacking techniques, technical challenges, and detection approaches. They also categorised anomaly-based IDSs into frequency-based, ML-based, statistical-based, and hybrid-based. Despite its contribution, only a limited number (five) of works were discussed under the ML-based approach.

Loukas et al.[Loukas et al.(2019)Loukas, Karapistoli, Panaousis, Sarigiannidis, Bezemskij, and Vuong] provided a comprehensive taxonomy focusing on IDS characteristics and architectures designed for different types of vehicles, such as aircraft, land vehicles, and watercraft. They classified audit techniques into statistical, ML, and rule-based IDSs, and only 13 ML-based IDSs for the CAN bus are discussed.

Young et al.[Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom] categorised CAN bus IDSs into signature-based and anomaly-based methods. However, the reviewed ML-based IDSs are limited.

Lampe and Meng [Lampe and Meng(2023b)] provided a comprehensive overview of DL-based IDSs in automotive networks, categorizing them based on their topologies and techniques, such as DNN-based IDSs, CNN-based IDSs, LSTM-based IDSs, attention- and transformer-based IDSs, and GAN-based IDSs. They also discuss the advantages and disadvantages of each approach.

Nagarajan et al.[Nagarajan et al.(2023)Nagarajan, Mansourian, Shahid, Jaekel, Saini, Zhang, and Kneppers] presented a comprehensive review of ML-based IDSs for in-vehicle and inter-vehicle communications. They reviewed available datasets, summarised current testbeds, and discussed open research issues.

Taslimasa et al.[Taslimasa et al.(2023a)Taslimasa, Dadkhah, Neto, Xiong, Ray, and Ghorbani] provided a comprehensive literature review of proposed IDSs for Internet of Vehicles (IoV) networks (inter-vehicle and intra-vehicle) that utilise ML and DL algorithms. Additionally, they discussed IDS criteria, highlighting key factors to consider when assessing IoV network security.

Quadar et al.[Quadar et al.(2024)Quadar, Chehri, Debaque, Ahmed, and Jeon] reviewed and classified in-vehicle IDSs based on detection methods, including fingerprint-based methods, time- and frequency-based methods, and ML-based methods. However, the reviewed ML-based IDSs are limited.

Lampe and Meng [Lampe and Meng(2023a)] reviewed automotive intrusion detection methodologies, categorising IDSs into non-learning, traditional ML, and DL. They further classified IDSs by six dimensions: analysis, deployment, detection, evaluation, learning, and monitoring modes. Open challenges and future research opportunities were also discussed.

Limbasiya et al. [Limbasiya et al.(2022)Limbasiya, Teng, Chattopadhyay, and Zhou] present a systematic survey that extensively analyses different Attack Detection and Prevention System (ADPS) categories for CAVs. They discuss state-of-the-art research in each category, highlighting the latest findings in this domain. Additionally, they identify crucial open security challenges that must be addressed for the secure deployment of CAVs.

Almehdhar et al. [Almehdhar et al.(2024)Almehdhar, Albaseer, Khan, Abdallah, Menouar, Al-Kuwari, and Al-Fuqaha] categorised IDS techniques into conventional ML, DL, and hybrid models. They also explored emerging technologies

Reference	Year	Search Strategy	ML-DL Specific	FL-based IDSs	Evaluation metrics
[Tomlinson et al.(2018)Tomlinson, Bryans, and Shaikh]	2018		○		
[Rajbahadur et al.(2018)Rajbahadur, Malton, Walenstein, and Hassan]	2018	●	○		
[Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis]	2019		○		●
[Wu et al.(2019)Wu, Li, Xie, An, Bai, Zhou, and Li]	2019		○		
[Loukas et al.(2019)Loukas, Karapistoli, Panaousis, Sarigiannidis, Bezemskij, and Vuong]	2019		○		
[Lokman et al.(2019)Lokman, Othman, and Abu-Bakar]	2019		○		
[Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom]	2019		○		
[Dupont et al.(2019a)Dupont, den Hartog, Etalle, and Lekidis]	2019		○		
[Jo and Choi(2021)]	2021		○		
[Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap]	2021		○		
[Karopoulos et al.(2022)Karopoulos, Kambourakis, Chatzoglou, Hernández-Ramos, and Kouliaridis]	2022		○		
[Limbasiya et al.(2022)Limbasiya, Teng, Chattopadhyay, and Zhou]	2022	●	○		
[Taslimasa et al.(2023a)Taslimasa, Dadkhan, Neto, Xiong, Ray, and Ghorbani]	2023		●		●
[Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]	2023	●	●		○
[Lampe and Meng(2023b)]	2023		●		○
[Nagarajan et al.(2023)Nagarajan, Mansourian, Shahid, Jaekel, Saini, Zhang, and Kneppers]	2023		●		○
[Lampe and Meng(2023a)]	2023		○		○
[Quadar et al.(2024)Quadar, Chehri, Debaque, Ahmed, and Jeon]	2024		○		
[Almehdhar et al.(2024)Almehdhar, Albaseer, Khan, Abdallah, Menouar, Al-Kuwari, and Al-Fuqaha]	2024		●	○	○
Our Survey	2025	●	●	●	●

●: Extensive, ○: Partial

Table 1: Comparison with In-Vehicle IDS Surveys

such as FL and Transfer Learning. Key limitations in current methodologies and potential directions for future research were identified.

Even though existing in-vehicle surveys and reviews 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 [Almehdhar et al.(2024)Almehdhar, Albaseer, Khan, Abdallah, Menouar, Al-Kuwari, and Al-Fuqaha], which briefly presents some studies in this area. Although Chellapandi et al. [Chellapandi et al.(2023)Chellapandi, Yuan, Žak, and Wang] provide a survey on FL for CAVs, they do not include any in-vehicle IDSs and instead focus on FL applications such as steering wheel angle prediction, vehicle trajectory prediction, object detection, motion control, and driver monitoring. 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 other existing surveys on in-vehicle IDSs, highlighting the key contributions of this work.

4 Methodology

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

4.1 Search Strategy

To select studies for inclusion, we followed Kitchenham’s [Kitchenham and Brereton(2013)] method, a well-established and effective guide for identifying relevant literature. Although originally designed for the software engineering domain, this method has been widely applied in other fields, including cybersecurity [Alhirabi et al.(2021)Alhirabi, Rana, and Perera]. Our process began with an automatic search using Google Scholar to minimise bias towards any specific publisher [Wohlin(2014)] and to identify key publishers and conferences in the field. Based on this search, we compile a list of publishers and conferences for a subsequent manual search. To ensure comprehensive coverage, we employ a snowball approach to locate all related papers. After gathering a substantial number of publications, we applied filtering processes to select the most relevant ones. Finally, we analysed the collected articles (from any time up to and including January 2025) to develop the literature review presented in this paper. Figure 3 shows the search strategy adopted.

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. These terms are listed in Table 2. 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. This step also provided insights into the digital libraries and journals that prioritise CAN bus security. Subsequently, we conducted a hybrid search using more complex queries in specific journals and libraries, including IEEE Xplore, Scopus, ACM, MDPI, Springer, and ScienceDirect. Three search strategies were employed during this process: automatic, manual, and snowballing.

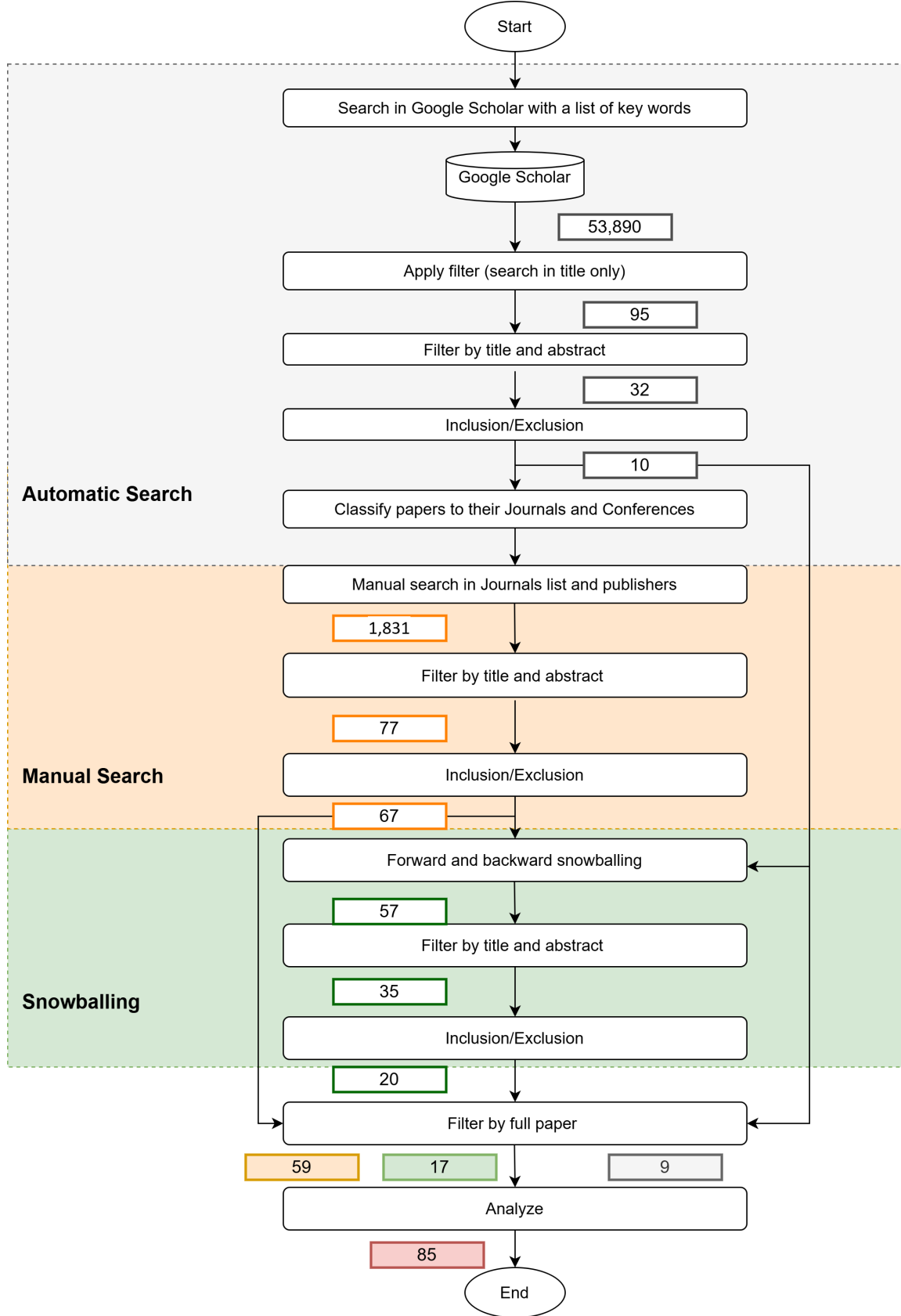


Figure 3: Search and selection processes flowchart

1. **Automatic search** We carried out this stage using the advanced search function in Google Scholar, employing key terms such as "CAN bus", "controller area network", "in-vehicle", "intrusion detection system", "IDS", "anomaly detection", "unknown attacks", and "federated". These root words were chosen because Google Scholar automatically searches for variations of the same word; for instance, searching for "federated" returns results like "federated", "federated learning", "federated-based", and "federated environment". Additionally, the search was not restricted to a specific time period. Initially, using the filter "anywhere in the article," we retrieved an unwieldy number of results (53,890) (see Table 2). 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. Table 3 lists examples of queries used in the manual search.
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. We found that backward snowballing using critical papers was an effective means of identifying relevant papers. The set of articles selected was updated to include any additional relevant studies found by snowballing.

Key Terms	Anywhere in the Article	In the Title
("CAN bus" OR "Controller Area Network") AND ("IDS" OR "intrusion detection system")	9,220	10
"In-vehicle" AND ("IDS" OR "Intrusion Detection System")	20,000	25
("CAN bus" OR "Controller Area Network") AND "anomaly detection"	4,690	11
"In-vehicle" AND "anomaly detection"	9,970	39
("CAN bus" OR "Controller Area Network") AND "unknown attacks"	658	0
"In-vehicle" AND "unknown attacks"	892	0
("CAN bus" OR "Controller Area Network") AND "federated"	1,860	0
"In-vehicle" AND "federated"	6,600	10
Total	53,890	95

Table 2: Google Scholar search terms and results

4.1.2 Selection Strategy

The selection strategy involved defining inclusion and exclusion criteria, as well as applying filtering during the search process. The steps in the selection process included applying the inclusion and exclusion criteria, followed by an additional filtering stage involving a quality assessment to ensure the selection of high-quality studies. Each of these steps is discussed below.

- **Filtering Irrelevant Papers:** The papers collected through manual, automatic, and snowballing approaches included several that did not apply directly to our study and had to be eliminated. Elimination was conducted in two steps. In the first, papers were excluded based on the title, keywords, abstract, and sometimes the conclusion in case of any doubts. Based on this step, a decision was made regarding whether to include each paper in the next step. Eliminating was undertaken after each search (automatic, manual, and snowballing)

Category	Queries and Terms
General	"CAN bus" OR "controller area network" OR "in-vehicle" AND "intrusion detection system" OR "IDS" OR "anomaly detection" OR "unknown attacks" OR "federated".
	ACM: [[Title: "can bus"] OR [Title: "controller area network"] OR [Title: "in-vehicle"]] AND [[Title: "intrusion detection system"] OR [Title: "ids"] OR [Title: "anomaly detection"] OR [Title: "unknown attacks"] OR [Title: "federated"]]
More Specific	IEEE Xplore: ("Document Title": "CAN bus") OR ("Document Title": "controller area network") OR ("Document Title": "in-vehicle") AND ("Document Title": "intrusion detection system") OR ("Document Title": "IDS") OR ("Document Title": "anomaly detection") OR ("Document Title": "unknown attacks") OR ("Document Title": "federated")
	Scopus: (TITLE-ABS-KEY ("CAN bus" OR "controller area network" OR "in-vehicle") AND TITLE-ABS-KEY ("intrusion detection system" OR "IDS" OR "anomaly detection" OR "unknown attacks" OR "federated"))

Table 3: Examples of queries and terms used for the online library search

to reduce the number of papers. When a paper passed the initial elimination step, it was subjected to the inclusion and exclusion criteria.

- **Inclusion and Exclusion Criteria** In this process, we defined our exclusion and inclusion criteria. A paper was considered for exclusion if it met one or more criteria. 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 Figure 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. In the manual search within journal lists and publishers, the initial 1,831 papers were filtered down to 59. For snowballing, 57 papers were initially selected and reduced through filters to 17. Thus, the total number of collected papers comprises 9 from the automatic search, 59 from the manual search, and 17 from snowballing, resulting in a total of 85 papers for analysis. Of these, 38 focused on known attack detection, 27 on unknown attack detection, 11 on IDSs capable of detecting both known and unknown attacks, and 9 on FL-based IDSs. Figure 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.

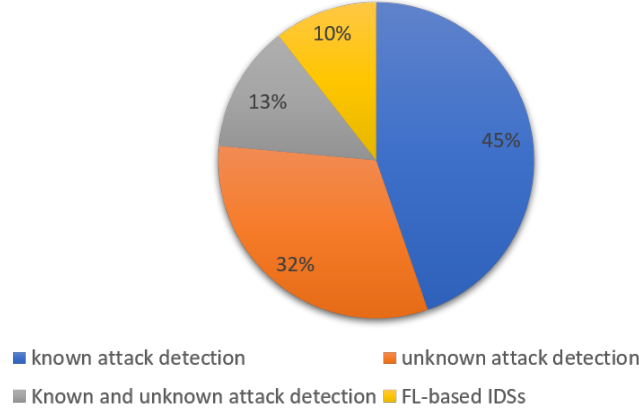


Figure 4: Distribution of collected papers by category

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. Figure 5 illustrates the categories and subcategories of the reviewed literature. Lastly, we review all the evaluation metrics used in the reviewed papers.

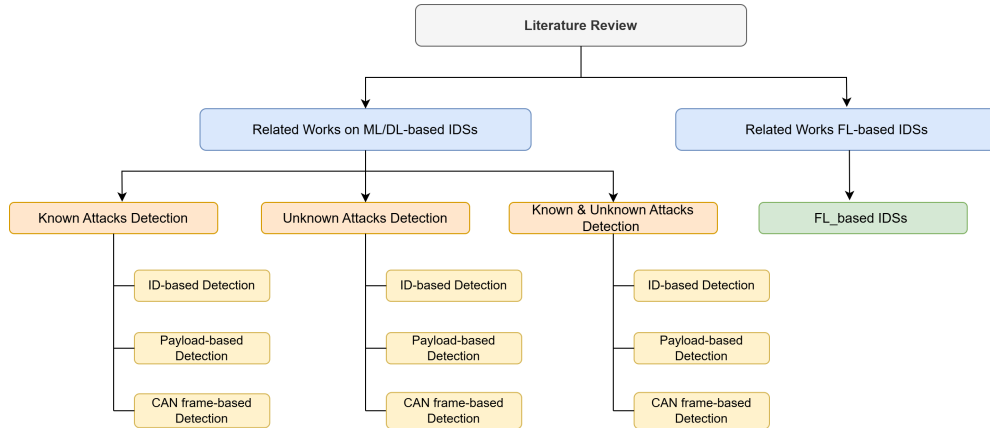


Figure 5: Categories of reviewed literature

5.1 Intrusion Detection System for In-Vehicle Networks

According to NIST SP 800-94, intrusion detection is “the process of monitoring the events occurring in a computer system or network and analysing them for signs of possible incidents” [Hernandez-Ramos et al.(2023)Hernandez-Ramos, Karopoulos, Chatzoglou, Kouliaridis, Marmol, Gonzalez-Vidal, and Kambourakis]. Therefore, an IDS is typically considered a software or hardware system designed to automatically detect suspicious activity in a network [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom]. In vehicle networks, IDSs are crucial for identifying malicious attacks [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap]. They can be implemented as either host-based or network-based systems [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom]. 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, Host-based IDSs are not a viable solution for vehicles, as they require a change in ECUs that are not cost effective

[Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap]. In contrast, deploying a network-based IDS as an additional node on the CAN bus is a more feasible and practical solution, as it avoids the need for any CAN bus modifications [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. Unlike IDSs in other applications, in-vehicle IDSs are constrained by computing power, memory size, 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 [Wu et al.(2019)Wu, Li, Xie, An, Bai, Zhou, and Li].

5.2 Overview of In-Vehicle IDS Approaches

Research on developing in-vehicle IDSs has grown significantly in recent years, driven by the critical need to enhance 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 [Hoppe et al.(2009)Hoppe, Kiltz, and Dittmann]. Anomaly-based IDSs are further categorised into statistical, ML, rule-based, and physical fingerprinting methods [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. However, this work specifically focuses on the development of in-vehicle IDSs using ML and DL approaches. 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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. 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 in-vehicle networks. Detection of known attacks typically relies on supervised learning, where models are trained on labelled data. The section is organised into three subsections based on the features used to build the model: ID-based detection, payload-based detection, and CAN frame-based detection. Each subsection examines different approaches for identifying malicious activities, emphasizing their strengths and limitations. Figure 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. [Song et al.(2020)Song, Woo, and Kim] 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, eliminating the need for additional feature engineering. They introduced a deep convolutional neural network (DCNN) model that was redesigned by minimizing unnecessary complexities within the Inception-ResNet architecture to achieve an optimised input size of $(29 \times 29 \times 1)$ and a binary classification output.

Refat et al. [Refat et al.(2022)Refat, Elkhail, Hafeez, and Malik] used graph-based techniques to extract features from the CAN IDs in in-vehicle networks. The authors converted a window of CAN IDs into a graph and extracted seven graph properties, including the number of nodes, number of edges, radius, diameter, density, reciprocity, average clustering coefficient, and assortative coefficient, to use as input features. These extracted features were then used to train two traditional ML algorithms: support vector machine (SVM) and k-nearest neighbours (KNN) models. The experimental results demonstrated that using graph-based features outperformed the traditional CAN bus features.

Nandam et al. [Nandam et al.(2022)Nandam, Vamshi, and Sucharitha] employed an Long Short-Term Memory (LSTM) model to detect DoS attacks on the CAN bus. The model utilises the CAN ID of incoming messages to identify 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.

Rangsikunpum et al. [Rangsikunpum et al.(2024a)Rangsikunpum, Amiri, and Ost] introduced a Binarised Neural Network (BNN)-based IDS to identify attacks on the CAN bus. The primary objective of the proposed IDS is to deploy a ML 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, making it suitable for deployment on cost-effective FPGA devices with reduced power require-

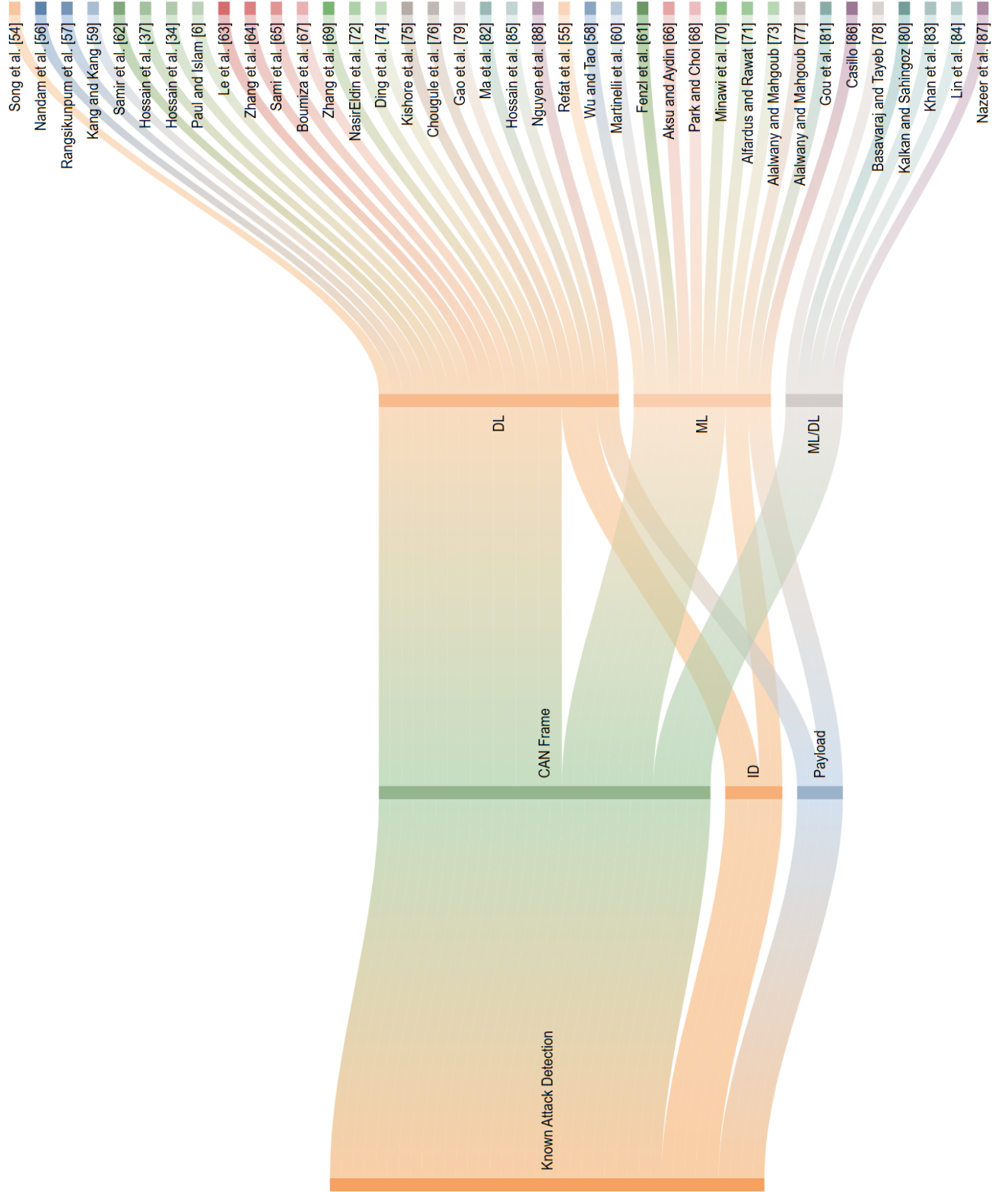


Figure 6: Related work on known attack detection

ments. 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 [Wu and Tao(2024)] 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 (XG-Boost). 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.

5.3.2 Payload-Based Detection

Some attacks, such as spoofing, use legitimate CAN IDs but modify the CAN payload values, depending on the characteristics of the particular attack. These changes alter the pattern of payload sequences. This section discusses IDSs that utilise this property and use the CAN payload as an input feature to detect known attacks.

Kang and Kang [Kang and Kang(2016)] developed a deep neural network (DNN)-based IDS to defend against malicious attacks. The authors utilised the 8-byte CAN payload to extract features, employing mode and value information to achieve dimensionality reduction. The initial weights for the DNN model were obtained from a separate Deep Belief Network. They then applied a template-matching method to compare the training samples with new CAN packets for detecting malicious messages. Although the proposed model demonstrated improved detection performance, it relied on mode and value information from CAN data, which presents significant challenges, especially without access to the DBC file.

Similarly, Martinelli et al. [Martinelli et al.(2017)Martinelli, Mercaldo, Nardone, and Santone] 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 identify four types of attacks targeting the CAN bus, including DoS, fuzzy, RPM, and gear spoofing. These algorithms include two types of fuzzy-rough KNN, the discernibility classifier, and a fuzzy unordered rule induction algorithm. The classification analysis was performed using the Weka3 tool. Experimental results indicated that the feature vector is a potential candidate for accurately classifying between injected and normal messages.

Fenzl et al. [Fenzl et al.(2021)Fenzl, Rieke, and Dominik] used decision trees (DTs) trained through genetic programming (GP) to detect intrusions in the CAN bus. 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). Experimental results showed 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. [Samir et al.(2024)Samir, Raissa, Touati, Hadded, and Ghazzai] investigated two DL-based IDSs: one leveraging LSTM and the other utilising a one-dimensional convolutional neural network (CNN). These two supervised learning algorithms serve as classifiers capable of categorising attacks into different types. The authors employed two public datasets and a new dataset that they manually generated using the ICSim simulation tool to cover more complex scenarios and attack types. Experimental results show that the LSTM-based IDS outperforms the CNN-based IDS, leveraging its ability to capture temporal patterns for robust detection of diverse CAN bus attacks.

5.3.3 CAN Frame-Based Detection

Rather than relying solely on CAN IDs or CAN payload as features, IDSs in the literature have utilised a combination of features to identify pattern changes in CAN data sequences. This approach offers the advantage of detecting 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 differences between consecutive CAN IDs, in combination with CAN ID and payload.

Hossain et al. [Hossain et al.(2020b)Hossain, Inoue, Ochiai, Fall, and Kadobayashi] proposed an LSTM-based IDS. The model considers both CAN ID, DLC, and payload as input features to detect point and contextual anomalies. The proposed LSTM is trained on both benign and attack data and employs both binary and multi-class classification. The authors collected CAN messages from an actual Toyota hybrid car and generated three attack scenarios, including DoS, fuzzy, and spoofing attacks. They compared the performance of the proposed LSTM method with the survival analysis method and found that the LSTM model achieves a higher detection rate than the other methods.

Following their earlier research, Hossain et al. [Hossain et al.(2020a)Hossain, Inoue, Ochiai, Fall, and Kadobayashi] introduced a 1D CNN model as an alternative to the LSTM model proposed in their previous study. They collected normal datasets from three cars: Toyota, Subaru, and Suzuki, and injected anomalous frames to create attacks, including DoS, fuzzy, RPM, and gear spoofing. The proposed model achieved a high attack detection rate for all types of

attacks. However, they considered fuzzy to be the most critical attack in the in-vehicle system, as it is difficult to detect due to its similarity to legitimate traffic within the CAN bus network.

Similarly, Paul and Islam [Paul and Islam(2021)] proposed an ANN-based anomaly detection method to identify unauthorised messages in CAN bus. They utilised benign and attack classes from DoS and fuzzy datasets to train their ANN model. The model demonstrated a high detection accuracy in distinguishing between legitimate and anomalous messages, achieving nearly negligible rates of false positives and false negatives.

Le et al. [Le et al.(2024)Le, Truong, Kim, et al.] proposed an IDS for multiclass classification based on a combination of AE models and a time-embedded transformer. The AE-based packet-level extraction model learns a compressed representation of each CAN frame within a CAN sequence, while the time-embedded transformer, which replaces positional encoding with a timestamp encoding component, is used as the sequence extraction component.

Zhang et al. [Zhang et al.(2024)Zhang, Yan, and Ma] introduced a Binarized CNN (BCNN)-based IDS, designed to leverage the temporal and spatial characteristics of CAN messages. The proposed IDS consists of two main components: 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. [Sami et al.(2020)Sami, Ibarra, Esparza, Al-Jufout, Aliasgari, and Mozumdar] 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 [Aksu and Aydin(2022)] proposed 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), Logistic Regression Classifier (LRC), Decision Tree Classifier (DTC), k-Nearest Neighbors Classifier (KNC), and Linear Discriminant Analysis Classifier (LDAC).

Boumiza et al. [Boumiza and Braham(2019)] proposed an IDS for the CAN bus based on a Multi-Layer Perceptron (MLP) neural network. The IDS first partitions data by the ID field of CAN packets, using 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 operates separately for each CAN ID, combining the individual decisions to calculate a final score and trigger an alert in the event of an attack detection.

Park and Choi [Park and Choi(2020)] proposed a multi-labeled hierarchical classification (MLHC) IDS to detect message injection attacks. MLHC identifies the occurrence of attacks and classifies them using only pre-existing labeled attack data. The authors evaluated the method's performance using four ML algorithms: SGD, kNN, DT, and RF. Simulation results showed that the MLHC model achieved high accuracy with the RF algorithm and rapid detection with the DT algorithm.

Zhang et al. [Zhang et al.(2020)Zhang, Cui, Cheng, and Zhang] proposed a Convolutional Encoder Network (CEN) model designed to detect network intrusions in CAN networks. The architecture integrates an encoder for dimensionality reduction, a CNN to increase network depth, and Inception ResNet to optimise training time. Additionally, the authors introduced a Feature-based Sliding Window method to extract features from the CAN Data Field and CAN IDs. Experimental results highlight the effectiveness of the feature-based sliding window in improving detection performance.

Minawi et al. [Minawi et al.(2020)Minawi, Whelan, Almeahmadi, and El-Khatib] proposed an ML-based IDS system 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 Tree (RT), Random Forest (RF), Stochastic Gradient Descent (SGD) with hinge loss, and Naive Bayes (NB) to detect different types of attacks. Additionally, this layer is designed with multiple modules, each tailored to detect specific types of attacks.

similarly, Alfaridus and Rawat [Alfaridus and Rawat(2021)] used the same proposed IDS in [Minawi et al.(2020)Minawi, Whelan, Almeahmadi, and El-Khatib] but with four different ML algorithms, including KNN, RF, SVM, and Multilayer Perceptron (MLP), to detect CAN bus attacks.

NasirEldin et al. [NasirEldin et al.(2021)NasirEldin, Bahaa-Eldin, and Sobh] proposed an attention-based model to detect CAN bus intrusions. The model consists of an attention layer that assigns higher importance to the most prominent features by calculating attention scores between the input features and the target, followed by a self-attention layer to identify relationships between data elements. Experimental results demonstrate that the proposed model outperformed baseline models, including an LSTM.

Alalwany and Mahgoub [Alalwany and Mahgoub(2022)] proposed an ML-based IDS for detecting attacks on the CAN bus 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 combined all supervised models using three ensemble methods: voting, stacking, and bagging. The ensemble learning strategy offers the advantage of enabling models with different capabilities to complement one another in the classification task. Compared to individual models, the ensemble classifiers outperformed the supervised classifiers, improving the effectiveness of the supervised ML models by leveraging diverse learning mechanisms to support one another.

Ding et al. [Ding et al.(2022)Ding, Zhu, Xie, and Lin] proposed an IDS based on a Bidirectional LSTM (Bi-LSTM) network 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. Experimental results demonstrate that the proposed model outperforms other network models, except for DoS attacks.

similarly, Kishore et al. [Kishore et al.(2024)Kishore, Rao, Nayak, and Behera] proposed a Bi-LSTM, which processes input data in both forward and backward orientations to detect anomalies in the CAN bus.

Chougule et al. [Chougule et al.(2024)Chougule, Kulkarni, Alladi, Chamola, and Yu] proposed HybridSecNet, a hybrid two-step LSTM-CNN IDS designed to enhance in-vehicle security. HybridSecNet consists of two classification stages: the first stage uses an LSTM to classify input data as either normal or attacked. If an attack is detected in the initial stage, the second stage is activated, employing a CNN-based multiclass classifier to identify and categorise the specific type of attack.

Moreover, Alalwany and Mahgoub [Alalwany and Mahgoub(2024)] proposed an in-vehicle IDS to improve the accurate detection and classification of CAN bus attacks in real time using ensemble techniques and the Kappa Architecture. The Kappa Architecture facilitates real-time attack detection, while ensemble learning combines multiple ML classifiers, including RF, DT, and XGBoost, to enhance detection accuracy. The study demonstrated that ensemble approaches, which integrate the strengths of multiple models, significantly improved detection accuracy and robustness.

Basavaraj and Tayeb [Basavaraj and Tayeb(2022)] proposed a lightweight DNN-based model to detect and classify attacks on the CAN bus. The proposed model outperformed baseline models, including RF, DTs, and the kNN algorithm.

Gao et al. [Gao et al.(2023)Gao, Huang, Liu, Du, and Zhang] 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 [Kalkan and Sahingoz(2020)] applied six different ML models: RF, bagging, ADA boosting, NB, LR, and ANN. Their 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. [Gou et al.(2023)Gou, Zhang, and Zhang] proposed an adaptive tree-based ensemble network (ATBEN) as the intrusion detection engine for IDS in the IoV. 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 demonstrated the effectiveness of the proposed IDS by evaluating its performance against a range of cyberattacks targeting both in-vehicle systems and external networks within the IoV.

Ma et al. [Ma et al.(2022)Ma, Cao, Mi, Huang, Liu, and Li] introduced a lightweight IDS for the CAN bus, leveraging a GRU-based architecture. To enhance efficiency, they employed a low-complexity feature extraction algorithm to derive features from CAN frames. The proposed model demonstrated near real-time performance and outperformed baseline models in detection accuracy.

Khan et al. [Khan et al.(2024)Khan, Javed, Iqbal, Asim, and Awad] 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 detect intrusions on the CAN bus. To improve detection performance, a meta-classifier aggregates the outputs of the base classifiers in a weighted and adaptive manner, considering their performances and correlations. This work addresses the trade-off between false positives and time complexity in CAN bus IDS.

Lin et al. [Lin et al.(2022)Lin, Wang, Chao, Lin, and Chen] proposed a CNN-based approach leveraging the VGG16 classifier to learn 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 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. [Hossain et al.(2020c)Hossain, Inoue, Ochiai, Fall, and Kadobayashi] proposed an LSTM-based IDS for detecting in-vehicle attacks. The CAN message data was collected using a tool called Vehicle Spy 3. 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 DoS, Fuzzing, and Spoofing attacks on the CAN bus of a Toyota Hybrid car using a Python-based program.

Casillo [Casillo et al.(2019)Casillo, Coppola, De Santo, Pascale, and Santonicola] 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 hours 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. [Nazeer et al.(2024)Nazeer, Alasiry, Qayyum, Madhan, Patil, and Srilatha] proposed a hybrid approach, DeepXG, which combines the XGBoost and DNN models to detect and classify attacks on the CAN bus. The XGBoost model 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. [Nguyen et al.(2023)Nguyen, Nam, and Kim] proposed an IDS based on a Transformer attention network for a CAN bus, designed to analyse a single message. The proposed IDS includes two models: one using only a single message and another leveraging sequential CAN IDs. The first model effectively detects 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 detect replay attacks by incorporating sequential CAN ID information. Additionally, the proposed model employs transfer learning to enhance the performance of models trained on small datasets from other car models.

Table 4 summarises the related work on known attack detection methods, including the learning approach, binary or multi-class classification, dataset used, detectable attacks, employed algorithm, and the model size or the size of trainable parameters. In Table 4, we assume that papers that do not explicitly state the input features used are referring to CAN frame features. Among these studies, only three [Song et al.(2020)Song, Woo, and Kim, Fenzl et al.(2021)Fenzl, Rieke, and Dominik, Le et al.(2024)Le, Truong, Kim, et al.] measure the trainable parameters, reflecting the model’s size, while the others did not consider model size for deployment.

Reference	Year	ML / DL	Category	Classification Type	Dataset	ID	Payload	Attack Types	Algorithm	Model Size / Trainable Parameters
ID-Based Attack Detection										
[Song et al.(2020)Song, Woo, and Kim]	2020	DL	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	Message Injection	DCNN	1.76 Million
[Refat et al.(2022)Refat, Elkhail, Hafeez, and Malik]	2022	ML	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	DoS, Fuzzy, RPM Spoofing	SVM, KNN	N/A
[Nandam et al.(2022)Nandam, Vamsi, and Sucharitha]	2022	DL	Supervised	Binary	car-hacking [Song et al.(2020)Song, Woo, and Kim]	✓	✓	DoS	LSTM	N/A
[Rangskunpum et al.(2024a)Rangskunpum, Amiri, and Osi]	2024	DL	Supervised	Binary / Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	BNN	4.85 Mb
[Wu and Tao(2024)]	2024	ML	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	Ensemble model (DT, ET, XGBoost)	N/A
Payload-Based Attack Detection										
[Kang and Kang(2016)]	2016	DL	Supervised	Binary	Simulation	✓	✓	Injection Attacks	DNN	N/A
[Martinielli et al.(2017)Martinielli, Mercaldo, Nardone, and Santone]	2017	ML	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing	KNN	N/A
[Fenzl et al.(2021)Fenzl, Riecke, and Dominek]	2021	ML	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim], Tesla Model X data, Renault Zoe electric car data	✓	✓	(RPM, Gear) Spoofing	DT, GP	1,101
[Samir et al.(2024)Samir, Raissa, Touati, Haddad, and Ghazzai]	2024	DL	Supervised	Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim], OTIDS [Lee et al.(2017)Lee, Jeong, and Kim], Own	✓	✓	DoS, Fuzzy, Spoofing, Replay	CNN, LSTM	N/A
CAN Frame-Based Attack Detection										
[Hossain et al.(2020b)Hossain, Inoue, Ochiai, Fall, and Kadoyayashi]	2020	DL	Supervised	Binary / Multi-class	Own	✓	✓	DoS, Fuzzy, Spoofing	LSTM	N/A
[Hossain et al.(2020a)Hossain, Inoue, Ochiai, Fall, and Kadoyayashi]	2020	DL	Supervised	Binary / Multi-class	Own	✓	✓	DoS, Fuzzy, Spoofing	CNN	N/A
[Paul and Islam(2021)]	2021	DL	Supervised	Binary	OTIDS [Lee et al.(2017)Lee, Jeong, and Kim]	✓	✓	DoS, Fuzzy	ANN	N/A
[Le et al.(2024)Le, Truong, Kim, et al.]	2024	DL	Supervised	Multi-class	car-hacking [Song et al.(2020)Song, Woo, and Kim], ROAD [Verma et al.(2020)Verma, Iannacoe, Bridges, Holtfield, Kay, and Combs]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Fabrication, Masquerade	AE, Time-embedded Transformer	259,000
[Zhang et al.(2024)Zhang, Yan, and Ma]	2024	DL	Supervised	Binary	Own	✓	✓	Replay, Spoofing	BCNN	N/A
[Sami et al.(2020)Sami, Ibarra, Esparra, Al-Jufout, Aliasgari, and Moramdar]	2020	DL	Supervised	Binary	OTIDS [Lee et al.(2017)Lee, Jeong, and Kim], ML-280 [Sami(2019)]	✓	✓	DoS, Fuzzy, Impersonation	DNN	N/A
[Akou and Aydin(2022)]	2022	ML	Supervised	Binary / Multi-class	car-hacking [Song et al.(2020)Song, Woo, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	SVC, LRC, DTC, KNC, LDAC	N/A
[Boumitza and Brahmi(2019)]	2019	DL	Supervised	Binary	Dataset [Taylor et al.(2018)Taylor, Leblanc, and Japkowicz]	✓	✓	Frequency modification, Data-content modification	MLP	N/A
[Park and Choi(2020)]	2020	ML	Supervised	Binary / Multi-class	Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	✓	✓	Fuzzy, Flooding, Malfunction	SGD, LNN, DT, RF	N/A
[Zhang et al.(2020)Zhang, Cui, Cheng, and Zhang]	2020	DL	Supervised	Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim], car-hacking [Song et al.(2020)Song, Woo, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	CEN	N/A
[Minawi et al.(2020)Minawi, Whelan, Almelhadi, and El-Khatib]	2020	ML	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	RT, RF, SGD, NB	N/A
[Alfaridus and Rawat(2021)]	2021	ML	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	KNN, RF, SVM, MLP	N/A
[NaseEldin et al.(2021)NaseEldin, Bahaa-Eldin, and Sothi]	2021	DL	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	Attention-based model	N/A
[Alshwary and Mahgoub(2022)]	2022	ML	Supervised	Binary	Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	✓	✓	Flooding, Spoofing, Replay, Fuzzy	LR, GaussianNB, k-NN, RF, Gradient Boosting, AdaBoost, DT, XGBoost	N/A
[Ding et al.(2022)Ding, Zhu, Xie, and Lin]	2022	DL	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	Bi-LSTM	N/A
[Kishore et al.(2024)Kishore, Ray, Nayak, and Behera]	2024	DL	Supervised	Binary	Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	✓	✓	Flooding, Spoofing, Replay, Fuzzy	Bi-LSTM	N/A
[Chougule et al.(2024)Chougule, Kulkarni, Alhadi, Chamola, and Yu]	2024	DL	Supervised	Binary / Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	DoS, Fuzzy, (Gear, RPM) Spoofing	LSTM-CNN	N/A
[Alshwary and Mahgoub(2024)]	2024	ML	Supervised	Multi-class	Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	✓	✓	DoS, Spoofing, Replay, Fuzzy	RF, DT, XGBoost	N/A
[Basarraj and Tayeb(2022)]	2022	DL / ML	Supervised	Multi-class	CAN dataset [Dupont et al.(2019b)Dupont, Lekidis, den Hartog, and Etalle]	✓	✓	Reconnaissance, DoS, Fuzzy	DNN	N/A
[Gao et al.(2023)Gao, Huang, Liu, Du, and Zhang]	2023	DL	Supervised	Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	DoS, (Gear, RPM) Spoofing, Fuzzy	CNN, bi-LSTM	N/A
[Kalkan and Salinas(2020)]	2020	ML / DL	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	RF, bagging, ADA boosting, NB, LR, ANN	N/A
[Gou et al.(2023)Gou, Zhang, and Zhang]	2023	ML	Supervised	Multi-class	car-hacking [Song et al.(2020)Song, Woo, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	XGBoost, LightGBM, RF, ET	N/A
[Ma et al.(2022)Ma, Cao, Mi, Huang, Liu, and Li]	2022	DL	Supervised	Binary	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	DoS, Spoofing, Fuzzy	GRU	N/A
[Khan et al.(2024)Khan, Javed, Iqbal, Asim, and Awad]	2024	ML / DL	Supervised	Multi-class	OTIDS [Lee et al.(2017)Lee, Jeong, and Kim]	✓	✓	DoS, Fuzzy, Impersonation	DNN, MLP, light gradient-boosting machine, ET, RF, Bagging, KNN	N/A
[Lin et al.(2022)Lin, Wang, Chao, Lin, and Chen]	2022	DL / ML	Supervised	Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	VGG16, XGBoost	N/A
[Hossain et al.(2020c)Hossain, Inoue, Ochiai, Fall, and Kadoyayashi]	2020	DL	Supervised	Binary / Multi-class	Own	✓	✓	DoS, Fuzzy, Spoofing	LSTM	N/A
[Castillo et al.(2019)Castillo, Coppola, De Santo, Pascale, and Santonicola]	2019	ML	Supervised	Binary	Simulation	✓	✓	Turn right, Turn left, Brake	Bayesian Network	N/A
[Nazeer et al.(2024)Nazeer, Alasiry, Qayyum, Madhan, Paul, and Sripatha]	2024	ML / DL	Supervised	Multi-class	Own	✓	✓	Flooding, Replay, Spoofing	XGBoost, DNN	N/A
[Nguyen et al.(2023)Nguyen, Nam, and Kim]	2023	DL	Supervised	Binary / Multi-class	Car Hacking [Seo et al.(2018)Seo, Song, and Kim], VYN Hacking and (HRC) (2019)], Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Replay, Malfunction	Transformer	N / A

Table 4: Summary of related work on known attack detection methods

5.3.4 Limitations of Known Attacks Detection

Despite the high accuracy and low false alarm rate (FAR) of the proposed known attack detection models, their performance heavily depends on well-labelled attack data and balanced datasets. However, obtaining labelled data remains a significant challenge for researchers [Hernandez-Ramos et al.(2023)Hernandez-Ramos, Karopoulos, Chatzoglou, Kouliaridis, Marmol, Gonzalez-Vidal, and Kambourakis]. Moreover, the labeling process is often time-consuming, prone to errors, and tedious [Said Elsayed et al.(2020)Said Elsayed, Le-Khac, Dev, and Jurcut].

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. Because attackers continuously attempt to evade detection and use new, previously unseen attacks, supervised learning-based models struggle to recognise unfamiliar attack patterns not present in the training data [Vikram et al.(2020)]. This limitation can lead to significant security consequences.

5.4 Unknown Attacks Detection

As mentioned in Section 4.1 there are 27 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 in-vehicle networks. Detection of unknown attacks typically relies on unsupervised learning, where models are trained solely on normal data, relying on profiling normal traffic behaviours to detect anomalous traffic that could indicate a potential attack. As a result, unsupervised learning-based models are well suited to detecting previously unseen attacks [Pratomo et al.(2018)Pratomo, Burnap, and Theodorakopoulos]. The section is organised into three subsections based on the features used to build the model: ID-based detection, payload-based detection, and CAN frame-based detection. Each subsection examines different approaches for identifying malicious activities, emphasizing their strengths and limitations. Figure 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.

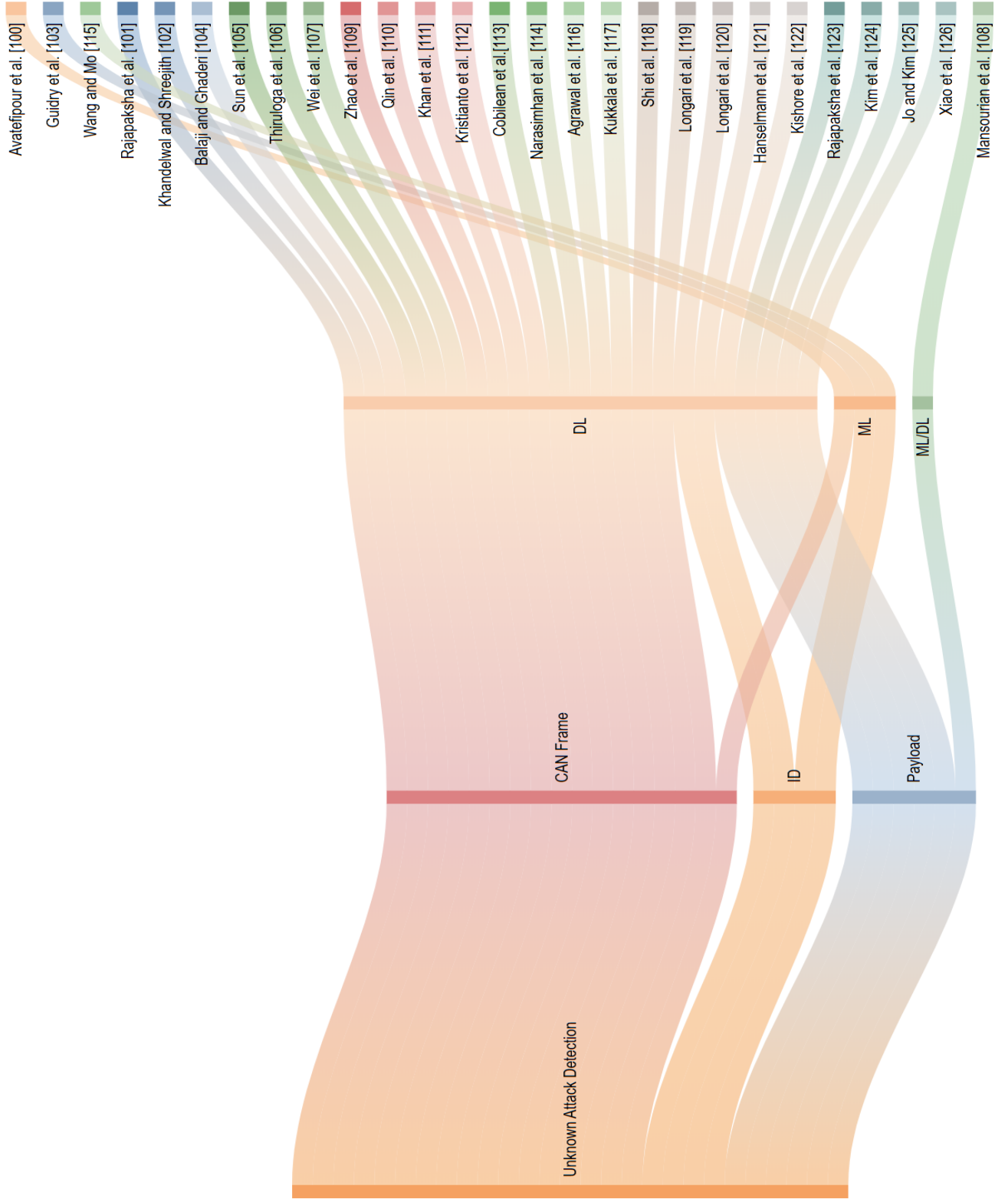


Figure 7: Related work on unknown attack detection

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. [Avatefipour et al.(2019)Avatefipour, Al-Sumaiti, El-Sherbeeney, Awwad, Elmeligy, Mohamed, and Malik] proposed an anomaly detection model based on a modified one-class support vector machine (OCSVM) incorporating the modified bat algorithm (MBA). The model was built using normal CAN bus 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 by the model as malicious activity. For evaluation, CAN bus data were collected from a licensed, unmodified vehicle and two public CAN bus datasets. 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 false alarm rate compared to both alternatives.

Rajapaksha et al. [Rajapaksha et al.(2022)Rajapaksha, Kalutarage, Al-Kadri, Madzudzo, and Petrovski] 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 detecting diverse attacks on the CAN bus.

Khandelwal and Shreejith [Khandelwal and Shreejith(2023)] presented a convolutional autoencoder (CAE) model for detecting zero-day attacks, trained solely on benign CAN messages. Leveraging Vitis-AI tools, they quantised the model to optimise performance on resource-constrained platforms. The proposed IDS achieves state-of-the-art classification accuracy across multiple unseen attacks, along with a 1.3x speed-up in processing latency and approximately 2x reduction in power consumption compared to existing state-of-the-art IDSs.

Guidry et al. [Guidry et al.(2023)Guidry, Sohrab, Gottumukkala, Katragadda, and Gabbouj] proposed the use of a One-Class Support Vector Machine (OC-SVM) to detect anomalous data on a vehicle's 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.

5.4.2 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 [Balaji and Ghaderi(2021)] 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 performs a linear transformation of the input data from each CAN ID, passes it through a sigmoid function, and accumulates it over all IDs. This is followed by an LSTM and an output layer, forming a prediction-based anomaly detector. The use of payload values from other IDs as context enables the capture of inter-ID correlations. However, the model is trained separately for each CAN ID, resulting in high memory and computational costs.

Sun et al. [Sun et al.(2021)Sun, Chen, Weng, Liu, and Geng] proposed a CNN-LSTM-based IDS with an attention mechanism. The model used one-dimensional convolution to extract abstract features, while a bi-directional LSTM was employed to capture time dependencies. The bit flip rate was used to identify 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 fed into 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. [Thiruloga et al.(2022)Thiruloga, Kukkala, and Pasricha] introduced TENET, a novel anomaly detection framework based on temporal convolutional neural attention (TCNA) networks. TENET takes a sequence of signal values from a message as input and uses CNNs to predict the signal values of the next message instance by learning the underlying probability distribution of normal data. 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. [Wei et al.(2022)Wei, Wang, Dai, Li, and He] 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. [Mansourian et al.(2023)Mansourian, Zhang, Jaekel, Zamanirafe, and Kneppers] proposed an anomaly-based IDS to detect attacks on the in-vehicle CAN bus. The proposed IDS comprises three modules: an LSTM model, a prediction error calculator, and a Gaussian Naïve Bayes (GNB) classifier. The LSTM is trained on normal CAN messages to learn the typical sequential behaviour of each ECU. Once trained, the network predicts the next expected payload of an ECU based on past observations and compares it to the actual received value. When an attack occurs, the trained LSTM network fails to make accurate predictions, resulting in higher-than-normal prediction errors. The GNB classifier then classifies messages as either normal or an attack based on these prediction errors.

Zhao et al. [Zhao et al.(2022a)Zhao, Xun, Liu, and Ma] 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. Experiments on two real vehicles show that GVIDS successfully detects SOME attacks as well as other existing attack types.

5.4.3 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. [Qin et al.(2021)Qin, Yan, and Ji] proposed an LSTM-based anomaly detection algorithm to detect abnormal behaviour on the CAN bus. CAN data were collected from the network of a real vehicle, with simulated attacks such as tampering or inserting duplicate random packets into the CAN bus. 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 could detect anomalous data with over 90% accuracy.

Khan et al. [Khan et al.(2021)Khan, Moustafa, Pi, Haider, Li, and Jolfaei] used a bidirectional LSTM model with an improved feature processing technique to address the challenge of 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 applied enhanced data pre-processing to improve the scalability and performance efficiency of the IDS, including feature conversion, feature reduction, and feature normalization. Principal component analysis was used for feature reduction. Experimental results showed that the feature pre-processing led to a 19.31% improvement in accuracy compared to raw data.

Kristianto et al. [Kristianto et al.(2024)Kristianto, Lin, and Hwang] proposed a lightweight unsupervised IDS on 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. [Cobilean et al.(2023)Cobilean, Mavikumbure, Wickramasinghe, Varghese, Pennington, and Manic] proposed a Transformer neural network-based IDS designed to predict anomalous behaviour within CAN protocol communication. The Transformer model is trained to predict the next communication sequence, and anomalies are detected when the difference between the predicted sequence and the actual received sequence exceeds a defined threshold. A key advantage of this model is that it does not require labelled attack data for learning the communication sequence.

Narasimhan et al. [Narasimhan et al.(2021)Narasimhan, Ravi, and Mohammad] 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 demonstrated that the proposed method achieved superior performance across 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 [Wang and Mo(2021)] proposed a CAN bus anomaly detection model based on the 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. [Agrawal et al.(2022a)Agrawal, Alladi, Agrawal, Chamola, and Benslimane] 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 normal CAN data, and the system classifies incoming CAN data as genuine or anomalous using a reconstruction-based thresholding approach.

Kukkala et al. [Kukkala et al.(2020)Kukkala, Thiruloga, and Pasricha] proposed INDRA, an IDS based on a GRU-based recurrent autoencoder designed to learn latent representations of normal CAN traffic and detect anomalies 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 identify anomalous signals. The authors trained separate autoencoder models for each CAN ID, enabling ID-specific anomaly detection model.

Shi et al. [Shi et al.(2024)Shi, Xie, Dong, Jiang, and Jin] 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 demonstrate that the proposed IDS achieves superior detection performance compared to traditional ML algorithms and other deep clustering methods.

Longari et al. [Longari et al.(2020)Longari, Valcarcel, Zago, Carminati, and Zanero] introduced CANnolo, an IDS based on LSTM autoencoders for identifying anomalies on the CAN bus. CANnolo analyses CAN message streams to construct a model of normal data sequences and detects 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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah].

To improve the overall architecture and reduce the computational requirements of CANnolo, ensuring it meets the real-time constraints of the automotive domain, Longari et al. [Longari et al.(2023)Longari, Pozzoli, Nichelini, Carminati, and Zanero] then proposed CANDito, an unsupervised IDS that leverages LSTM autoencoders to detect anomalies using a signal reconstruction process. CANDito reconstructs the time series of CAN packets for each ID and calculates anomaly scores based on the reconstruction error.

Hanselmann et al. [Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer] proposed CANet, an LSTM-based autoencoder designed to identify attacks on the CAN bus. Separate LSTM models were used for each CAN ID, with their outputs concatenated into a single latent vector. The difference between the original and reconstructed signal values was utilised to determine the normal status. Experimental results showed that the model achieved a high detection rate with low false-positive and false-negative rates across various attack types.

Similarly, Kishore et al. [Kishore et al.(2022)Kishore, Rao, and Behera] proposed an LSTM-based anomaly detection method. The model outperforms previous tree-based ML algorithms, including AdaBoost, GBoost, Bagging, XGBoost, and LGBM.

Rajapaksha et al. [Rajapaksha et al.(2023b)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, and Madzudzo] 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 false negatives

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. [Kim et al.(2023)Kim, Kim, and You] proposed an IDS based on multiple LSTM-Autoencoders that utilise diverse features, including transmission intervals and payload value changes, to capture various characteristics of normal network behaviour. The system consists of a feature sequence extractor, LSTM-Autoencoder models, and an anomaly detector. The time interval sequence extractor calculates the intervals between consecutive frames with the same ID, generating a chronological 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 [Jo and Kim(2024)] 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. [Xiao et al.(2019)Xiao, Wu, and Li] proposed an anomaly detection IDS for in-vehicle networks based on 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 5 summarises the related work on unknown attack detection methods, including the learning approach, dataset used, detectable attacks, employed algorithm, and the model size or the size of trainable parameters. In Table 5, we assume that papers that do not explicitly state the input features used are referring to CAN frame features.

Among these studies, only three [Song et al.(2020)Song, Woo, and Kim, Fenzl et al.(2021)Fenzl, Rieke, and Dominik, Le et al.(2024)Le, Truong, Kim, et al.] measure the trainable parameters, reflecting the model’s size, while the others did not consider model size for deployment.

Reference	Year	ML / DL	Category	Dataset	ID	Payload	Attack Types	Algorithm	Model Size / Trainable Parameters
ID-Based Attack Detection									
[Avatefpour et al.(2019)Avatefpour, Al-Sumaiti, El-Sherbeeny, Awwad, Elmelig, Mohamed, and Malik]	2019	ML	Unsupervised	Own, Dodge (Courtroom)[], OTIDS [Lee et al.(2017)Lee, Jeong, and Kim]	✓		Injection	MBA-OCSVM	N/A
[Rajapaksha et al.(2022)Rajapaksha, Kalutarage, Al-Kadri, Madrazo, and Petrovski]	2022	DL	Unsupervised	ROAD [Verna et al.(2020)Verna, Iannaccone, Bridges, Hollifield, Kay, and Combs], car-hacking [Song et al.(2020)Song, Woo, and Kim], Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	✓		Fabrication, Spoofing, Masquerade	GRU	N/A
[Khandewal and Shreejith(2023)]	2023	DL	Unsupervised	car-hacking [Song et al.(2020)Song, Woo, and Kim]	✓		DoS, Fuzzy, (Gear, RPM) Spoofing	AE	N/A
[Guidry et al.(2023)Guidry, Sohrab, Göttemukala, Katragadda, and Gabbouj]	2023	ML	Unsupervised	Own	✓		Random ID, Zero ID, Replay	OC-SVM	N/A
Payload-Based Attack Detection									
[Bahji and Ghaderi(2021)]	2021	DL	Unsupervised	Two Public Datasets from [Seo et al.(2018)Seo, Song, and Kim]	✓		Flood, Replay, Drop, Spoofing, Fuzzy	LSTM	N/A
[Sun et al.(2021)Sun, Chen, Weng, Liu, and Geng]	2021	DL	Unsupervised	CAN Signal Extraction and Translation [Song and Kim(2020)]	✓		Flood, Replay, Drop, Spoofing, Fuzzy	CNN-LSTM	682 KB
[Thiruloga et al.(2022)Thiruloga, Kukkala, and Patiricha]	2022	DL	Unsupervised	Simulation	✓		Platoon, Continuous Change, Playback, Suppress	CNN	59.62 KB / 6064
[Wei et al.(2022)Wei, Wang, Dai, Li, and He]	2022	DL	Unsupervised	OTIDS [Lee et al.(2017)Lee, Jeong, and Kim]	✓		Payload value Manipulation	AE	N/A
[Mansourian et al.(2023)Mansourian, Zhang, Jaekel, Zamanirafae, and Kneppers]	2023	ML / DL	Unsupervised	Car Hacking [Seo et al.(2018)Seo, Song, and Kim], Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	✓		(Gear, RPM) Spoofing: DoS, Fuzzy, Flooding, Malfunction	LSTM, GNB	N/A
[Zhao et al.(2022a)Zhao, Xun, Liu, and Ma]	2022	DL	Unsupervised	Own	✓		Spoofing, Bit-off, Masquerade, SOME attacks	GAN	N/A
CAN Frame-Based Attack Detection									
[Qin et al.(2021)Qin, Yan, and Ji]	2021	DL	Unsupervised	Own	✓	✓	Random CAN payload Values	LSTM	N/A
[Khan et al.(2021)Khan, Moustafa, Pi, Haider, Li, and Joffe]	2021	DL	Unsupervised	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	DoS, Fuzzy, RPM, Gear Spoofing	LSTM	N/A
[Kristianto et al.(2024)Kristianto, Lin, and Hwang]	2024	DL	Unsupervised	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy	RNNs and AEs	Multiple IDSs (119~272) KB for each gateway
[Coblean et al.(2023)Coblean, Movikombore, Wickramasinghe, Varghese, Pennington, and Manic]	2023	DL	Self-supervised	Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	✓	✓	Malfunction	Transformer	N/A
[Narasimhan et al.(2021)Narasimhan, Kiri, and Mohanram]	2021	DL	Unsupervised	ME-SSE [Kang(2019)]	✓	✓	DoS, Fuzzy	AE, GMM	N/A
[Wang and Mao(2021)]	2021	ML	Supervised	Simulation	✓	✓	(Gear, RPM) Spoofing	FLXGBoost	N/A
[Agrawal et al.(2022a)Agrawal, Alladi, Agrawal, Chumola, and Bendimane]	2022	DL	Unsupervised	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	✓	✓	(Gear, RPM) Spoofing: DoS, Fuzzy	CNNs, LSTMs	N/A
[Kukkala et al.(2020)Kukkala, Thiruloga, and Patiricha]	2020	DL	Unsupervised	[Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer]	✓	✓	Flooding, Platoon, Continuous, Suppress, Playback	GRU AE	443 kB
[Shi et al.(2024)Shi, Xie, Dong, Jiang, and Jin]	2024	DL	Unsupervised	Car Hacking [Seo et al.(2018)Seo, Song, and Kim], Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	✓	✓	(Gear, RPM) Spoofing: DoS, Replay, Fuzzy	LSTM, CNN, AE	N/A
[Longari et al.(2020)Longari, Vakracel, Zago, Carminati, and Zanero]	2020	DL	Unsupervised	Recan [Zago et al.(2020)Zago, Longari, Tricarico, Carminati, Pérez, Pérez, and Zanero]	✓	✓	Interleave, Discontinuity, Data field anomalies	LSTM-AE	Less than 10 MB
[Longari et al.(2023)Longari, Pozzoli, Nichelini, Carminati, and Zanero]	2023	DL	Unsupervised	Recan [Zago et al.(2020)Zago, Longari, Tricarico, Carminati, Pérez, Pérez, and Zanero], car-hacking [Song et al.(2020)Song, Woo, and Kim]	✓	✓	(Gear, RPM) Spoofing, DoS, Fuzzy, Masquerade, Seamless change, Replay	LSTM AE	N/A
[Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer]	2020	DL	Unsupervised	SynCAN [Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer]	✓	✓	Flooding, Platoon, Continuous, Suppress, Playback	LSTM AE	N/A
[Kishore et al.(2022)Kishore, Rao, and Behera]	2022	DL	Unsupervised	Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	✓	✓	Flooding, Spoofing, Replay, Fuzzy	LSTM	N/A
[Rajapaksha et al.(2023b)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, and Madrazo]	2023	DL	Unsupervised/Supervised	SynCAN [Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer], ROAD [Verna et al.(2020)Verna, Iannaccone, Bridges, Hollifield, Kay, and Combs]	✓	✓	13 different attacks	GRU, Latent AE	94MB
[Kim et al.(2023)Kim, Kim, and You]	2023	DL	Unsupervised	Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim], Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	✓	✓	Spoofing, Replay, Fuzzy	LSTM-AEs	3.88 - 3.98 MB
[Jo and Kim(2024)]	2024	DL	Unsupervised	Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	✓	✓	Flooding, Fuzzy, Malfunction	Transformer	N/A
[Xiao et al.(2019)Xiao, Wu, and Li]	2019	DL	Unsupervised	OTIDS [Lee et al.(2017)Lee, Jeong, and Kim]	✓	✓	DoS, Fuzzy, Impersonation	ConvLSTM	N/A

Table 5: Summary of related work on unknown attack detection methods

5.4.4 Limitations of Unknown Attacks Detection

All the proposed anomaly detection IDSs are trained on normal data and use binary classification to classify traffic data as either normal or anomalous, detecting any deviations from the normal data. 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 [Zhao et al.(2022b)Zhao, Chen, Gu, Luan, Zeng, and Chakraborty]. 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 detect and classify known attacks while also identifying new, unknown attacks.

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 capable of identifying both known and unknown attacks [Zhang et al.(2019)Zhang, Li, Zhang, Li, and Li, Hoang and Kim(2022), Seo et al.(2018)Seo, Song, and Kim, Yang et al.(2022a)Yang, Moubayed, and Shami, Nakamura et al.(2021)Nakamura, Takeuchi, Kashima, Kishikawa, Ushio, Haga, and Sasaki, Rangsikunpum et al.(2024b)Rangsikunpum, Amiri, and Ost, Han et al.(2021)Han, Kwak, and Kim, Gherbi et al.(2020)Gherbi, Hanczar, Janodet, and Klaudel, Nguyen et al.(2024)Nguyen, Cho, and Kim, Lin et al.(2021)Lin, Wei, Li, and Long], demonstrating significant advancements in this critical area of cybersecurity. This section reviews state-of-the-art studies and their limitations. Figure 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. [Hoang and Kim(2022)] and Seo et al. [Seo et al.(2018)Seo, Song, and Kim] have showcased their IDSs' ability to detect both seen and unseen attacks. However, their IDSs mainly rely on the CAN ID as a singular feature; selecting only the CAN ID feature will limit the detection ability to detect attacks that involve payload manipulation [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah].

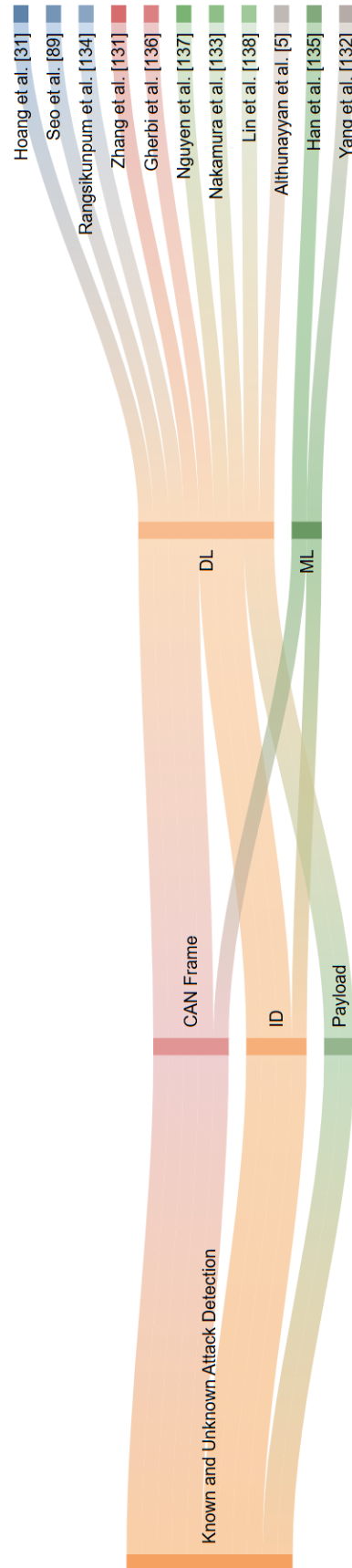


Figure 8: Related work on known and unknown attack detection

Hoang et al. [Hoang and Kim(2022)] propose a lightweight, semi-supervised, learning-based IDS to detect in-vehicle network attacks. The proposed IDS in their study integrates two DL models: autoencoders and generative adversarial networks (GAN). Their IDS was trained on unlabelled data to learn the patterns of normal and malicious data. Only a few labelled samples were used during the subsequent supervised training phase. 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. [Seo et al.(2018)Seo, Song, and Kim] have developed a GAN-based IDS (GIDS) for in-vehicle network security. The proposed IDS was trained by solely utilising the patterns of CAN IDs from CAN data and then converting the extracted CAN IDs into a simple image. GIDS has two discriminative models to detect both seen and unseen attack data. The first discriminator is specifically trained to handle attacks. In contrast, the second discriminator and the generator are co-trained through an adversarial process. While the generator generates modified images, the second discriminator receives both modified and real CAN images, and its role is to differentiate between the modified and real images.

Rangsikunpum et al. [Rangsikunpum et al.(2024b)Rangsikunpum, Amiri, and Ost] proposed a Binarized Neural Network (BNN)-based IDS (BIDS) for unknown attack detection and known attack classification, utilizing a BNN and a GAN. The model is hierarchically structured into two stages: attack detection in the first stage and known attack classification in the second. To capture sequential patterns in CAN IDs, consecutive CAN IDs are encoded into one-hot vectors and arranged into a 48×48 2D grid. The proposed model is resource-efficient, requiring minimal computational power, making it highly suitable for deployment on low-cost FPGA platforms.

Han et al. [Han et al.(2021)Han, Kwak, and Kim] proposed 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, such as mean, variance, quartile deviation, skewness, and kurtosis, 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 makes the model lightweight, 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. [Zhang et al.(2019)Zhang, Li, Zhang, Li, and Li] have proposed a DNN-based IDS that aims to automatically extract features for the IDS from the vehicle's data packets. The authors applied gradient descent with momentum (GDM) and gradient descent with momentum and adaptive gain (GDM/AG) techniques. The study's results demonstrate the model's capability to detect replay attacks effectively. The authors accessed the sensor readings, using them as separate features. However, the main limitation of the proposed IDS is that it requires either access to the DBC file or knowledge of the CAN payload, which is confidential and proprietary to the vehicle manufacturer [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar].

Gherbi et al. [Gherbi et al.(2020)Gherbi, Hanczar, Janodet, and Klaudel] proposed 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.

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. [Nguyen et al.(2024)Nguyen, Cho, and Kim] propose 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 reducing the amount of labelled data required.

Nakamura et al. [Nakamura et al.(2021)Nakamura, Takeuchi, Kashima, Kishikawa, Ushio, Haga, and Sasaki] proposed a hybrid model combining a LightGBM-based supervised model and an autoencoder-based unsupervised model 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. Experimental results showed that the hybrid model outperformed the pre-trained LightGBM model.

Lin et al. [Lin et al.(2021)Lin, Wei, Li, and Long] proposed 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 [Nakamura et al.(2021)Nakamura, Takeuchi, Kashima, Kishikawa, Ushio, Haga, and Sasaki] and [Lin et al.(2021)Lin, Wei, Li, and Long] are limited to binary classification, do not consider multi-class classification for known attacks, and do not account for the model size.

Most of the aforementioned studies employ either supervised learning-based methods or unsupervised learning-based methods. To leverage the strengths of both approaches, Yang et al. [Yang et al.(2022a)Yang, Moubayed, and Shami] have developed a multi-tiered IDS, MTH-IDS, to protect intra-vehicle and external networks from cyberattacks. MTH-IDS use 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 has certain 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 false positives (FPs) and false negatives (FNs) may lead to poorer performance when testing the model on new, unseen data. Furthermore, adding this tier shifts the model from being purely unsupervised, creating a dependency on labelled datasets, which are often challenging to implement in practical scenarios. 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 [Kocher and Kumar(2021)]. This presents a critical limitation in CAN bus data for three reasons. First, selecting a subset of CAN bus payload features as important while discarding others could allow attackers to exploit the neglected features and bypass the model [Li and Vorobeychik(2014), Zhang et al.(2015)Zhang, Chan, Biggio, Yeung, and Roli]. Second, the evolving landscape of attack scenarios means that features chosen to detect one category of attack may become outdated or insufficient to address new, unseen attacks [Kocher and Kumar(2021)].

Yang et al. [Yang et al.(2022a)Yang, Moubayed, and Shami] 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 [Jan et al.(2019)Jan, Farman, Khan, Imran, Islam, Ahmad, Ali, and Jeon]. Considering that modern vehicle ECUs produce around 2,000 CAN frames per second [Seo et al.(2018)Seo, Song, and Kim], this capability is essential to handle the extensive data of the CAN bus. Moreover, multiple studies have found that DL-based IDSs outperform traditional ML-based IDSs in automotive applications [Mehedi et al.(2021)Mehedi, Anwar, Rahman, and Ahmed]. This superiority is due to several factors: DL methods are more adaptive, continually being refined with incoming data, which is particularly suitable for the nature of CAN bus data [Zhang et al.(2019)Zhang, Li, Zhang, Li, and Li]. Additionally, traditional ML often requires manual feature engineering, such as applying correlation-based feature selection, which can be time-consuming [Nagarajan et al.(2023)Nagarajan, Mansourian, Shahid, Jaekel, Saini, Zhang, and Kneppers]. In contrast, DL automatically deduces features, allowing algorithms to directly discern optimal features from raw data [Lampe and Meng(2023b)]. Furthermore, DL-based IDSs are especially capable of detecting novel attacks and can scale more effectively to highly complex in-vehicle network data while maintaining efficacy [Lampe and Meng(2023b)].

To address these limitations, Althunayyan et al. [Althunayyan et al.(2024a)Althunayyan, Javed, and Rana] proposed a DL-based IDS with a multi-stage approach designed to detect both known and unknown attacks, considering that some attacks may evade detection and be misclassified as normal. The first stage employs a supervised ANN to detect and classify known attacks, while the second stage utilizes an unsupervised LSTM autoencoder to identify unknown attacks that bypass the first model. If the supervised model misclassifies malicious traffic as normal, the anomaly detection model detects deviations from learned patterns and flags them as unseen attacks. Despite incorporating two

models, the approach remains lightweight and practical for deployment. Table 6 summarises the details of known and unknown attack detection studies.

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

Reference	Year	ML / DL	Category	Dataset	Algorithm	M-C	ID	Payload	Model Size/ Trainable Parameters	FL
ID-Based Attack Detection										
[Hoang and Kim(2022)]	2022	DL	Semi-supervised	Car-Hacking [Seo et al.(2018)Seo, Song, and Kim]	AE, GAN		✓		2.15 million	
[Seo et al.(2018)Seo, Song, and Kim]	2018	DL	Unsupervised	Car-Hacking [Seo et al.(2018)Seo, Song, and Kim]	GAN			✓	N/A	
[Rangskunpum et al.(2024b)Rangskunpum, Amiri, and Oot]	2024	DL	Semi-supervised	Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	BNN, GAN	✓	✓		4.07 Mb	
[Han et al.(2021)Han, Kwak, and Kim]	2021	ML	Unsupervised	Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	DT, RF, XGBoost		✓	✓	N/A	
Payload-Based Attack Detection										
[Zhang et al.(2019)Zhang, Li, Zhang, Li, and Li]	2019	DL	Supervised	Simulation	DNN			✓	N/A	
[Gherbi et al.(2020)Gherbi, Hanczar, Janodet, and Klaudel]	2020	DL	Supervised/ Unsupervised	SynCAN [Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer]	FCN, CNN, TCN, LSTM, AE			✓	0.01 - 0.3MB	
CAN Frame-Based Attack Detection										
[Nakamura et al.(2021)Nakamura, Takeuchi, Kashima, Kishikawa, Ushio, Haga, and Sasaki]	2021	DL	Unsupervised	Survival Analysis Dataset [Han et al.(2018)Han, Kwak, and Kim]	LightGBM, AE		✓	✓	N/A	
[Nguyen et al.(2024)Nguyen, Cho, and Kim]	2024	DL	Semi-supervised	car-hacking [Song et al.(2020)Song, Woo, and Kim], ROAD [Verma et al.(2020)Verma, Iannaccone, Bridges, Hollifield, Kiy, and Combs]	VAE and AERL	✓	✓	✓	2.542 KB	
[Lin et al.(2021)Lin, Wei, Li, and Long]	2021	DL	Supervised/ Semi-supervised	car-hacking [Song et al.(2020)Song, Woo, and Kim]	DNN and IL			✓	N/A	
[Yang et al.(2022a)Yang, Moubayed, and Shami]	2022	ML	Hybrid	Car-Hacking [Seo et al.(2018)Seo, Song, and Kim], CICIDS2017 [Sharafaldin et al.(2018)Sharafaldin, Lashkari, and Ghorbani]	DT, RF, ET, XGBoost, CL-A-means	✓	✓	✓	2.61 MB	
[Alhunayyan et al.(2024a)Alhunayyan, Javed, and Rana]	2024	DL	Hybrid	Car-Hacking [Seo et al.(2018)Seo, Song, and Kim]	ANN-LSTM-AE	✓	✓	✓	2.98 MB / 253,582	✓

DL: Deep Learning, **FL:** Federated Learning, **M-C:** Multi-class classification.

5.5.4 Limitations of Existing known and Unknown Attacks Detection

Although we have discussed the limitations of each of the previous work in the previous section, a common limitation of most of the proposed approaches pertains to their deployment strategy. The majority of these studies have implemented their IDSs using a traditional centralised learning approach, which requires transmitting large volumes of data to the cloud for both training and testing on the CAN bus. This method raises significant issues, such as privacy concerns, high communication overhead, and longer response times [Chellapandi et al.(2023)Chellapandi, Yuan, Žak, and Wang].

5.6 Evaluation Metrics

In this section, we review all the evaluation metrics used to assess the proposed models 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 categorize 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 [Guidry et al.(2023)Guidry, Sohrab, Gottumukkala, Katragadda, and Gabbouj, Thiruloga et al.(2022)Thiruloga, Kukkal, and Pasricha]. Other relevant metrics include kappa and loss. 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 [Le et al.(2024)Le, Truong, Kim, et al.].

Table 7 shows each evaluation metric used in the reviewed papers. 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 raw data to a centralised server [Li et al.(2020)Li, Fan, Tse, and Lin]. Instead, it transfers model parameters to a centralised server, which aggregates the clients’ models to build a shared global model

[illegible]

Table 7: Evaluation metrics used in existing works

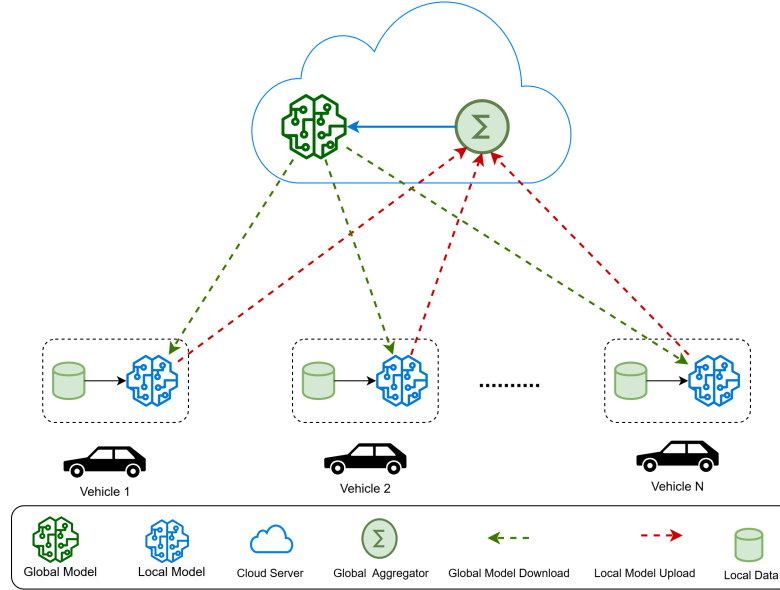


Figure 9: Federated Learning Architecture

[Agrawal et al.(2022b)Agrawal, Sarkar, Aouedi, Yenduri, Piamrat, Alazab, Bhattacharya, Maddikunta, and Gadekallu]. This integration 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 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 [Alsamiri and Alsubhi(2023)]. FL is well-suited for in-vehicle IDSs for several compelling reasons:

- The FL approach preserves 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 allows multiple participants to efficiently develop a robust global model while preserving user data privacy. It enables real-time model updates and data access without the need to communicate with a central server.
- FL reduces latency by avoiding sending raw data to a central server [Agrawal et al.(2022b)Agrawal, Sarkar, Aouedi, Yenduri, Piamrat, Alazab, Bhattacharya, Maddikunta, and Gadekallu].
- Referring to the 2020 guidelines of the International Telecommunication Union [X.1375 Working Group(2020)] for IDS in vehicular networks, an in-vehicle IDS must have the ability to regularly update its set of rules.
- 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 [Althunayyan et al.(2024b)Althunayyan, Javed, Rana, and Spyridopoulos].

As depicted in Figure 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. [Driss et al.(2022)Driss, Almomani, e Huma, and Ahmad] introduced an FL-based framework for detecting attacks in vehicular sensor networks. The authors highlighted the importance of lightweight security solutions, recognising the resource limitations of smart sensing devices in these 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. [Shibly et al.(2022)Shibly, Hossain, Inoue, Taenaka, and Kadobayashi] proposed a personalised FL-based IDS that eliminates the need for 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 did not account for non-IID data distributions.

Yu et al. [Yu et al.(2022)Yu, Hua, Wang, Yang, and Hu] presented an FL-based IDS using LSTM for in-vehicle networks. They leveraged the periodicity of CAN communications to forecast the arbitration IDs of incoming messages. The 11-bit arbitration ID is converted into vectors through one-hot encoding, which are then used by the LSTM to predict the next arbitration ID. The data is equally divided among clients, with each client containing 1,000 instances for training and 200 for testing. A comparison between the FL-based and 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. [Zhang et al.(2023)Zhang, Zeng, and Lin] designed an anomaly detection system using a graph neural network, able to detect CAN bus intrusions in just 3 milliseconds. The IDS utilises a two-stage classifier cascade, with one classifier dedicated to anomaly detection within a single class and the other to classifying attacks into multiple categories. An openmax layer is incorporated into the multi-class classifier to handle novel anomalies from unseen classes.

Yang et al. [Yang et al.(2022b)Yang, Hu, and Yu] developed an IDS for in-vehicle networks using federated deep learning. Their approach capitalises on the periodicity of network messages, incorporates the ConvLSTM model, and trains the model via federated DL. To simulate a non-IID environment, clients were given different numbers of data samples (ranging from 50 to 3500), though details on how the data was distributed among clients and across classes were not specified.

Taslimasa et al. [Taslimasa et al.(2023b)Taslimasa, Dadkhah, Neto, Xiong, Iqbal, Ray, and Ghorbani] introduced ImageFed, a privacy-preserving IDS that employs federated CNNs. To create a non-IID environment, data were allocated to vehicles using a Dirichlet(μ) distribution, with (μ) values varying from 0.1 to 0.7. To assess ImageFed’s resilience, they investigated two potential scenarios that could cause a decline in FL performance: non-IID clients and restricted access to training data.

Longari et al. [Longari et al.(2023)Longari, Pozzoli, Nichelini, Carminati, and Zanero] deployed their proposed IDS, CANDito, presented in Section 5.4, in an FL setting to evaluate its detection efficiency and communication overhead, comparing it to a centralised version of the same algorithm. Experimental results suggest that FL could be a suitable approach in real-world scenarios where data privacy and security cannot be ignored. While the detection capabilities of the federated model are slightly lower than those of the centralised model, it still demonstrates robust performance.

To overcome the challenge of DL models requiring large amounts of data to achieve optimal performance—particularly in the case of CAN bus IDS—Hoang et al. [Hoang et al.(2023)Hoang, Islam, Yim, and Kim] proposed CANPerFL, an IDS that employs a personalised FL approach to aggregate datasets from different car models. Their approach builds a universal model trained on a small amount of data from each manufacturer, providing global knowledge that enhances the performance of individual participants. Experimental results show that the proposed model improves F1 scores by 4% overall compared to baselines. Moreover, it offers significant advantages when the local dataset of each participant is relatively small.

Althunayyan et al. [Althunayyan et al.(2024b)Althunayyan, Javed, Rana, and Spyridopoulos] deployed their proposed IDS from [Althunayyan et al.(2024a)Althunayyan, Javed, and Rana] 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 performance bottlenecks and introducing a single point of failure that compromises robustness and scalability. By incorporating multiple edge aggregators along with the central aggregator, the proposed H-FL mitigates

the risk of single-point failures, enhances scalability, and optimises the distribution of computational load. Experimental results demonstrate that deploying the IDS within the H-FL framework can improve the F1-score by up to 10.63%, effectively overcoming the limitations of edge-FL in terms of dataset diversity and attack coverage.

Table 8 summarises previous work. In cases where the aggregation function is not explicitly stated, as in [Driss et al.(2022)Driss, Almomani, e Huma, and Ahmad, Shibly et al.(2022)Shibly, Hossain, Inoue, Taenaka, and Kadobayashi], it is assumed that FedAvg was employed.

Reference	FL	Non-IID	Aggregation Function	Dataset	FL Implementation
[Taslimasa et al.(2023b)Taslimasa, Dadkhah, Neto, Xiong, Iqbal, Ray, and Ghorbani]	Standard	✓	FedAvg	car-hacking [Song et al.(2020)Song, Woo, and Kim]	PyTorch
[Yu et al.(2022)Yu, Hua, Wang, Yang, and Hu]	Standard	x	FedAvg	HCRL CAN Intrusion Detection [Lee et al.(2017)Lee, Jeong, and Kim]	N/A
[Shibly et al.(2022)Shibly, Hossain, Inoue, Taenaka, and Kadobayashi]	Standard	x	FedAvg	car-hacking [Song et al.(2020)Song, Woo, and Kim], NAIST CAN attack dataset[Hossain et al.(2020b)Hossain, Inoue, Ochiai, Fall, and Kadobayashi]	Keras, TensorFlow
[Zhang et al.(2023)Zhang, Zeng, and Lin]	Standard	N/A	FedAvg, FedProx	READ [Marchetti and Stabli(2018)]	N/A
[Driss et al.(2022)Driss, Almomani, e Huma, and Ahmad]	Standard	x	FedAvg	Car Hacking: Attack & Defence Challenge 2020 [Kang et al.(2021)Kang, Kwak, Lee, Lee, and Kim]	Keras, TensorFlow
[Yang et al.(2022b)Yang, Hu, and Yu]	Standard	✓	FedAvg	HCRL CAN Intrusion Detection [Lee et al.(2017)Lee, Jeong, and Kim]	N/A
[Longari et al.(2023)Longari, Pozzoli, Nichelini, Carminati, and Zanero]	Standard	N/A	FedAvg, FedProx	Recan [Zago et al.(2020)Zago, Longari, Tricarico, Carminati, Pérez, and Zanero]	N/A
[Hoang et al.(2023)Hoang, Islam, Yim, and Kim]	Standard	-	FedAvg	Own	Pytorch, Flower
[Althunayyan et al.(2024b)Althunayyan, Javed, Rana, and Spyridopoulos]	Hierarchical	✓	FedAvg	car-hacking [Song et al.(2020)Song, Woo, and Kim], Car Hacking [Seo et al.(2018)Seo, Song, and Kim]	Flower

Table 8: FL-based IDSs for in-vehicle network

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 the training data on each client varies significantly, leading to differing data distributions among clients [McMahan et al.(2017)McMahan, Moore, Ramage, Hampson, and y Arcas]. In real-world applications, data is typically Non-IID due to variations in user behaviour, preferences, and environments [Li et al.(2022)Li, Diao, Chen, and He]. However, most existing studies do not account for Non-IID data and instead assume data partitions where clients receive either an equal number of samples or samples from all classes (i.e., types of attacks). This assumption contradicts real-world FL scenarios, which inherently involve Non-IID data distributions [Hernandez-Ramos et al.(2023)Hernandez-Ramos, Karopoulos, Chatzoglou, Kouliaridis, Marmol, Gonzalez-Vidal, and Kambourakis], resulting in an unrealistic evaluation of FL-based IDS performance [Zhao et al.(2018)Zhao, Li, Lai, Suda, Civin, and Chandra]. Only a few studies [Yang et al.(2022b)Yang, Hu, and Yu, Taslimasa et al.(2023b)Taslimasa, Dadkhah, Neto, Xiong, Iqbal, Ray, and Ghorbani, Althunayyan et al.(2024b)Althunayyan, Javed, Rana, and Spyridopoulos] have explicitly considered Non-IID data distributions. In [Yang et al.(2022b)Yang, Hu, and Yu], nine candidate clients are assumed, each possessing varying numbers of data samples (50, 100, 150, 1000, 1500, 2000, 2500, 3000, 3500), but the distribution of samples across classes and among clients remains unclear. In contrast, [Taslimasa et al.(2023b)Taslimasa, Dadkhah, Neto, Xiong, Iqbal, Ray, and Ghorbani] and [Althunayyan et al.(2024b)Althunayyan, Javed, Rana, and Spyridopoulos] implement a Non-IID setting by distributing data to vehicles using a *Dirichlet*(μ) distribution, where the (μ) parameter 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

Based on the survey in the previous sections, this section identifies the limitations of existing approaches and explores potential future research directions for enhancing 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 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 fail to capture the full complexity of real-world driving conditions, primarily due to privacy and legal constraints [Said Elsayed et al.(2020)Said Elsayed, Le-Khac, Dev, and Jurcut]. 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 [Aloraini et al.(2024)Aloraini, Javed, and Rana, Li et al.(2021)Li, Lin, and Xiong] have highlighted the vulnerabilities of these systems. Adversarial attacks manipulate input data to deceive models into producing incorrect or misclassified outputs [Li et al.(2021)Li, Lin, and Xiong], posing significant risks to the safety and security of CAVs. From the literature, it is evident that almost no proposed IDS has considered protecting the system from adversarial attacks, except for the work in [Li et al.(2021)Li, Lin, and Xiong], where Li et al. [Li et al.(2021)Li, Lin, and Xiong] developed a defense strategy to protect LSTM-based IDSs from adversarial attacks. Consequently, deploying IDSs without properly evaluating their adversarial robustness not only fails to protect the vehicle but also potentially escalates the risk of vehicle manipulation. Thus, training IDSs on adversarial samples to detect these attacks is a possible solution. Moreover, adapting defense 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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah], anomaly detection models usually suffer from false positives. In critical systems, minimising false alarms is essential for maintaining system reliability. Some existing approaches train biased classifiers to reduce false positives and false negatives, but this shifts the model away from being purely unsupervised. Future research should focus on finding practical solutions that reduce false positives 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 [Axelsson and Sands(2006)]. XAI methods, such as SHapley Additive exPlanations (SHAP) or Local Interpretable Model-agnostic Explanations (LIME), can provide clearer insights into the decision-making process of IDSs, allowing for better responses to alarms and fostering greater trust in AI-driven security systems [Lundberg et al.(2022)Lundberg, Mowla, Abedin, Thar, Mahmood, Gidlund, and Raza]. 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 bridge the gap between different vehicle models and make the system more general across all 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 [Yang et al.(2022b)Yang, Hu, and Yu]. 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 data, driving behaviour, or geographical area to ensure that the FL process remains efficient.
- **Evaluation Metrics:** The majority of the reviewed literature focused on evaluating 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 [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah], making many proposed IDSs impractical for real-world applications. Given the memory constraints of ECUs and the real-time requirements in in-vehicle networks [Kristianto et al.(2024)Kristianto, Lin, and Hwang], an efficient IDS must be lightweight, have a small memory footprint [Zhang et al.(2024)Zhang, Yan, and Ma, Kukkala et al.(2020)Kukkala, Thiruloga, and Pasricha], and satisfy real-time performance requirements. When designing in-vehicle IDS solutions, it is essential to consider the deployment requirements [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar]. The develop-

ment and deployment of IDSs are significantly impacted by the constraints of ECUs in in-vehicle networks, which include limited memory storage, computing power, and bandwidth [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah]. 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 refers to the amount of time required for a trained model to generate predictions on a new data batch [Le et al.(2024)Le, Truong, Kim, et al.]. Latency, on the other hand, is the time taken for a packet to travel from its source to its destination [Yang et al.(2022a)Yang, Moubayed, and Shami]. The United States (US) Department of Transportation states that critical vehicle safety services, such as collision and attack warnings, should operate with a latency of 10 to 100 ms [Abualhou et al.(2016)Abualhou, Shagdar, and Nashashibi]. Meanwhile, Vehicle-to-everything (V2X)-based autonomous and cooperative driving applications require even stricter latency, typically between 10 and 20 ms [Moubayed et al.(2020)Moubayed, Shami, Heidari, Larabi, and Brunner]. Thus, for a vehicle-level IDS, the time required to process each network packet must be less than 10 ms to meet real-time requirements.

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 comprehensive survey of existing ML and DL approaches for building in-vehicle IDSs, focusing on detecting known attacks (38 papers), unknown attacks (27 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 85. 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.

References

- [Aloraini et al.(2024)Aloraini, Javed, and Rana] Fatimah Aloraini, Amir Javed, and Omer Rana. Adversarial attacks on intrusion detection systems in in-vehicle networks of connected and autonomous vehicles. *Sensors*, 24(12): 3848, 2024. doi: <https://doi.org/10.3390/s24123848>.
- [SMMT Driving the Motor Industry(2019)] 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.
- [Pickford et al.(2024)Pickford, Attale, Shaikh, Nguyen, and Harrison] James Pickford, Rasadhi Attale, Siraj Shaikh, Hoang Nga Nguyen, and Lee Harrison. Systematic risk characterisation of hardware threats to automotive systems. *Journal on Autonomous Transportation Systems*, 1(4):1–36, 2024.
- [Al-Jarrah et al.(2019)Al-Jarrah, Maple, Dianati, Oxtoby, and Mouzakitis] Omar Y Al-Jarrah, Carsten Maple, Mehrdad Dianati, David Oxtoby, and Alex Mouzakitis. Intrusion detection systems for intra-vehicle networks: A review. *Ieee Access*, 7:21266–21289, 2019.
- [Althunayyan et al.(2024a)Althunayyan, Javed, and Rana] Muzun Althunayyan, Amir Javed, and Omer Rana. A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning. *Vehicular Communications*, 49:100837, 2024a.
- [Paul and Islam(2021)] Avishek Paul and Md Rabiul 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)*, pages 1–5. IEEE, 2021. doi: <https://doi.org/10.1109/ACMI53878.2021.9528201>.

- [Aliwa et al.(2021)Aliwa, Rana, Perera, and Burnap] Emad Aliwa, Omer Rana, Charith Perera, and Peter Burnap. Cyberattacks and countermeasures for in-vehicle networks. *ACM computing surveys (CSUR)*, 54(1):1–37, 2021. doi: <https://doi.org/10.1145/3431233>.
- [Ltd.(2024)] Upstream Security Ltd. Upstream’s 2024 global automotive cybersecurity report. <https://upstream.auto/reports/global-automotive-cybersecurity-report/#>, 2024. Accessed: 2025-01-18.
- [Young et al.(2019)Young, Zambreno, Olufowobi, and Bloom] Clinton Young, Joseph Zambreno, Habeeb Olufowobi, and Gedare Bloom. Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test*, 36(6):48–55, 2019. doi: <https://doi.org/MDAT.2019.2899062>.
- [Tindell(2023)] Ken Tindell. Can injection: keyless car theft. https://kentindell.github.io/2023/04/03/can-injection/?utm_source=chatgpt.com, 2023. Accessed: 2025-01-18.
- [Golson(2016)] Jordan Golson. Jeep hackers at it again, this time taking control of steering and braking systems. <https://www.theverge.com/2016/8/2/12353186/car-hack-jeep-cherokee-vulnerability-miller-valasek>, 2016. (accessed 1 April 2023).
- [Lab(2018)] Tencent Keen Security Lab. New vehicle security research by keenlab: Experimental security assessment of bmw cars. <https://keenlab.tencent.com/en/2018/05/22/New-CarHacking-Research-by-KeenLab-Experimental-Security-Assessment-of-BMW-Cars/>, 2018. (accessed 10 April 2023).
- [Lab(2020)] Tencent Keen Security Lab. Experimental security assessment on lexus cars. <https://keenlab.tencent.com/en/2020/03/30/Tencent-Keen-Security-Lab-Experimental-Security-Assessment-on-Lexus-Cars/>, 2020. (accessed 10 April 2023).
- [Bertoncello et al.(2021)Bertoncello, Martens, Möller, and Schneiderbauer] Michele Bertoncello, Christopher Martens, Timo Möller, and Tobias Schneiderbauer. Unlocking the full life-cycle value from connected-car data. <https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/unlocking-the-full-life-cycle-value-from-connected-car-data>, 2021. (accessed 6 April 2023).
- [Hoppe et al.(2009)Hoppe, Kiltz, and Dittmann] Tobias Hoppe, Stefan Kiltz, and Jana Dittmann. Applying intrusion detection to automotive it-early insights and remaining challenges. *Journal of Information Assurance and Security (JIAS)*, 4(6):226–235, 2009.
- [Rajapaksha et al.(2023a)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, Madzudzo, and Cheah] Sampath Rajapaksha, Harsha Kalutarage, M Omar Al-Kadri, Andrei Petrovski, Garikayi Madzudzo, and Madeline Cheah. Ai-based intrusion detection systems for in-vehicle networks: A survey. *ACM Computing Surveys*, 55(11):1–40, 2023a. doi: <https://doi.org/10.1145/3570954>.
- [Liu et al.(2020)Liu, Zhang, Song, and Letaief] Lumin Liu, Jun Zhang, SH Song, and Khaled B Letaief. Client-edge-cloud hierarchical federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020. doi: <https://doi.org/10.1109/ICC40277.2020.9148862>.
- [Ahmad et al.(2024)Ahmad, Han, Jolfaei, Jabbar, Ibrar, Erbad, Song, and Alkhrijah] Usman Ahmad, Mu Han, Alireza Jolfaei, Sohail Jabbar, Muhammad Ibrar, Aiman Erbad, Houbing Herbert Song, and Yazeed Alkhrijah. A comprehensive survey and tutorial on smart vehicles: Emerging technologies, security issues, and solutions using machine learning. *IEEE Transactions on Intelligent Transportation Systems*, 2024.
- [Boumiza and Braham(2017)] Safa Boumiza and Rafik Braham. Intrusion threats and security solutions for autonomous vehicle networks. In *2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)*, pages 120–127. IEEE, 2017. doi: <https://DOI.org/10.1109/AICCSA.2017.42>.
- [Kumar and Ramesh(2014)] B Vinodh Kumar and J Ramesh. Automotive in vehicle network protocols. In *2014 International Conference on Computer Communication and Informatics*, pages 1–5. IEEE, 2014. doi: <https://DOI.org/10.1109/ICCCI.2014.6921836>.
- [Lokman et al.(2019)Lokman, Othman, and Abu-Bakar] Siti-Farhana Lokman, Abu Talib Othman, and Muhammad-Husaini Abu-Bakar. Intrusion detection system for automotive controller area network (can) bus system: a review. *EURASIP Journal on Wireless Communications and Networking*, 2019:1–17, 2019. doi: <https://doi.org/10.1186/s13638-019-1484-3>.
- [Carsten et al.(2015)Carsten, Andel, Yampolskiy, and McDonald] Paul Carsten, Todd R Andel, Mark Yampolskiy, and Jeffrey T McDonald. In-vehicle networks: Attacks, vulnerabilities, and proposed solutions. In *Proceedings of the 10th Annual Cyber and Information Security Research Conference*, pages 1–8, 2015.

- [Liu et al.(2017)Liu, Zhang, Sun, and Shi] Jiajia Liu, Shubin Zhang, Wen Sun, and Yongpeng Shi. In-vehicle network attacks and countermeasures: Challenges and future directions. *IEEE Network*, 31(5):50–58, 2017.
- [Dupont et al.(2019a)Dupont, den Hartog, Etalle, and Lekidis] Guillaume Dupont, Jerry den Hartog, Sandro Etalle, and Alexios Lekidis. A survey of network intrusion detection systems for controller area network. In *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pages 1–6. IEEE, 2019a.
- [Wu et al.(2019)Wu, Li, Xie, An, Bai, Zhou, and Li] Wufei Wu, Renfa Li, Guoqi Xie, Jiyao An, Yang Bai, Jia Zhou, and Keqin Li. A survey of intrusion detection for in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):919–933, 2019.
- [Limbsiya et al.(2022)Limbsiya, Teng, Chattopadhyay, and Zhou] Trupil Limbsiya, Ko Zheng Teng, Sudipta Chattopadhyay, and Jianying Zhou. A systematic survey of attack detection and prevention in connected and autonomous vehicles. *Vehicular Communications*, 37:100515, 2022.
- [Checkoway et al.(2011)Checkoway, McCoy, Kantor, Anderson, Shacham, Savage, Koscher, Czeskis, Roesner, and Kohno] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX Security Symposium (USENIX Security 11)*, San Francisco, CA, 2011. USENIX Association.
- [Koscher et al.(2010)Koscher, Czeskis, Roesner, Patel, Kohno, Checkoway, McCoy, Kantor, Anderson, Shacham, et al.] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, et al. Experimental security analysis of a modern automobile. In *2010 IEEE symposium on security and privacy*, pages 447–462. IEEE, 2010.
- [Chockalingam et al.(2016)Chockalingam, Larson, Lin, and Nofzinger] Valliappa Chockalingam, Ian Larson, Daniel Lin, and Spencer Nofzinger. Detecting attacks on the can protocol with machine learning. *Annu EECS*, 558(7), 2016.
- [Woo et al.(2014)Woo, Jo, and Lee] Samuel Woo, Hyo Jin Jo, and Dong Hoon Lee. A practical wireless attack on the connected car and security protocol for in-vehicle can. *IEEE Transactions on intelligent transportation systems*, 16(2):993–1006, 2014.
- [Hoang and Kim(2022)] Thien-Nu Hoang and Daehee Kim. Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *Vehicular Communications*, 38, 2022. doi: <https://doi.org/10.1016/j.vehcom.2022.100520>.
- [Lee et al.(2017)Lee, Jeong, and Kim] Hyunsung Lee, Seong Hoon Jeong, and Huy Kang 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)*, pages 57–5709. IEEE, 2017. doi: <https://doi.org/10.1109/PST.2017.00017>.
- [Cho and Shin(2016)] Kyong-Tak Cho and Kang G Shin. Error handling of in-vehicle networks makes them vulnerable. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1044–1055, 2016.
- [Hossain et al.(2020a)Hossain, Inoue, Ochiai, Fall, and Kadobayashi] Md Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. An effective in-vehicle can bus intrusion detection system using cnn deep learning approach. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020a. doi: <https://doi.org/10.1109/GLOBECOM42002.2020.9322395>.
- [Fowler et al.(2018)Fowler, Bryans, Shaikh, and Wooderson] Daniel S Fowler, Jeremy Bryans, Siraj Ahmed Shaikh, and Paul Wooderson. Fuzz testing for automotive cyber-security. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 239–246. IEEE, 2018.
- [Iehira et al.(2018)Iehira, Inoue, and Ishida] Kazuki Iehira, Hiroyuki Inoue, and Kenji 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)*, pages 1–4. IEEE, 2018.
- [Hossain et al.(2020b)Hossain, Inoue, Ochiai, Fall, and Kadobayashi] Md Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki Kadobayashi. Lstm-based intrusion detection system for in-vehicle can bus communications. *IEEE Access*, 8:185489–185502, 2020b. doi: <https://doi.org/10.1109/ACCESS.2020.3029307>.
- [Jo and Choi(2021)] Hyo Jin Jo and Wonsuk Choi. A survey of attacks on controller area networks and corresponding countermeasures. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6123–6141, 2021.

- [Karopoulos et al.(2022)Karopoulos, Kambourakis, Chatzoglou, Hernández-Ramos, and Kouliaridis] Georgios Karopoulos, Georgios Kambourakis, Efstratios Chatzoglou, José L Hernández-Ramos, and Vasileios Kouliaridis. Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy. *Electronics*, 11(7):1072, 2022.
- [Tomlinson et al.(2018)Tomlinson, Bryans, and Shaikh] Andrew Tomlinson, Jeremy Bryans, and Siraj Ahmed Shaikh. Towards viable intrusion detection methods for the automotive controller area network. In *2nd ACM Computer Science in Cars Symposium*, pages 1–9, 2018.
- [Rajbahadur et al.(2018)Rajbahadur, Malton, Walenstein, and Hassan] Gopi Krishnan Rajbahadur, Andrew J Malton, Andrew Walenstein, and Ahmed E Hassan. A survey of anomaly detection for connected vehicle cybersecurity and safety. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 421–426. IEEE, 2018.
- [Loukas et al.(2019)Loukas, Karapistoli, Panaousis, Sarigiannidis, Bezemskij, and Vuong] George Loukas, Eirini Karapistoli, Emmanouil Panaousis, Panagiotis Sarigiannidis, Anatolij Bezemskij, and Tuan Vuong. A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles. *Ad Hoc Networks*, 84:124–147, 2019.
- [Quadar et al.(2024)Quadar, Chehri, Debaque, Ahmed, and Jeon] Nordine Quadar, Abdellah Chehri, Benoit Debaque, Imran Ahmed, and Gwangil Jeon. Intrusion detection systems in automotive ethernet networks: challenges, opportunities and future research trends. *IEEE Internet of Things Magazine*, 7(2):62–68, 2024.
- [Lampe and Meng(2023a)] Brooke Lampe and Weizhi Meng. Intrusion detection in the automotive domain: A comprehensive review. *IEEE Communications Surveys & Tutorials*, 25(4):2356–2426, 2023a.
- [Nagarajan et al.(2023)Nagarajan, Mansourian, Shahid, Jaekel, Saini, Zhang, and Kneppers] Jay Nagarajan, Pegah Mansourian, Muhammad Anwar Shahid, Arunita Jaekel, Ikjot Saini, Ning Zhang, and Marc Kneppers. Machine learning based intrusion detection systems for connected autonomous vehicles: A survey. *Peer-to-Peer Networking and Applications*, pages 1–33, 2023. doi: <https://doi.org/10.1007/s12083-023-01508-7>.
- [Lampe and Meng(2023b)] Brooke Lampe and Weizhi Meng. A survey of deep learning-based intrusion detection in automotive applications. *Expert Systems with Applications*, 221:119771, 2023b. doi: <https://doi.org/10.1016/j.eswa.2023.119771>.
- [Almehdhar et al.(2024)Almehdhar, Albaseer, Khan, Abdallah, Menouar, Al-Kuwari, and Al-Fuqaha] Mohammed Almehdhar, Abdullatif Albaseer, Muhammad Asif Khan, Mohamed Abdallah, Hamid Menouar, Saif Al-Kuwari, and Ala Al-Fuqaha. Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. *IEEE Open Journal of Vehicular Technology*, 2024.
- [Taslimasa et al.(2023a)Taslimasa, Dadkhah, Neto, Xiong, Ray, and Ghorbani] Hamideh Taslimasa, Sajjad Dadkhah, Euclides Carlos Pinto Neto, Pulei Xiong, Suprio Ray, and Ali A Ghorbani. Security issues in internet of vehicles (ioV): A comprehensive survey. *Internet of Things*, 22:100809, 2023a.
- [Chellapandi et al.(2023)Chellapandi, Yuan, Żak, and Wang] Vishnu Pandi Chellapandi, Liangqi Yuan, Stanislaw H Żak, and Ziran Wang. A survey of federated learning for connected and automated vehicles. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2485–2492. IEEE, 2023. doi: [10.48550/arXiv.2303.10677](https://doi.org/10.48550/arXiv.2303.10677).
- [Kitchenham and Brereton(2013)] Barbara Kitchenham and Pearl Brereton. A systematic review of systematic review process research in software engineering. *Information and software technology*, 55(12):2049–2075, 2013.
- [Alhirabi et al.(2021)Alhirabi, Rana, and Perera] Nada Alhirabi, Omer Rana, and Charith Perera. Security and privacy requirements for the internet of things: A survey. *ACM Transactions on Internet of Things*, 2(1):1–37, 2021.
- [Wohlin(2014)] Claes 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*, pages 1–10, 2014.
- [Hernandez-Ramos et al.(2023)Hernandez-Ramos, Karopoulos, Chatzoglou, Kouliaridis, Marmol, Gonzalez-Vidal, and Kambourakis] Jose L Hernandez-Ramos, Georgios Karopoulos, Efstratios Chatzoglou, Vasileios Kouliaridis, Enrique Marmol, Aurora Gonzalez-Vidal, and Georgios Kambourakis. Intrusion detection based on federated learning: a systematic review. *arXiv preprint arXiv:2308.09522*, 2023. doi: [10.48550/arXiv.2308.09522](https://doi.org/10.48550/arXiv.2308.09522).

- [Song et al.(2020)Song, Woo, and Kim] Hyun Min Song, Jiyoung Woo, and Huy Kang Kim. In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications*, 21:100198, 2020. doi: 10.1016/j.vehcom.2019.100198.
- [Refat et al.(2022)Refat, Elkhail, Hafeez, and Malik] Rafi Ud Daula Refat, Abdulrahman Abu Elkhail, Azeem Hafeez, and Hafiz 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*, pages 730–748. Springer, 2022.
- [Nandam et al.(2022)Nandam, Vamshi, and Sucharitha] Srinivasa Rao Nandam, Adouthu Vamshi, and Inapanuri Sucharitha. Can intrusion detection using long short-term memory (Lstm). In *Computer Communication, Networking and IoT: Proceedings of 5th ICICC 2021, Volume 2*, pages 295–302. Springer, 2022.
- [Rangsikunpum et al.(2024a)Rangsikunpum, Amiri, and Ost] Auangkun Rangsikunpum, Sam Amiri, and Luciano Ost. An fpga-based intrusion detection system using binarised neural network for can bus systems. In *2024 IEEE International Conference on Industrial Technology (ICIT)*, pages 1–6. IEEE, 2024a.
- [Wu and Tao(2024)] Yuxi Wu and Xiaodong 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)*, pages 240–245. IEEE, 2024.
- [Kang and Kang(2016)] Min-Joo Kang and Je-Won Kang. Intrusion detection system using deep neural network for in-vehicle network security. *PloS one*, 11(6):e0155781, 2016.
- [Martinelli et al.(2017)Martinelli, Mercaldo, Nardone, and Santone] Fabio Martinelli, Francesco Mercaldo, Vittoria Nardone, and Antonella Santone. Car hacking identification through fuzzy logic algorithms. In *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pages 1–7. IEEE, 2017.
- [Fenzl et al.(2021)Fenzl, Rieke, and Dominik] Florian Fenzl, Roland Rieke, and Andreas Dominik. In-vehicle detection of targeted can bus attacks. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pages 1–7, 2021.
- [Samir et al.(2024)Samir, Raissa, Touati, Hadded, and Ghazzai] Said Ben Hassane Samir, Martin Raissa, Haifa Touati, Mohamed Hadded, and Hakim Ghazzai. Machine learning-based intrusion detection for securing in-vehicle can bus communication. *SN Computer Science*, 5(8):1082, 2024.
- [Le et al.(2024)Le, Truong, Kim, et al.] Tien-Dat Le, Hoang Bao Huy Truong, Daehee Kim, et al. Multi-classification in-vehicle intrusion detection system using packet-and sequence-level characteristics from time-embedded transformer with autoencoder. *Knowledge-Based Systems*, 299:112091, 2024.
- [Zhang et al.(2024)Zhang, Yan, and Ma] Linxi Zhang, Xuke Yan, and Di Ma. Efficient and effective in-vehicle intrusion detection system using binarized convolutional neural network. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 2299–2307. IEEE, 2024.
- [Sami et al.(2020)Sami, Ibarra, Esparza, Al-Jufout, Aliasgari, and Mozumdar] Muhammad Sami, Matthew Ibarra, Anamaria C Esparza, Saleh Al-Jufout, Mehrdad Aliasgari, and Mohammad 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)*, pages 1–6. IEEE, 2020.
- [Aksu and Aydin(2022)] Dogukan Aksu and Muhammed Ali 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. *Computers & Security*, 118:102717, 2022.
- [Boumiza and Braham(2019)] Safa Boumiza and Rafik 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)*, pages 1–6. IEEE, 2019.
- [Park and Choi(2020)] Seunghyun Park and Jin-Young Choi. Hierarchical anomaly detection model for in-vehicle networks using machine learning algorithms. *Sensors*, 20(14):3934, 2020.
- [Zhang et al.(2020)Zhang, Cui, Cheng, and Zhang] Xing Zhang, Xiaotong Cui, Kefei Cheng, and Liang Zhang. A convolutional encoder network for intrusion detection in controller area networks. In *2020 16th International Conference on Computational Intelligence and Security (CIS)*, pages 366–369. IEEE, 2020.
- [Minawi et al.(2020)Minawi, Whelan, Almeahmadi, and El-Khatib] Omar Minawi, Jason Whelan, Abdulaziz Almeahmadi, and Khalil 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*, pages 41–47, 2020.
- [Alfardus and Rawat(2021)] Asma Alfardus and Danda 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)*, pages 0944–0949. IEEE, 2021. doi: 10.1109/UEMCON53757.2021.9666745.
- [NasrEldin et al.(2021)] NasrEldin, Bahaa-Eldin, and Sobh] Ahmed NasrEldin, Ayman M Bahaa-Eldin, and Mohamed A Sobh. In-vehicle intrusion detection based on deep learning attention technique. In *2021 16th International Conference on Computer Engineering and Systems (ICCES)*, pages 1–7. IEEE, 2021.
- [Alalwany and Mahgoub(2022)] Easa Alalwany and Imad Mahgoub. Classification of normal and malicious traffic based on an ensemble of machine learning for a vehicle can-network. *Sensors*, 22(23):9195, 2022.
- [Ding et al.(2022)] Ding, Zhu, Xie, and Lin] Defeng Ding, Lu Zhu, Jiaying Xie, and Jiaying Lin. In-vehicle network intrusion detection system based on bi-lstm. In *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*, pages 580–583. IEEE, 2022.
- [Kishore et al.(2024)] Kishore, Rao, Nayak, and Behera] Ch Ravi Kishore, D Chandrasekhar Rao, Janmenjoy Nayak, and HS Behera. Intelligent intrusion detection framework for anomaly-based can bus network using bidirectional long short-term memory. *Journal of The Institution of Engineers (India): Series B*, pages 1–24, 2024.
- [Chougule et al.(2024)] Chougule, Kulkarni, Alladi, Chamola, and Yu] Amit Chougule, Ishan Kulkarni, Tejasvi Alladi, Vinay Chamola, and Fei Richard Yu. Hybridsecnet: In-vehicle security on controller area networks through a hybrid two-step lstm-cnn model. *IEEE Transactions on Vehicular Technology*, 2024.
- [Alalwany and Mahgoub(2024)] Easa Alalwany and Imad Mahgoub. An effective ensemble learning-based real-time intrusion detection scheme for an in-vehicle network. *Electronics*, 13(5):919, 2024.
- [Basavaraj and Tayeb(2022)] Dheeraj Basavaraj and Shahab Tayeb. Towards a lightweight intrusion detection framework for in-vehicle networks. *Journal of Sensor and Actuator Networks*, 11(1):6, 2022.
- [Gao et al.(2023)] Gao, Huang, Liu, Du, and Zhang] Kai Gao, Hao Huang, Linhong Liu, Ronghua Du, and Jinlai 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)*, pages 809–816. IEEE, 2023.
- [Kalkan and Sahingoz(2020)] Soner Can Kalkan and Ozgur Koray 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)*, pages 1–6. IEEE, 2020.
- [Gou et al.(2023)] Gou, Zhang, and Zhang] Wanting Gou, Haodi Zhang, and Ronghui Zhang. Multi-classification and tree-based ensemble network for the intrusion detection system in the internet of vehicles. *Sensors*, 23(21): 8788, 2023.
- [Ma et al.(2022)] Ma, Cao, Mi, Huang, Liu, and Li] Haoyu Ma, Jianqiu Cao, Bo Mi, Darong Huang, Yang Liu, and Shaoqian Li. A gru-based lightweight system for can intrusion detection in real time. *Security and Communication Networks*, 2022(1):5827056, 2022.
- [Khan et al.(2024)] Khan, Javed, Iqbal, Asim, and Awad] Muneeb Hassan Khan, Abdul Rehman Javed, Zafar Iqbal, Muhammad Asim, and Ali Ismail Awad. Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning. *Computers & Security*, 139:103712, 2024.
- [Lin et al.(2022)] Lin, Wang, Chao, Lin, and Chen] Hsiao-Chung Lin, Ping Wang, Kuo-Ming Chao, Wen-Hui Lin, and Jia-Hong Chen. Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks. *Electronics*, 11(14):2180, 2022.
- [Hossain et al.(2020c)] Hossain, Inoue, Ochiai, Fall, and Kadobayashi] Md Delwar Hossain, Hiroyuki Inoue, Hideya Ochiai, Doudou Fall, and Youki 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)*, pages 10–17. IEEE, 2020c.
- [Casillo et al.(2019)] Casillo, Coppola, De Santo, Pascale, and Santonicola] Mario Casillo, Simone Coppola, Massimo De Santo, Francesco Pascale, and Emanuele Santonicola. Embedded intrusion detection system for detecting

- attacks over can-bus. In *2019 4th International Conference on System Reliability and Safety (ICSRS)*, pages 136–141. IEEE, 2019.
- [Nazeer et al.(2024)Nazeer, Alasiry, Qayyum, Madhan, Patil, and Srilatha] Mohd Nazeer, Areej Alasiry, Mohammed Qayyum, Vemana Karunakar Madhan, Gouri Patil, and Pulipati Srilatha. Enhancing cyber security in autonomous vehicles: A hybrid xg boost-deep learning approach for intrusion detection in the can bus. *Journal Européen des Systèmes Automatisés*, 57(5), 2024.
- [Nguyen et al.(2023)Nguyen, Nam, and Kim] Trieu Phong Nguyen, Heungwoo Nam, and Daehee Kim. Transformer-based attention network for in-vehicle intrusion detection. *IEEE Access*, 11:55389–55403, 2023.
- [Seo et al.(2018)Seo, Song, and Kim] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. Gids: Gan based intrusion detection system for in-vehicle network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6. IEEE, 2018. doi: <https://doi.org/10.1109/PST.2018.8514157>.
- [Verma et al.(2020)Verma, Iannacone, Bridges, Hollifield, Kay, and Combs] Miki E Verma, Michael D Iannacone, Robert A Bridges, Samuel C Hollifield, Bill Kay, and Frank 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.
- [Sami(2019)] Muhammad Sami. Intrusion detection in can bus, 2019. URL <https://dx.doi.org/10.21227/24m9-a446>.
- [Taylor et al.(2018)Taylor, Leblanc, and Japkowicz] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz. Probing the limits of anomaly detectors for automobiles with a cyberattack framework. *IEEE Intelligent Systems*, 33(2):54–62, 2018.
- [Han et al.(2018)Han, Kwak, and Kim] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. Anomaly intrusion detection method for vehicular networks based on survival analysis. *Vehicular communications*, 14:52–63, 2018.
- [Kang et al.(2021)Kang, Kwak, Lee, Lee, Lee, and Kim] Hyunjae Kang, Byung Il Kwak, Young Hun Lee, Haneol Lee, Hwejae Lee, and Huy Kang Kim. Car hacking and defense competition on in-vehicle network. In *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, volume 2021, page 25, 2021. doi: 10.14722/autosec.2021.23035.
- [Dupont et al.(2019b)Dupont, Lekidis, den Hartog, and Etalle] Guillaume Dupont, Alexios Lekidis, J. (Jerry) den Hartog, and S. (Sandro) Etalle. Automotive controller area network (can) bus intrusion dataset v2, 2019b. URL https://data.4tu.nl/articles/_/12696950/2.
- [Hacking and (HCRL)(2019)] Hacking and Countermeasure Research Lab (HCRL). In-vehicle network intrusion detection challenge, 2019. URL <https://sites.google.com/hksecurity.net/hcrl/Datasets/datachallenge2019/car>.
- [Said Elsayed et al.(2020)Said Elsayed, Le-Khac, Dev, and Jurecut] Mahmoud Said Elsayed, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurecut. Network anomaly detection using lstm based autoencoder. In *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pages 37–45, 2020.
- [Vikram et al.(2020)] Aditya Vikram et al. Anomaly detection in network traffic using unsupervised machine learning approach. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 476–479. IEEE, 2020. doi: <https://doi.org/10.1109/ICCES48766.2020.9137987>.
- [Pratomo et al.(2018)Pratomo, Burnap, and Theodorakopoulos] Baskoro Adi Pratomo, Pete Burnap, and George 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)*, pages 1–8. IEEE, 2018. doi: 10.1109/CyberSecPODS.2018.8560678.
- [Avatefipour et al.(2019)Avatefipour, Al-Sumaiti, El-Sherbeeny, Awwad, Elmeligy, Mohamed, and Malik] Omid Avatefipour, Ameena Saad Al-Sumaiti, Ahmed M El-Sherbeeny, Emad Mahrous Awwad, Mohammed A Elmeligy, Mohamed A Mohamed, and Hafiz Malik. An intelligent secured framework for cyberattack detection in electric vehicles’ can bus using machine learning. *Ieee Access*, 7:127580–127592, 2019.
- [Rajapaksha et al.(2022)Rajapaksha, Kalutarage, Al-Kadri, Madzudzo, and Petrovski] Sampath Rajapaksha, Harsha Kalutarage, M Omar Al-Kadri, Garikayi Madzudzo, and Andrei 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)*, volume 700, pages 309–330. IEEE, 2022.

- [Khandelwal and Shreejith(2023)] Shashwat Khandelwal and Shanker 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)*, pages 139–146. IEEE, 2023.
- [Guidry et al.(2023)Guidry, Sohrab, Gottumukkala, Katragadda, and Gabbouj] Jake Guidry, Fahad Sohrab, Raju Gottumukkala, Satya Katragadda, and Moncef Gabbouj. One-class classification for intrusion detection on vehicular networks. In *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1176–1182. IEEE, 2023.
- [Balaji and Ghaderi(2021)] Prashanth Balaji and Majid Ghaderi. Neurocan: Contextual anomaly detection in controller area networks. In *2021 IEEE International Smart Cities Conference (ISC2)*, pages 1–7. IEEE, 2021.
- [Sun et al.(2021)Sun, Chen, Weng, Liu, and Geng] Heng Sun, Miaomiao Chen, Jian Weng, Zhiquan Liu, and Guang-gang Geng. Anomaly detection for in-vehicle network using cnn-lstm with attention mechanism. *IEEE Transactions on Vehicular Technology*, 70(10):10880–10893, 2021.
- [Thiruloga et al.(2022)Thiruloga, Kukkala, and Pasricha] Sooryaa Vignesh Thiruloga, Vipin Kumar Kukkala, and Sudeep 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)*, pages 326–331. IEEE, 2022.
- [Wei et al.(2022)Wei, Wang, Dai, Li, and He] Pengcheng Wei, Bo Wang, Xiaojun Dai, Li Li, and Fangcheng He. A novel intrusion detection model for the can bus packet of in-vehicle network based on attention mechanism and autoencoder. *Digital Communications and Networks*, 2022. doi: <https://doi.org/10.1016/j.dcan.2022.04.021>.
- [Mansourian et al.(2023)Mansourian, Zhang, Jaekel, Zamanirafe, and Kneppers] Pegah Mansourian, Ning Zhang, Arunita Jaekel, Mina Zamanirafe, and Marc Kneppers. Anomaly detection for connected autonomous vehicles using lstm and gaussian naïve bayes. In *International Conference on Wireless and Satellite Systems*, pages 31–43. Springer, 2023.
- [Zhao et al.(2022a)Zhao, Xun, Liu, and Ma] Yilin Zhao, Yijie Xun, Jiajia Liu, and Siyu Ma. Gvids: A reliable vehicle intrusion detection system based on generative adversarial network. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pages 4310–4315. IEEE, 2022a.
- [Qin et al.(2021)Qin, Yan, and Ji] Hongmao Qin, Mengru Yan, and Haojie Ji. Application of controller area network (can) bus anomaly detection based on time series prediction. *Vehicular Communications*, 27:100291, 2021.
- [Khan et al.(2021)Khan, Moustafa, Pi, Haider, Li, and Jolfaei] Izhar Ahmed Khan, Nour Moustafa, Dechang Pi, Waqas Haider, Bentian Li, and Alireza Jolfaei. An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(12):25469–25478, 2021.
- [Kristianto et al.(2024)Kristianto, Lin, and Hwang] Edy Kristianto, Po-Ching Lin, and Ren-Hung Hwang. Sustainable and lightweight domain-based intrusion detection system for in-vehicle network. *Sustainable Computing: Informatics and Systems*, 41:100936, 2024.
- [Cobilean et al.(2023)Cobilean, Mavikumbure, Wickramasinghe, Varghese, Pennington, and Manic] Victor Cobilean, Harindra S Mavikumbure, Chathurika S Wickramasinghe, Benny J Varghese, Timothy Pennington, and Milos Manic. Anomaly detection for in-vehicle communication using transformers. In *IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6. IEEE, 2023.
- [Narasimhan et al.(2021)Narasimhan, Ravi, and Mohammad] Harini Narasimhan, Vinayakumar Ravi, and Nazeeruddin Mohammad. Unsupervised deep learning approach for in-vehicle intrusion detection system. *IEEE Consumer Electronics Magazine*, 12(1):103–108, 2021.
- [Wang and Mo(2021)] Jie Wang and Xiuliang 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)*, pages 1558–1564. IEEE, 2021.
- [Agrawal et al.(2022a)Agrawal, Alladi, Agrawal, Chamola, and Benslimane] Kushagra Agrawal, Tejasvi Alladi, Ayush Agrawal, Vinay Chamola, and Abderrahim Benslimane. Novelads: A novel anomaly detection system for intra-vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22596–22606, 2022a.

- [Kukkala et al.(2020)Kukkala, Thiruloga, and Pasricha] Vipin Kumar Kukkala, Sooryaa Vignesh Thiruloga, and Sudeep Pasricha. Indra: Intrusion detection using recurrent autoencoders in automotive embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11):3698–3710, 2020. doi: <https://doi.org/10.1109/TCAD.2020.3012749>.
- [Shi et al.(2024)Shi, Xie, Dong, Jiang, and Jin] Jiahao Shi, Zhijun Xie, Li Dong, Xianliang Jiang, and Xing Jin. Ids-dec: A novel intrusion detection for can bus traffic based on deep embedded clustering. *Vehicular Communications*, 49:100830, 2024.
- [Longari et al.(2020)Longari, Valcarcel, Zago, Carminati, and Zanero] Stefano Longari, Daniel Humberto Nova Valcarcel, Mattia Zago, Michele Carminati, and Stefano Zanero. Cannolo: An anomaly detection system based on lstm autoencoders for controller area network. *IEEE Transactions on Network and Service Management*, 18(2): 1913–1924, 2020.
- [Longari et al.(2023)Longari, Pozzoli, Nichelini, Carminati, and Zanero] Stefano Longari, Carlo Alberto Pozzoli, Alessandro Nichelini, Michele Carminati, and Stefano Zanero. Candito: improving payload-based detection of attacks on controller area networks. In *International Symposium on Cyber Security, Cryptology, and Machine Learning*, pages 135–150. Springer, 2023.
- [Hanselmann et al.(2020)Hanselmann, Strauss, Dormann, and Ulmer] Markus Hanselmann, Thilo Strauss, Katharina Dormann, and Holger Ulmer. Canet: An unsupervised intrusion detection system for high dimensional can bus data. *Ieee Access*, 8:58194–58205, 2020.
- [Kishore et al.(2022)Kishore, Rao, and Behera] Ch Ravi Kishore, D Chandrasekhar Rao, and HS Behera. Deep learning approach for anomaly detection in can bus network: An intelligent lstm-based intrusion detection system. In *International Conference on Computational Intelligence in Pattern Recognition*, pages 531–544. Springer, 2022.
- [Rajapaksha et al.(2023b)Rajapaksha, Kalutarage, Al-Kadri, Petrovski, and Madzudzo] Sampath Rajapaksha, Harsha Kalutarage, M Omar Al-Kadri, Andrei Petrovski, and Garikayi Madzudzo. Beyond vanilla: Improved autoencoder-based ensemble in-vehicle intrusion detection system. *Journal of information security and applications*, 77:103570, 2023b.
- [Kim et al.(2023)Kim, Kim, and You] Taeguen Kim, Jiyeon Kim, and Ilseon You. An anomaly detection method based on multiple lstm-autoencoder models for in-vehicle network. *Electronics*, 12(17):3543, 2023.
- [Jo and Kim(2024)] Hyunjun Jo and Deok-Hwan Kim. Intrusion detection using transformer in controller area network. *IEEE Access*, 2024.
- [Xiao et al.(2019)Xiao, Wu, and Li] Junchao Xiao, Hao Wu, and Xiangxue 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)*, pages 1390–1397. IEEE, 2019.
- [Consortium()] Crash Reconstruction Research Consortium. Dodge can messages. <https://www.engr.colostate.edu/~jdaily/tucrrc/DodgeCAN.html>. (accessed 10 October 2024).
- [Song and Kim(2020)] Hyun Min Song and Huy Kang Kim. Discovering can specification using on-board diagnostics. *IEEE Design & Test*, 38(3):93–103, 2020.
- [Zago et al.(2020)Zago, Longari, Tricarico, Carminati, Pérez, Pérez, and Zanero] Mattia Zago, Stefano Longari, Andrea Tricarico, Michele Carminati, Manuel Gil Pérez, Gregorio Martínez Pérez, and Stefano Zanero. Recan-dataset for reverse engineering of controller area networks. *Data in brief*, 29:105149, 2020.
- [Zhao et al.(2022b)Zhao, Chen, Gu, Luan, Zeng, and Chakraborty] Qingling Zhao, Mingqiang Chen, Zonghua Gu, Siyu Luan, Haibo Zeng, and Samarjit Chakraborty. Can bus intrusion detection based on auxiliary classifier gan and out-of-distribution detection. *ACM Transactions on Embedded Computing Systems (TECS)*, 21(4): 1–30, 2022b. doi: <https://doi.org/10.1145/3540198>.
- [Zhang et al.(2019)Zhang, Li, Zhang, Li, and Li] Jiayan Zhang, Fei Li, Haoxi Zhang, Ruxiang Li, and Yalin Li. Intrusion detection system using deep learning for in-vehicle security. *Ad Hoc Networks*, 95:101974, 2019. doi: 10.1016/j.adhoc.2019.101974.
- [Yang et al.(2022a)Yang, Moubayed, and Shami] Li Yang, Abdallah Moubayed, and Abdallah Shami. Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles. *IEEE Internet of Things Journal*, 9(1): 616–632, 2022a. doi: <https://doi.org/10.1109/JIOT.2021.3084796>.

- [Nakamura et al.(2021)Nakamura, Takeuchi, Kashima, Kishikawa, Ushio, Haga, and Sasaki] Shu Nakamura, Koh Takeuchi, Hisashi Kashima, Takeshi Kishikawa, Takashi Ushio, Tomoyuki Haga, and Takamitsu Sasaki. In-vehicle network attack detection across vehicle models: A supervised-unsupervised hybrid approach. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1286–1291. IEEE, 2021.
- [Rangsikunpum et al.(2024b)Rangsikunpum, Amiri, and Ost] Auangkun Rangsikunpum, Sam Amiri, and Luciano Ost. Bids: An efficient intrusion detection system for in-vehicle networks using a two-stage binarised neural network on low-cost fpga. *Journal of Systems Architecture*, 156:103285, 2024b.
- [Han et al.(2021)Han, Kwak, and Kim] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network. *IEEE Transactions on Information Forensics and Security*, 16:2941–2956, 2021.
- [Gherbi et al.(2020)Gherbi, Hanczar, Janodet, and Klaudel] Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, and Witold 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*, pages 50–58. Springer, 2020.
- [Nguyen et al.(2024)Nguyen, Cho, and Kim] Trieu-Phong Nguyen, Jeongho Cho, and Daehee Kim. Semi-supervised intrusion detection system for in-vehicle networks based on variational autoencoder and adversarial reinforcement learning. *Knowledge-Based Systems*, 304:112563, 2024.
- [Lin et al.(2021)Lin, Wei, Li, and Long] Jiaying Lin, Yehua Wei, Wenjia Li, and Jing Long. Intrusion detection system based on deep neural network and incremental learning for in-vehicle can networks. In *International Conference on Ubiquitous Security*, pages 255–267. Springer, 2021.
- [Kocher and Kumar(2021)] Geeta Kocher and Gulshan Kumar. Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, 25(15):9731–9763, 2021. doi: <https://doi.org/10.1007/s00500-021-05893-0>.
- [Li and Vorobeychik(2014)] Bo Li and Yevgeniy Vorobeychik. Feature cross-substitution in adversarial classification. *Advances in neural information processing systems*, 27, 2014.
- [Zhang et al.(2015)Zhang, Chan, Biggio, Yeung, and Roli] Fei Zhang, Patrick PK Chan, Battista Biggio, Daniel S Yeung, and Fabio Roli. Adversarial feature selection against evasion attacks. *IEEE transactions on cybernetics*, 46(3):766–777, 2015. doi: <https://doi.org/10.1109/TCYB.2015.2415032>.
- [Jan et al.(2019)Jan, Farman, Khan, Imran, Islam, Ahmad, Ali, and Jeon] Bilal Jan, Haleem Farman, Murad Khan, Muhammad Imran, Ihtesham Ul Islam, Awais Ahmad, Shaukat Ali, and Gwanggil Jeon. Deep learning in big data analytics: a comparative study. *Computers & Electrical Engineering*, 75:275–287, 2019.
- [Mehedi et al.(2021)Mehedi, Anwar, Rahman, and Ahmed] Sk Tanzir Mehedi, Adnan Anwar, Ziaur Rahman, and Kawsar Ahmed. Deep transfer learning based intrusion detection system for electric vehicular networks. *Sensors*, 21(14):4736, 2021. doi: 10.3390/s21144736.
- [Sharafaldin et al.(2018)Sharafaldin, Lashkari, and Ghorbani] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.
- [Li et al.(2020)Li, Fan, Tse, and Lin] Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Computers & Industrial Engineering*, 149:106854, 2020. doi: <https://doi.org/10.1016/j.cie.2020.106854>.
- [Agrawal et al.(2022b)Agrawal, Sarkar, Aouedi, Yenduri, Piamrat, Alazab, Bhattacharya, Maddikunta, and Gadekallu] Shaashwat Agrawal, Sagnik Sarkar, Ons Aouedi, Gokul Yenduri, Kandaraj Piamrat, Mamoun Alazab, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, and Thippa Reddy Gadekallu. Federated learning for intrusion detection system: Concepts, challenges and future directions. *Computer Communications*, 195:346–361, 2022b. doi: <https://doi.org/10.1016/j.comcom.2022.09.012>.
- [Alsamiri and Alsubhi(2023)] Jadir Alsamiri and Khalid Alsubhi. Federated learning for intrusion detection systems in internet of vehicles: A general taxonomy, applications, and future directions. *Future Internet*, 15(12):403, 2023. doi: 10.3390/fi15120403.
- [X.1375 Working Group(2020)] X.1375 Working Group. Guidelines for an intrusion detection system for in-vehicle networks. Technical Report X.1375, International Telecommunication Union, 2020.

- [Althunayyan et al.(2024b)Althunayyan, Javed, Rana, and Spyridopoulos] Muzun Althunayyan, Amir Javed, Omer Rana, and Theodoros Spyridopoulos. Hierarchical federated learning-based intrusion detection for in-vehicle networks. *Future Internet*, 16(12):451, 2024b.
- [Driss et al.(2022)Driss, Almomani, e Huma, and Ahmad] Maha Driss, Iman Almomani, Zil e Huma, and Jawad Ahmad. A federated learning framework for cyberattack detection in vehicular sensor networks. *Complex & Intelligent Systems*, 8(5):4221–4235, 2022. doi: 10.1007/s40747-022-00705-w.
- [Shibly et al.(2022)Shibly, Hossain, Inoue, Taenaka, and Kadobayashi] Kabid Hassan Shibly, Md Delwar Hossain, Hiroyuki Inoue, Yuzo Taenaka, and Youki Kadobayashi. Personalized federated learning for automotive intrusion detection systems. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 544–549. IEEE, 2022. doi: <https://doi.org/10.1109/FNWF55208.2022.00101>.
- [Yu et al.(2022)Yu, Hua, Wang, Yang, and Hu] Tianqi Yu, Guodong Hua, Huaisheng Wang, Jianfeng Yang, and Jianling Hu. Federated- lstm based network intrusion detection method for intelligent connected vehicles. *IEEE International Conference on Communications (ICC)*, 2022. doi: <https://doi.org/10.1109/ICC45855.2022.9838655>.
- [Zhang et al.(2023)Zhang, Zeng, and Lin] Hengrun Zhang, Kai Zeng, and Shuai Lin. Federated graph neural network for fast anomaly detection in controller area networks. *IEEE Transactions on Information Forensics and Security*, 18:1566–1579, 2023. doi: <https://doi.org/10.1109/TIFS.2023.3240291>.
- [Yang et al.(2022b)Yang, Hu, and Yu] Jianfeng Yang, Jianling Hu, and Tianqi Yu. Federated ai-enabled in-vehicle network intrusion detection for internet of vehicles. *Electronics*, 2022b. doi: <https://doi.org/electronics11223658>.
- [Taslimasa et al.(2023b)Taslimasa, Dadkhah, Neto, Xiong, Iqbal, Ray, and Ghorbani] Hamideh Taslimasa, S. Dadkhah, E. P. Neto, Pulei Xiong, Shahrear Iqbal, S. Ray, and A. Ghorbani. Imagefed: Practical privacy preserving intrusion detection system for in-vehicle can bus protocol. *IEEE BigDataSecurity*, 2023b. doi: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00031.
- [Hoang et al.(2023)Hoang, Islam, Yim, and Kim] Thien-Nu Hoang, Md Rezanur Islam, Kangbin Yim, and Daehee Kim. Canperfl: improve in-vehicle intrusion detection performance by sharing knowledge. *Applied Sciences*, 13(11):6369, 2023.
- [Marchetti and Stabili(2018)] Mirco Marchetti and Dario Stabili. Read: Reverse engineering of automotive data frames. *IEEE Transactions on Information Forensics and Security*, 14(4):1083–1097, 2018. doi: 10.1109/TIFS.2018.2870826.
- [McMahan et al.(2017)McMahan, Moore, Ramage, Hampson, and y Arcas] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017. doi: 10.48550/arXiv.1602.05629.
- [Li et al.(2022)Li, Diao, Chen, and He] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022. doi: 10.48550/arXiv.2102.02079.
- [Zhao et al.(2018)Zhao, Li, Lai, Suda, Civin, and Chandra] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018. doi: 10.48550/arXiv.1806.00582.
- [Li et al.(2021)Li, Lin, and Xiong] Yi Li, Jing Lin, and Kaiqi Xiong. An adversarial attack defending system for securing in-vehicle networks. In *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2021.
- [Axelsson and Sands(2006)] Stefan Axelsson and David Sands. *Understanding intrusion detection through visualization*, volume 24. Springer Science & Business Media, 2006.
- [Lundberg et al.(2022)Lundberg, Mowla, Abedin, Thar, Mahmood, Gidlund, and Raza] Hampus Lundberg, Nishat I Mowla, Sarder Fakhrul Abedin, Kyi Thar, Aamir Mahmood, Mikael Gidlund, and Shahid Raza. Experimental analysis of trustworthy in-vehicle intrusion detection system using explainable artificial intelligence (xai). *IEEE Access*, 10:102831–102841, 2022.

- [Abualhoul et al.(2016)Abualhoul, Shagdar, and Nashashibi] Mohammad Y Abualhoul, Oyunchimeg Shagdar, and Fawzi Nashashibi. Visible light inter-vehicle communication for platooning of autonomous vehicles. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 508–513. IEEE, 2016. doi: 10.1109/IVS.2016.7535434.
- [Moubayed et al.(2020)Moubayed, Shami, Heidari, Larabi, and Brunner] Abdallah Moubayed, Abdallah Shami, Parisa Heidari, Adel Larabi, and Richard Brunner. Edge-enabled v2x service placement for intelligent transportation systems. *IEEE Transactions on Mobile Computing*, 20(4):1380–1392, 2020.