Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.6140

Learning-Based Intrusion Detection Systems for In-Vehicle CAN Networks: A Comprehensive Survey with Deployment and Real-Time Considerations

Athar Ghadi [**Eastern Michigan University** | ***aghadi@emich.edu***]
Tauheed Khan Mohd [**Eastern Michigan University** | ***tkhanmoh@emich.edu***]

School of Information Security and Applied Computing, Eastern Michigan University, Ypsilanti, Michigan 48197, USA.

Received: 21 June 2025 • **Accepted:** 05 April 2026 • **Published:** 02 July 2026

Abstract The Controller Area Network (CAN) bus continues to serve as the core communication backbone of modern vehicles. However, its original design did not incorporate fundamental security mechanisms, leaving in-vehicle networks vulnerable to cyberattacks such as spoofing, replay, message injection, and denial-of-service. As a result, Intrusion Detection Systems (IDSs) have become an essential component of automotive cybersecurity, providing continuous monitoring of CAN traffic to identify malicious behavior. In recent years, researchers have increasingly turned to intelligent IDS solutions based on Machine Learning (ML), Deep Learning (DL), and hybrid learning approaches to enhance detection capability. Although many of these studies report impressive detection accuracy, their evaluation practices often vary significantly, and claims related to real-time performance or lightweight deployment are frequently made without sufficient practical validation. This survey provides a structured, deployment-focused review of learning-based IDSs for CAN bus security published between 2019 and 2025. Building on prior surveys that emphasize detection accuracy and high-level method categorization, this work evaluates IDS approaches from a practical perspective by considering detection performance alongside computational efficiency, real-time feasibility, and deployment readiness in resource-constrained automotive environments. ML-based, DL-based, and hybrid IDS approaches are organized within a unified taxonomy and systematically compared across model architectures, attack scenarios, datasets, real-time feasibility, lightweight design claims, and validation strategies. A key contribution of this survey is the introduction of explicit and consistent criteria for labeling IDSs as lightweight, real-time, or deployable, based only on substantiated evidence such as ECU-oriented runtime analysis, embedded evaluation, or real-vehicle experimentation. Through a set of unified comparative tables, the survey highlights common evaluation gaps, and mismatches between reported performance and practical feasibility.

Keywords: Controller Area Network, Intrusion Detection System, Machine Learning, Deep Learning, Automotive Cybersecurity, Cyberattacks, CAN Bus Security, Hybrid, Vehicle Networks.

1 Introduction

The automotive industry is undergoing a rapid transformation, with vehicles evolving from mechanical systems into sophisticated cyber-physical platforms integrating connectivity, automation, and software-defined functionalities. This transition, while improving safety, comfort, and efficiency, has significantly expanded the attack surface of in-vehicle networks. At the heart of vehicle communication is the Controller Area Network (CAN) bus, a lightweight and real-time protocol that enables efficient message exchange among Electronic Control Units (ECUs) responsible for critical functions such as braking, steering, engine control, and infotainment [Al-Jarrah *et al*., 2019; Lotto *et al*., 2024a; Bozdal *et al*., 2018]. Despite its widespread adoption, the CAN bus was designed without inherent security mechanisms such as encryption, authentication, or access control [Koscher *et al*., 2010]. It is highly vulnerable to cyberattacks, including spoofing, replay, message injection, denial-of-service (DoS), and fuzzing, that can affect vehicle safety and user privacy [Kang *et al*., 2018; Palanca *et al*., 2017; Li *et al*., 2021]. As vehicles increasingly integrate remote connectivity via Bluetooth, cellular, and Vehicle-to-Everything (V2X) interfaces, attackers can exploit not only physical but also remote attack vectors, increasing the risk of cyber intrusions [Talebi, 2015].

Conventional security measures such as cryptography and firewalls are often unsufficient alone in CAN networks due to their tight constraints on latency, bandwidth, and computational resources [Bozdal *et al*., 2018]. Intrusion Detection Systems (IDS) have emerged as a key component of automotive cybersecurity, providing mechanisms to monitor in-vehicle network traffic in order to detect anomalies and mitigate attacks in real time [Almehdhar *et al*., 2024; Khraisat *et al*., 2019].

In recent years, researchers have increasingly adopted artificial intelligence (AI) methods to improve intrusion detection in automotive systems. These include Machine Learning (ML), Deep Learning (DL), and hybrid approaches, each offering specialized strengths for IDS development [Tanksale, 2019; Hossain *et al*., 2020b; Chougule *et al*., 2024]. Machine Learning-based IDS uses a variety of algorithms, such as Support Vector Machines (SVM) [Cortes and Vapnik, 1995], Decision Trees (DT) [Quinlan, 1986], Random Forests (RF) [Breiman, 2001], K-Nearest Neighbors (KNN) [Cover and

Hart, 1967], Extreme Gradient Boosting (XGBoost) [Chen *et al*., 2015], and Naïve Bayes (NB) [Duda *et al*., 2006], which offer inherent model transparency with low computational overhead, making them suitable for deployment on resource-limited ECUs. However, their dependence on manually engineered features limits their ability to generalize to unseen attacks [Alfardus and Rawat, 2021] [Gundu and Maleki, 2022]. Deep Learning-based IDS, such as Recurrent Neural Networks (RNNs) [Elman, 1990], Long Short-Term Memory (LSTM) networks [Hochreiter and Schmidhuber, 1997], Convolutional Neural Networks (CNNs) [LeCun *et al*., 2002], and Transformer-based models [Vaswani *et al*., 2017], overcome these limitations by automatically extracting feature and sequence-based patterns from raw CAN traffic, achieving state-of-the-art accuracy in detecting complex attack vectors [Samir *et al*., 2024; Alkhatib *et al*., 2022; Zhou *et al*., 2024]. Hybrid IDS approaches, which combine the complementary strengths of ML and DL, further enhance detection performance and adaptability, addressing the diverse and evolving threat environment in vehicular networks [Chougule *et al*., 2024; Althunayyan *et al*., 2024; Nichelini *et al*., 2023]. Several recent surveys have explored IDS in broader cybersecurity domains, including network-based IDS, industrial control systems, IoT environments, and general AI-driven detection paradigms. These works investigate advanced learning strategies such as Transformer-based models Kheddar *et al*. [2024], large language models Kheddar [2025], and deep transfer learning Kheddar *et al*. [2023], offering valuable insights into IDS design trends and evaluation methodologies. However, their analyses remain largely domain-agnostic and do not address the protocol-specific characteristics, real-time constraints, and deployment challenges unique to in-vehicle CAN bus networks. Consequently, a dedicated and CAN-focused survey remains necessary. Table 1 lists the acronyms used in this survey.

To systematically examine the current state of research in learning-based intrusion detection for in-vehicle CAN networks, this survey is guided by a set of research questions aimed at structuring the analysis of existing studies and identifying key research gaps:

1. What categories of learning-based intrusion detection approaches, including ML, DL, and hybrid methods, have been proposed for in-vehicle CAN networks?

2. What datasets, attack scenarios, and evaluation practices are most commonly used to assess CAN bus IDSs?

3. How consistently do existing studies substantiate claims related to lightweight design, real-time capability, and deployment feasibility in resource-constrained automotive environments?

4. What key limitations, unresolved challenges, and future research directions can be identified in the current body of literature on learning-based CAN IDS?

2 Review Methodology

This section outlines the methodology used to conduct the review. It describes the literature search strategy, the criteria applied for study selection, and the data extraction process used to organize and analyze the studies included in this survey.

2.1 Search Strategy

A systematic literature search was performed using Google Scholar as the primary discovery tool to identify and retrieve relevant studies from major scholarly databases, including IEEE Xplore, ACM Digital Library, SpringerLink, and ScienceDirect. This approach ensured comprehensive coverage across the domains of automotive cybersecurity, embedded systems, and machine learning, leveraging Google Scholar's broad indexing to capture interdisciplinary research.

The search primarily focused on studies published between 2019 and 2025 in order to identify recent and relevant research on learning-based IDSs for CAN-based in-vehicle networks. Earlier works were included where appropriate to provide foundational background on CAN security, early IDS concepts, and classical machine learning techniques referenced by later studies. The review primarily prioritized peer-reviewed journal and conference publications. However, a limited number of recent preprints, technical reports, and official industry resources were also considered when they provided important contextual or emerging information relevant to the survey.

2.2 Search Terms

The literature search used combinations of keywords related to CAN networks, intrusion detection, and learning-based approaches. Representative search strings included combinations such as:

- ("CAN bus" OR "Controller Area Network") AND ("intrusion detection" OR "IDS" OR "anomaly detection")
- ("CAN bus" OR "in-vehicle network") AND ("machine learning" OR "deep learning")
- ("automotive cybersecurity") AND ("CAN intrusion detection")
- ("vehicle CAN network") AND ("hybrid intrusion detection" OR "deep learning IDS")
- ("lightweight IDS" OR "lightweight intrusion detection") AND ("CAN bus" OR "in-vehicle network")
- ("real-time detection" OR "real-time IDS") AND ("automotive cybersecurity" OR "CAN network")
- ("embedded systems" OR "embedded deployment") AND ("intrusion detection" OR "IDS")
- ("automotive cybersecurity") AND ("lightweight IDS" OR "real-time intrusion detection")

2.3 Inclusion and Exclusion Criteria

To maintain consistency and relevance, explicit inclusion and exclusion criteria were applied during the study selection process.

Studies were included if they met the following criteria:

- The review primarily focused on peer-reviewed journal articles and conference papers. A limited number of preprints, technical reports, and official industry resources were also consulted when they offered additional recent or contextual information relevant to the survey.

Table 1. List of acronyms used throughout this paper.

Acronym	Definition	Acronym	Definition
ACK	Acknowledgment	KNN	K-Nearest Neighbors
AD	Attack and Defense	LDA	Linear Discriminant Analysis
AI	Artificial Intelligence	LLC	Logical Link Control
AMP	Arbitration on Message Priority	LR	Logistic Regression
ANFIS	Adaptive Neuro-Fuzzy Inference System	LSTM	Long Short-Term Memory
ANN	Artificial Neural Network	MAC	Medium Access Control
AUC	Area Under Curve	MBR	Modified Bat Algorithm
Bi	Bidirectional	MCFO	Metaheuristic Central Force Optimization
BNN	Binarised Neural Network	ML	Machine Learning
C2F	Coarse-to-Fine	MLP	Multilayer Perceptron
CAN	Controller Area Network	NB	Naive Bayes
CAN FD	Controller Area Network with Flexible Data Rate	NN	Neural Network
CAN XL	Controller Area Network Extra Long	NRC	National Research Council (Canada)
CH	Car Hacking	NRZ	Non-Return to Zero
CNN	Convolutional Neural Network	OCSVM	One-Class Support Vector Machine
CRC	Cyclic Redundancy Check	OTA	Over-the-Air
DANN	Dynamic Artificial Neural Network	RL	Reinforcement Learning
DIDS	Distributed Intrusion Detection System	RNN	Recurrent Neural Network
DL	Deep Learning	ROAD	ORNL Automotive Dynamometer
DNN	Deep Neural Network	RPM	Revolutions Per Minute
DoS	Denial of Service	RTR	Remote Transmission Request
DT	Decision Tree	SA	Survival Analysis
DTL	Deep Transfer Learning	SC	Subtractive Clustering
ECU	Electronic Control Unit	SOF	Start of Frame
EOF	End of Frame	SRR	Substitute Remote Request
ET	Extra Trees	SVC	Support Vector Classification
FCM	Fuzzy C-Means (Clustering)	SVM	Support Vector Machine
FFNN	Feed-Forward Neural Network	SVDD	Support Vector Data Description
FL	Federated Learning	TF-IDF	Term Frequency – Inverse Document Frequency
FNR	False Negative Rate	TPMS	Tire Pressure Monitoring System
FPR	False Positive Rate	UAVCAN	Uncomplicated Application-level Vehicular Communication and Networking
GAN	Generative Adversarial Network	V2I	Vehicle-to-Infrastructure
GCN	Graph Convolutional Network	V2X	Vehicle-to-Everything
GMM	Gaussian Mixture Model	VGG	Visual Geometry Group
GNN	Graph Neural Network	XGBoost	Extreme Gradient Boosting
GRU	Gated Recurrent Unit		
HSNRFM	Hybrid Similar Neighborhood Robust Factorization Machine		
IDS	Intrusion Detection System		
IDE	Identifier Extension		
iForest	Isolation Forest		
IoT	Internet of Things		
IVN	In-Vehicle Network		

- Research focused on intrusion detection in CAN-based in-vehicle networks.
- Studies proposing or evaluating ML, DL, or hybrid IDS approaches.
- Papers providing experimental evaluation, datasets, or quantitative performance results.
- Publications written in English, with the main review emphasis placed on studies published between 2019 and 2025.
- Earlier foundational studies were included where appropriate to support the technical background of the reviewed IDS approaches.

Studies were excluded if they met any of the following conditions:

- Papers addressing general automotive cybersecurity without focusing on CAN intrusion detection.
- Studies lacking sufficient technical detail or experimental evidence for meaningful comparison.
- Works focused exclusively on cryptographic protection or authentication mechanisms without clear relevance to IDS design, evaluation, or comparative discussion, except when cited for background context.

2.4 Article Selection and Data Extraction

Articles were selected using a staged screening approach. First, titles and abstracts were reviewed to identify studies relevant to CAN bus intrusion detection. Papers that passed this initial screening were subsequently assessed through full-text review to confirm their relevance for inclusion in the survey.

For each selected article, several attributes were extracted to enable structured comparison across the literature. These attributes included the learning paradigm (ML, DL, or hybrid), model architecture, attack scenarios, dataset characteristics, evaluation metrics, reported detection performance, latency or runtime evidence, lightweight design claims, and, where reported, evidence related to real-time operation, lightweight implementation, or deployment feasibility. This information forms the basis of the comparative analysis and taxonomy presented in the subsequent sections of this survey.

While the search strategy aimed to capture the most relevant literature, the goal of this survey is not to exhaustively include every published study, but rather to identify representative and relevant works that reflect current research trends in learning-based CAN IDSs.

3 Related Surveys

Several surveys have examined the cybersecurity challenges associated with CAN systems in modern vehicles.

Al-Jarrah et al. categorized IDSs into flow-based, payload-based, and hybrid approaches, and further grouped them by detection technique, including rule-based methods, statistical and entropy-based models, and early machine-learning approaches. While several learning-based techniques were discussed, such as early neural networks, Hidden Markov Models(HMM), SVMs, and sequence-modeling approaches, the coverage of ML and DL methods remained limited and non-systematic. [Al-Jarrah *et al*., 2019]. Authors in [Young *et al*., 2019] provided an early and structured overview of IDSs for automotive CAN security, with the primary objective of unifying threat models, assumptions, and terminology used in automotive IDS research, however, the survey offered only limited coverage of learning-based approaches. In [Dupont *et al*., 2019a] the authors focused primarily on organizing existing approaches rather than evaluating detection performance, and introduced a CAN-specific taxonomy based on the number of frames analyzed, the type of data exploited (timing, CAN ID, and payload), and the method used to construct the detection model (specified versus learned). While several learning-based techniques, including early neural-network and sequence-modeling approaches, are briefly discussed, coverage of ML and DL methods remains limited and non-systematic. The study in [Lokman *et al*., 2019] focused on unifying CAN-related threat models, attack surfaces, IDS architectural placement within the IVN, and detection paradigms, and introduced a CAN-specific taxonomy based on four dimensions: detection approach, IDS placement location, attacking technique, and technical challenges. The primary objective of the survey was to organize existing research rather than to evaluate detection performance. The use of ML and DL approaches for IVN intrusion detection was acknowledged in [Wu *et al*., 2019], and ML-based IDSs, such as SVMs, HMMs, and DTs, along with early DL models including DNNs and LSTM-based predictors, were primarily introduced as illustrative examples of functional-level detection, but a structured comparison of model architectures was not provided. Authors in [Aliwa *et al*., 2021] discussed anomaly-based IDS approaches that include statistical methods and ML-based techniques for detecting abnormal CAN traffic patterns. However, ML and DL approaches are discussed at a high level, without a systematic breakdown of specific ML or DL models.

Karopoulos et al. presented a comprehensive survey-of-surveys on in-vehicle IDSs, proposing a unified meta-taxonomy while systematically reviewing CAN-centric ML, DL, and hybrid IDS approaches. The authors integrated diverse classification schemes from prior IVN-IDS surveys into a unified taxonomy based on deployment location, detection type, protocol layering, and reaction type, which enabled more consistent comparisons across studies. Within this framework, ML and DL approaches were categorized as subclasses of anomaly-based IDS, alongside statistical and physical-characteristics-based methods. However, the survey did not compare individual ML and DL models in terms of detection accuracy, computational complexity, or real-time feasibility [Karopoulos *et al*., 2022]. Fakhfakh et al. presented a systematic literature review of cybersecurity attacks targeting CAN bus based vehicles, with a strong emphasis on developing a comprehensive taxonomy of both local and remote attack scenarios and analyzing the validation strategies used to assess proposed security solutions. The survey examined validation approaches such as simulation, experimentation, and formal verification; however, these analyses focused on correctness and evaluation methodology rather than on practical deployment feasibility. While the authors acknowledged the role of IDS, learning based IDS approaches, including ML and DL methods, were discussed only at a high level and primarily as representative categories within the broader IDS landscape. The survey did not provide a structured or comparative analysis of ML or DL model architectures, feature representations, datasets, detection performance metrics, or real-time deployability on resource constrained ECUs [Fakhfakh *et al*., 2022]. The work in [Rajapaksha *et al*., 2023] systematically organized learning-based intrusion detection techniques, including traditional ML, DL, sequence learning, and hybrid models, and provided a strong taxonomical structure with broad coverage of ML and DL approaches and benchmark datasets. However, it did not present a structured evaluation of deployment feasibility or real time operational constraints. Another contribution presented in [Nagarajan *et al*., 2023] is a detailed taxonomy of

ML and DL techniques, ranging from traditional ML methods (e.g., OCSVM, HMMs, and DT) to DL architectures such as CNNs, RNNs (LSTM/GRU), autoencoders, attention-based models, and hybrid CNN-RNN frameworks; however, the emphasis is placed on algorithmic categorization rather than on practical deployment feasibility, lightweight operation, or real time constraints. [Sharmin *et al*., 2024] systematically analyzed existing CAN IDS benchmarking efforts along five dimensions: IDS type, attack model, evaluation type (offline versus online), workload generation, and evaluation metrics. Although ML and DL–based IDSs were included within the reviewed benchmarking and comparative studies, the paper did not provide a structured architectural comparison or a performance-level analysis.

The work in [Almehdhar *et al*., 2024] presented a well-structured taxonomy that categorized IDS techniques into non-ML, traditional ML, DL, and hybrid approaches. It demonstrated strong coverage of model architectures, including CNNs, RNNs, autoencoders, transformers, GANs, and DRL-based IDSs, and discussed emerging paradigms such as federated learning and transfer learning. However, deployment-oriented aspects and real-time feasibility were addressed primarily at a descriptive level. The survey did not provide a systematic comparison of these aspects across the reviewed studies. Although lightweight operation and practical deployability were acknowledged as important challenges, these aspects were not consistently evaluated or validated across the surveyed works.

The authors in [Lotto *et al*., 2024a] presented a thorough and well-structured analysis of cryptographic authentication protocols for the CAN, examining the security properties and operational considerations of 15 widely studied authentication schemes. Rather than broadly surveying CAN vulnerabilities, IDSs, or generic countermeasures, as was common in earlier works, the authors focused on authentication mechanisms. By doing so, they introduced a unified comparison framework built on clearly defined security requirements and practical operational criteria, enabling a more focused and technically grounded evaluation of CAN authentication protocols.

Authors in [Althunayyan *et al*., 2025] presented a comprehensive survey of learning-based IDSs for IVNs, systematically reviewing ML, DL, and federated learning approaches for CAN-bus security. The work employed a structured search methodology and categorized IDSs based on the types of attacks they detect, including known, unknown, and combined threats, while also reviewing commonly used evaluation metrics. Although the authors emphasized the importance of time and memory considerations for automotive environments, the survey did not provide a structured comparison of deployment validity, real-time operation, or lightweight validation across the reviewed studies.

Building on the organizational contributions of prior surveys, existing literature has systematically structured intrusion detection research for CAN-bus–based systems by classifying approaches into ML, DL, and hybrid categories. These surveys have been instrumental in consolidating fragmented research, clarifying detection paradigms, and establishing taxonomies that contrast learning strategies, feature representations, and attack models. However, a recurring limitation explicitly acknowledged in many of these surveys often highlighted as a research gap or future direction is the lack of consistent evaluation of lightweight operation, real time feasibility, and practical deployment suitability under in vehicle constraints.

Despite the growing number of surveys on CAN bus security, most existing reviews primarily emphasize detection accuracy and high-level method categorization, while the evaluation rigor of intrusion detection approaches is treated inconsistently. In particular, aspects such as whether lightweight claims are explicitly supported, real-time feasibility is clearly demonstrated, and validation extends beyond offline performance metrics are often insufficiently examined. To address this gap, this survey provides a focused and literature-driven review of IDS research published between 2019 and 2025, systematically analyzing ML, DL, and hybrid approaches for CAN bus security. Beyond summarizing attack scenarios, datasets, and reported detection performance, the survey explicitly documents whether each study claims or demonstrates lightweight operation under computationally constrained in-vehicle environments, and it critically examines deployment considerations and real-time feasibility, which are systematically reported in the comparative analysis presented in Section 6.

Table 2 provides a detailed comparison of prior related surveys and the present survey with respect to key aspects, including coverage of ML, DL, and hybrid approaches, the use of datasets, lightweight and ECU-oriented design considerations, real-time feasibility, and deployment validation.

4 Contribution of This Survey

This survey provides a focused and systematic review of IDSs for in-vehicle CAN security, addressing several limitations observed in prior surveys. While prior studies have made substantial contributions in organizing IDS taxonomies, characterizing attack surfaces, analyzing datasets, and reporting detection performance, this work places explicit emphasis on lightweight operation, real-time feasibility, and practical deployability under in-vehicle constraints.

The main contributions of this survey are summarized as follows:

- Deployment-Oriented Review Framework: This survey adopts explicit and consistent evaluation criteria to examine whether reported IDS approaches demonstrate or substantiate claims of real-time operation, lightweight design, and feasibility on resource-constrained ECUs, rather than relying solely on offline performance metrics.
- Structured Taxonomy of Intelligent IDS Approaches: A clear and unified taxonomy is presented to categorize intelligent CAN bus IDSs into ML, DL, and hybrid (ML + DL) approaches, enabling consistent comparison across learning paradigms, detection strategies, and operational assumptions.
- Comprehensive Comparative Analysis Across Multiple Dimensions: Beyond detection performance, the survey systematically documents and compares IDS studies with respect to attack scenarios, dataset characteristics, inference or execution time, lightweight claims, validation methodology, and deployment context. This

Table 2. Comparison of existing surveys on IDSs for CAN bus networks.

Survey	ML	DL	Hybrid	Datasets	Lightweight / ECU Focus	Real-Time Feasibility	Deployment Validation
[Al-Jarrah *et al*., 2019]	✓	✓	✓	✓	×	×	×
[Young *et al*., 2019]	Partial	Partial	×	×	Partial	×	×
[Dupont *et al*., 2019a]	Partial	Partial	Partial	×	×	×	×
[Lokman *et al*., 2019]	✓	✓	✓	×	×	×	×
[Wu *et al*., 2019]	✓	✓	×	✓	×	Partial	×
[Aliwa *et al*., 2021]	✓	✓	✓	✓	Partial	Partial	Partial
[Karopoulos *et al*., 2022]	✓	✓	✓	✓	×	×	×
[Fakhfakh *et al*., 2022]	✓	✓	✓	×	×	×	✓
[Rajapaksha *et al*., 2023]	✓	✓	✓	✓	×	×	×
[Nagarajan *et al*., 2023]	✓	✓	Partial	✓	×	×	×
[Sharmin *et al*., 2024]	✓	✓	✓	✓	Partial	×	×
[Almehdhar *et al*., 2024]	✓	✓	✓	✓	×	Partial	Partial
[Lotto *et al*., 2024a]	×	×	×	×	×	×	×
[Althunayyan *et al*., 2025]	✓	✓	✓	✓	Partial	Partial	Partial
This Survey	✓	✓	✓	✓	✓	✓	✓

Note: Partial *denotes high-level or conceptual coverage with limited systematic methodology, experimental evaluation, or deployment validation.*

multi-dimensional comparison provides a more realistic assessment of IDS maturity and readiness.

- Overview of CAN IDS: This survey discusses publicly available and restricted CAN intrusion detection datasets in terms of vehicle realism, attack coverage, labeling strategies, and data characteristics.
- Explicit Criteria for Real-Time and Lightweight Classification: Clear and consistent criteria are defined for labeling IDS approaches as real-time, deployable, or lightweight. Studies are marked accordingly only when supported by explicit evidence such as ECU-oriented runtime evaluation, real-vehicle experiments, or embedded feasibility discussions.
- Discussion of Research Gaps and Practical Constraints This survey discusses open challenges related to model generalization, evaluation realism, dataset bias, and deployment validation. The discussion highlights practical constraints that limit real-world adoption and outlines research directions that emphasize scalable, lightweight, and operationally viable IDS solutions for automotive environments.

4.1 Organization of this survey

The remainder of this survey is organized as follows: Section 5 provides background on the CAN protocol, including its architecture, communication mechanisms, inherent security vulnerabilities, common attack types, and an overview of publicly available and proprietary CAN IDS datasets. Section 6 presents a structured taxonomy and systematic review of recent intelligent IDS approaches for CAN bus security, categorized into machine learning based, deep learning based, and hybrid (ML + DL) methods. Within each category, IDS approaches are systematically analyzed with respect to detection performance, computational efficiency, real-time and deployment feasibility, lightweight design claims, datasets, and validation strategies. Section 7 discusses and synthesizes key observations drawn from the cross-study review, emphasizing evaluation practices, dataset influence, and trade-offs between detection capability and deployability. Section 8 examines deployment challenges and practical limitations that limit the transition of IDS solutions from experimental studies to real-vehicle environments. Section 9 identifies open research challenges and future directions, including benchmarking, robustness, interpretability, and scalable deployment strategies. Finally, Section 10 concludes the survey by summarizing the main findings and outlining the implications for advancing practical and reliable CAN bus IDSs.

5 Background

In-vehicle communication networks-particularly the CAN-serve as the backbone of modern automotive electronics, facilitating real-time data exchange among various ECUs. The widespread adoption of CAN can be attributed to its simplicity, fault tolerance, and cost-effectiveness [Bozdal *et al*., 2018]. However, the protocol lacks built-in security features such as message authentication and encryption, making it inherently vulnerable to attacks including spoofing, message injection, replay, and DoS. These vulnerabilities are further increased by the integration of remote connectivity technologies such as Bluetooth, cellular networks,Vehicle to Infrastructure (V2I), and V2X communications, which greatly increases the risk of attacks [Bozdal *et al*., 2018]. Due to the CAN bus's real-time and low-latency design requirements, conventional security mechanisms like encryption and firewalls are often unsuitable [Avatefipour and Malik, 2018]. Various intelligent and traditional techniques have been developed to provide efficient and accurate intrusion detection within CAN networks. This background section begins with an overview of the CAN bus, followed by a discussion of its security vulnerabilities

and IDS background.

5.1 Controller Area Network

The CAN is a communication system developed by Bosch in the early 1980s, used in vehicles to enable efficient data exchange between various nodes, commonly known as ECUs. Each ECU either senses and reports specific information or receives data from other ECUs, processes it, and shares the results across the network. Examples of nodes connected to a vehicle's CAN network include the engine control unit, airbag system, audio system, and tire pressure monitoring system (TPMS).

The CAN network and its communication protocol are relatively simple yet highly reliable, enabling seamless communication among all connected nodes. Its architecture prioritizes robustness, simplicity, and deterministic behavior which are key requirements for safety critical automotive systems. Each CAN node can receive messages on the network and determine whether the message is intended for it by interpreting specific encoded information within the message. This information defines the message type and other critical aspects, such as its priority, which is managed through a process known as arbitration. The CAN network primarily consists of two main wires: CAN High (CAN H) and CAN Low (CAN L). Each node is connected to the network and has the ability to both send and receive messages. The CAN protocol employs a multi-master, broadcast architecture, allowing any ECU to transmit a message when the bus is idle. Once a message is sent to the bus, all connected nodes can receive, parse, and process it if the message is intended for them. Figure 1 part (a) depicts the CAN bus with some nodes connected. Physical information of the CAN bus is governed by the physical layer, while data frame and error detection is governed by the Data link layer [Al-Jarrah *et al*., 2019].

Physical Layer

The CAN bus physical layer defines various parameters, such as cable types, electrical signal levels, node requirements, cable impedance, baud rate, cable length, and termination. For example, the classical CAN bus operates at a baud rate of 1 Mbit/s, while the flexible data rate version operates at 5 Mbit/s. Similarly, the CAN bus must be properly terminated with a resistance, typically 120 ohms. Signals on the CAN bus can be either 0 or 1, where 1 refers to a recessive state and 0 refers to a dominant state. The CAN bus utilizes the Non-Return to Zero (NRZ) encoding technique. Due to imperfections in signal transitions, stabilization time is required during bit transmission. Each bit is divided into 8 to 25 time segments, with sampling typically occurring at two-thirds of the bit duration. A synchronization process ensures that each ECU aligns its sampling phase with the bitstream edges. Receiving nodes adjust their sample points to match the transmitter's timing, accounting for delays introduced by physical distance and signal propagation.[Lotto *et al*., 2024b].

Data Link Layer

The data link layer services are implemented in the Logical Link Control (LLC) and Medium Access Control (MAC) sublayers of a CAN controller. The LLC provides acceptance filtering, overload notification and recovery management. The MAC is responsible for data encapsulation (de-capsulation), frame coding (stuffing/de-stuffing), medium access management, error detection, error signaling, acknowledgement, and serialization (de-serialization) [Specification, 1991] [Lotto *et al*., 2024b].

CAN frames include several built-in error detection mechanisms, such as Cyclic Redundancy Check (CRC), bit stuffing, and acknowledgment (ACK) slots, all of which help maintain message integrity in noisy environments [Specification, 1991].

Architecture of a CAN-Based ECU

Each ECU in a CAN-based vehicle network is typically composed of three essential components: a micro-controller, a CAN controller, and a CAN transceiver. The micro-controller executes the software logic, processes input from sensors, and controls actuators. The CAN controller manages communication on the network, handling tasks such as message framing, bit timing, error detection (e.g., CRC), and filtering based on message identifiers. This filtering is critical, as all messages on the CAN bus are broadcast to all ECUs-only those with matching acceptance filters process the message content [Bozdal *et al*., 2018][Specification, 1991]. The CAN transceiver serves as the interface between the controller and the physical bus. It converts logic-level signals to differential voltages on the two-wire bus (CAN-H and CAN-L), and vice versa. This differential signaling is key to the protocol's noise resilience and fault tolerance, particularly in the electromagnetically noisy environments of modern vehicles [Specification, 1991] [Presi, 2013]. This design facilitates a flexible and efficient communication network, allowing ECUs to exchange time-critical data across multiple vehicle subsystems. Figure 1 part (b) illustrates the internal structure of a typical CAN node, including its three core components: the Microcontroller, CAN Controller, and CAN Transceiver.

The CAN protocol has evolved through several versions, each addressing bandwidth and functionality needs while presenting new cybersecurity implications. CAN 2.0, the most widely used standard, supports up to 1 Mbps data rates and uses 11-bit or 29-bit identifiers but lacks native security features such as authentication or encryption, making it vulnerable to spoofing, injection, and replay attacks [Bozdal *et al*., 2018]. To address increasing data demands, CAN FD was introduced, enabling up to 64-byte payloads and transmission speeds up to 8 Mbps. However, its expanded frame length and flexible timing further expose the network to fuzzing and advanced injection attacks [Lotto *et al*., 2024a]. The latest iteration, CAN XL, offers bandwidths up to 20 Mbps and includes optional support for CANsec, which introduces authentication and encryption at the protocol level [Cena *et al*., 2025]. While CANsec represents a major step forward, most vehicles remain backward-compatible, inheriting vulnerabilities from earlier versions. As modern vehicles increasingly utilize mixed-version in-vehicle networks combining CAN 2.0, CAN FD, and CAN XL, IDSs must be robust enough to operate across varying frame formats, data rates, and security features [Althunayyan *et al*., 2024].

CAN Frame Structure

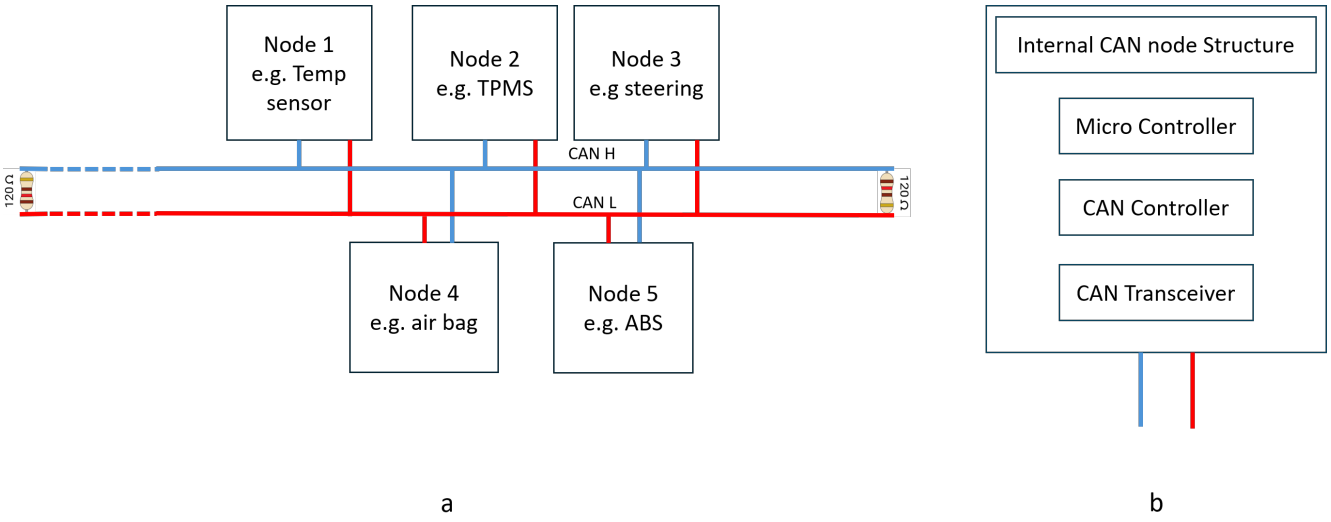

Figure 1. CAN bus Network. a) represents example of different nodes connected to the bus. b) represents a typical CAN node architecture

A CAN data frame consists of multiple fields and is available in two primary formats: the standard format, which uses an 11-bit identifier, and the extended format, which uses a 29-bit identifier [Al-Jarrah *et al*., 2019]. Figure 2 shows a typical CAN frame highlighting both formats. The key fields in each frame are described in Table 3.

The CAN utilizes a communication protocol known as Carrier Sense Multiple Access with Collision Detection and Arbitration on Message Priority (CSMA/CD+AMP) [Lotto *et al*., 2024b]. In this protocol, Carrier Sense Multiple Access (CSMA) ensures that each node monitors the bus and transmits only after detecting a period of inactivity. If two or more nodes attempt transmission simultaneously, Collision Detection and Arbitration on Message Priority (CD+AMP) is used to resolve conflicts.

Collisions are handled through a bit-wise arbitration mechanism based on the identifier field, which encodes the message priority. During arbitration, each transmitting node compares its own transmitted bits with those observed on the bus. Since CAN uses dominant (bit 0) and recessive (bit 1) logic, a node transmitting a recessive bit but detecting a dominant bit will stop transmitting. As a result, the message with the lowest numerical ID (i.e., highest priority) wins arbitration and continues transmission, while all others back off until the bus becomes idle again [Al-Jarrah *et al*., 2019].

This arbitration ensures that no data is lost, even when multiple nodes contend for the bus. Additionally, it guarantees that higher-priority messages are delivered deterministically, making CAN suitable for real-time applications. However, this mechanism imposes a limitation on bus bandwidth. According to protocol specifications, the signal propagation time between any two nodes must not exceed half the duration of one bit, effectively placing an upper bound on the data rate and the physical length of the CAN bus[Lotto *et al*., 2024b].

Security Challenges in CAN Networks

While the CAN protocol is efficient for real-time communication, it lacks foundational security measures. Moreover, As vehicles integrate more ECUs and external connectivity, critical vulnerabilities will increase [Bozdal *et al*., 2018; Koscher *et al*., 2010]. One major vulnerability arises from the broadcast-based transmission of CAN messages, which occurs without authentication. This permits attackers with physical or remote access to inject unauthorized messages, enabling spoofing and message injection attacks with minimal resistance. Furthermore, CAN does not support message confidentiality; any node connected to the network can monitor control and system-level traffic [Bozdal *et al*., 2018]. Message integrity is poorly enforced. While CRC can detect random transmission errors, it cannot prevent intentional modification or manipulation of message content by malicious attackers. [Almehdhar *et al*., 2024]. The arbitration-based bus access mechanism is also a point of exploitation. Attackers can initiate DoS attacks by transmitting high-priority messages in rapid succession, thereby blocking access for authorized ECUs and disrupting normal vehicle operations [Bozdal *et al*., 2018; Koscher *et al*., 2010].

The adoption of V2X communication, Over-the-Air (OTA) updates, and cloud-connected services has further intensified security concerns. Studies have demonstrated feasible attack vectors through telematics systems, mobile applications, and OTA firmware delivery mechanisms [Kang *et al*., 2018; Palanca *et al*., 2017; Talebi, 2015]. Several real-world incidents have highlighted the severity of vulnerabilities in CAN–based in-vehicle communication systems. In the widely publicized 2015 Jeep Cherokee hack by Miller and Valasek, researchers demonstrated a full remote-to-physical attack chain against a production vehicle. By exploiting vulnerabilities in the vehicle's Uconnect infotainment system, the attackers were able to remotely compromise the head unit over a wireless interface and subsequently pivot to the internal CAN networks. This access enabled the injection of malicious CAN messages targeting safety-critical electronic control units, allowing manipulation of vehicle functions in a moving vehicle. Importantly, the attack was performed on an unaltered, commercially available Jeep Cherokee, highlighting that the vulnerability was not the result of laboratory modifications but inherent architectural weaknesses in real-world automotive systems. The incident

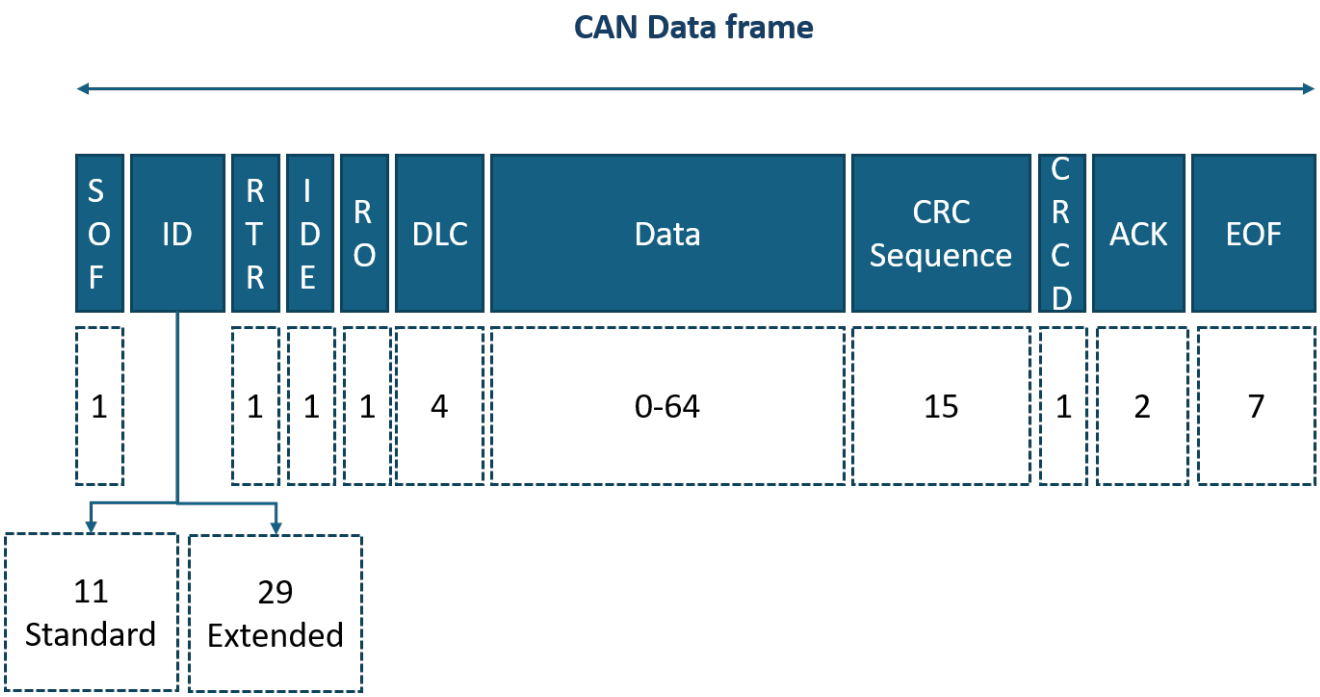

Figure 2. CAN Bus Typical Frame

has since been widely discussed in the automotive security community and is often cited as a motivating example in subsequent research, demonstrating the need for improved in-vehicle security mechanisms capable of detecting anomalous CAN traffic in deployed vehicles [Miller and Valasek, 2015]. Similarly, Ken Tindell and Ian Tabor documented how certain Toyota RAV4 models could be compromised by physically accessing the CAN bus through exposed wiring behind the vehicle's headlights. By tapping into this wiring, an attacker can directly connect to the CAN bus without triggering conventional intrusion mechanisms. The attack involves injecting carefully crafted CAN messages that spoof signals normally generated by the smart key receiver, effectively convincing the vehicle that a valid key is present. As a result, the vehicle's immobilizer system can be disabled, allowing the doors to be unlocked and the engine to be started. This attack highlights a critical architectural weakness in CAN-based in-vehicle networks, where the absence of message authentication and source verification enables unauthorized devices with physical access to impersonate legitimate ECUs and perform safety- and security-critical actions [Azzarà, 2023]. And also, Groza and Murvay introduced an intrusion detection approach based on Bloom filtering for in-vehicle networks. The filter is designed to avoid false negatives and achieves a reported recall of 100%. Their method relies on CAN frame identifiers and selected portions of the data field to capture message periodicity, allowing it to detect frame modification and replay attacks. While the authors note that the approach could be adapted to other in-vehicle communication protocols, it is mainly demonstrated on CAN traffic. A notable drawback is the additional processing load placed on ECUs, which may interfere with timely system responses [Bari *et al.*, 2023]. These incidents significantly increased awareness among automotive original equipment manufacturers (OEMs) of the security limitations of legacy in-vehicle networks and motivated the adoption of stronger cybersecurity practices, including secure development lifecycles, continuous risk assessment, and monitoring mechanisms, as reflected in regulatory guidance such as the NHTSA Cybersecurity Best Practices for the Safety of Modern Vehicles [National Highway Traffic Safety Administration, 2020].

Types of Attacks on CAN Bus

Understanding the classification of CAN-based attacks is essential for designing effective IDS. The following categories represent the most significant types of attacks observed in CAN bus networks:

- Denial-of-Service Attacks: These attacks aim to flood the CAN bus with high-priority messages, preventing authorized ECUs from accessing the network. Attackers exploit the priority-based arbitration mechanism to gain control over communication, leading to failure or unresponsiveness in critical vehicle systems.
- Fuzzing Attacks: In fuzzing, attackers send corrupted, randomized, or unexpected messages to ECUs in an attempt to exploit software vulnerabilities or failure system functions.
- Spoofing and Masquerade Attacks: Spoofing involves injecting messages that impersonate an authorized ECU by replicating its message identifiers and timing characteristics. This can manipulate vehicle behavior or hide the presence of malicious activity.
- Replay Attacks: These involve capturing authorized CAN messages and retransmitting them at a later time to produce unauthorized actions. For example, replaying door-unlock messages can grant unauthorized access to the vehicle.
- Suspension Attacks: Attackers intentionally prevent specific ECUs from sending their periodic or control mes-

Table 3. CAN data frame fields descriptions.

Field	Description
Start of Frame (SOF)	A single dominant bit that signals the start of the message transmission.
Identifier (ID)	An 11-bit field indicating the message priority and type. In extended frames, this field is expanded with additional bits.
Remote Transmission Request (RTR)	A single-bit field used to request data. A dominant bit indicates a data frame, while a recessive bit indicates a remote frame.
Substitute Remote Request (SRR)	Present only in extended frames, replacing the RTR bit and used as a placeholder.
Identifier Extension (IDE)	Indicates whether the frame is standard (dominant bit) or extended (recessive bit).
Reserved Bits (r0, r1)	Reserved for future CAN protocol use and typically set to dominant.
Data Length Code (DLC)	A 4-bit field representing the number of data bytes (0 to 8) in the payload.
Data Field	The actual message payload, ranging from 0 to 8 bytes.
Cyclic Redundancy Check (CRC)	Contains a checksum used to detect errors in the transmitted frame.
Acknowledge (ACK)	Comprises two bits: an ACK slot overwritten by receivers to indicate successful reception, followed by an ACK delimiter.
End of Frame (EOF)	A sequence of seven recessive bits marking the end of the data frame.

sages, often by exploiting flow control or arbitration rules. This disrupts time-sensitive subsystems such as braking or steering.

- Data Falsification Attacks: Here, the attacker modifies the content of messages, such as changing Revolutions Per Minute (RPM) values, gear positions, or sensor readings. This can mislead vehicle control mechanism and compromise safety.

These attack types emphasize the importance of robust and adaptive IDS capable of identifying emerging threats. Effectively tackling this diverse range of threats remains a central goal in securing automotive in-vehicle networks [Almehdhar *et al*., 2024].

5.2 Overview of CAN IDS Datasets

Several datasets have been introduced to support research on CAN bus intrusion detection systems. As summarized in Table 4, most of the available datasets are publicly accessible, while a smaller subset remains proprietary or subject to restricted access. These datasets are fundamental to the development, training, and evaluation of IDS solutions for CAN bus security. However, they differ considerably in terms of vehicle diversity, traffic realism, attack coverage, labeling strategy, and data scale, all of which directly affect the validity and generalizability of reported detection performance.

Early datasets such as the TU Eindhoven CAN Bus dataset [Dupont *et al*., 2019b] and OTIDS [Lee *et al*., 2017] focused on controlled attack injection and provided multiclass annotations for common attack types including denial of service, fuzzing, replay, and spoofing. The widely adopted Car Hacking dataset [Song *et al*., 2020; Seo *et al*., 2018] emphasizes large scale message volumes and reproducibility, but its reliance on simulated environments and binary labeling limits realism and fine grained attack differentiation. In contrast, benign only datasets such as AEGIS [He *et al*., 2020] and the HCRL CAN signal extraction dataset [Song and Kim, 2021] provide high fidelity real vehicle traffic and are primarily suited for anomaly based or unsupervised IDS evaluation rather than supervised attack classification.

Most datasets apply binary labeling schemes that distinguish between normal and malicious traffic, although several more recent datasets support multiclass labeling to differentiate attack categories such as denial of service, spoofing, fuzzing, and masquerade attacks. Labeling is typically performed through controlled injection procedures or predefined attack timestamps, as observed in datasets such as SynCAN [Hanselmann *et al*., 2020] and HCRL SA [Han *et al*., 2018]. However, the labeling process and its consistency across datasets are often insufficiently documented, which limits reproducibility and complicates cross dataset performance comparison in IDS studies.

More recent datasets reflect a shift toward higher realism and richer feature representations. The CRYSYS dataset [Gazdag *et al*., 2023] combines real and simulated traffic with structured multiclass attack scenarios, while the REAL ORNL dataset provides real vehicle data with decoded signals, timing information, and advanced targeted attack labels. The XPeng G9 based dataset [Sun *et al*., 2024] further extends this trend by incorporating real world traffic and hybrid attack scenarios at both message and signal levels.

6 Recent Intelligent IDS Approaches

As cybersecurity threats targeting the CAN bus continue to increase, IDS have become a critical component of IVN security architectures. In response, a wide range of intelligent IDS approaches have been proposed in the literature, differing in how CAN traffic is analyzed, how abnormal behavior is identified, and how malicious activity is classified. Based on a review of recent and representative studies, these approaches can be broadly grouped into ML, DL, and hybrid IDS methodologies. Figure 3 depicts the three categories

Table 4. Selected CAN bus intrusion detection datasets (chronologically ordered).

Dataset	Year	Vehicle Diversity	Classes	Features	Size	Labeling	Realism	Availability	Access Link
TU Eindhoven	2016	High (real vehicle)	Diagnostic, fuzzing, replay, suspension, DoS	Timestamp, CAN ID, DLC, payload	98 MB	Multiclass	Real	Public	link
OTIDS	2017	Low (real vehicle, injected attacks	Normal, flooding, spoofing, replay, fuzzing	Timestamp, CAN ID, DLC, data	8,694,507 messages	Multiclass	Real (Kia Soul)	Public	link
Car-Hacking	2018	Medium (real Toyota, injected attacks)	DoS, fuzzing, spoofing, injection	Timestamp, CAN ID, DLC, data	17,558,462 messages	Binary	Semi-synthetic	Public	link
AEGIS	2019	High (real vehicle)	None (benign only)	Sensor information	2.5 h	N/A	Real	Public	link
SynCAN	2020	Medium (simulated)	Normal, masquerade, insertion, suspension	CAN ID, payload, timing	24 h	Multiclass	Synthetic	Public	link
HCRL-SA	2020	High (real vehicle)	Flooding (DoS), fuzzing, malfunction	Timestamp, CAN ID, DLC, data bytes	1,736,840 messages	Multiclass	Real	Public	link
HCRL CAN Signal Extraction and Translation	2021	High (real vehicle)	None (benign only)	Timestamp, CAN ID, data bytes	∼5.1 million messages	None	Real	Public	link
CRYSYS Lab	2023	Mixed	Fabrication, masquerade	Timestamp, CAN ID, DLC, payload	2 h 33 m 43 s	Multiclass	Real and simulated	Public	link
REAL ORNL	2023	High (real vehicle)	Fabrication, masquerade, advanced targeted attacks	Raw CAN data, decoded signals, timing, attack labels	3.5 h	Binary and multiclass	Real	Public	link
XPeng G9-based	2024	High (real vehicle)	Normal, spoofing, fuzzing, replay, suppression, hybrid	CAN ID, payload, timing, signal-level	821,294 messages	Multiclass	Real-world	Restricted	Not public

along with representative foundational models.

In this work, IDS approaches are organized according to how CAN traffic information is represented and utilized for detection, how models learn to distinguish normal from malicious behavior, and the type of algorithms employed. In addition, the organization accounts for reported practical considerations, including computational efficiency, interpretability, and feasibility under in-vehicle deployment constraints

Tables 6, 8 and 10 provide a structured summary of the key characteristics of the reviewed ML, DL, and hybrid learning-based CAN bus intrusion detection studies, respectively. Specifically, for each study presented in these sections, the reported information includes the model architecture, attack types considered, real-time and deployment feasibility, datasets used for evaluation and lightweight feasibility. Each comparison table is followed by an additional table that reports the the highest reported detection metrics for each study, as presented in Tables 7, 9 and 11.

In this survey, real-time and deployability are marked only when the paper explicitly states real-time capability, or the evaluation includes real-vehicle testing, or the authors provide quantitative/qualitative evaluation consistent with in-vehicle constraints (e.g., ECU-oriented runtime/latency, embedded feasibility discussion, or similar). Reported training or testing time alone is not treated as sufficient evidence of real-time operation, particularly when experiments are conducted on desktop or server-class hardware; studies lacking explicit runtime evaluation or realistic deployment context are not considered to demonstrate real-time or deployment feasibility. In addition, each study is examined with respect to whether it explicitly claims lightweight design, supported by clear indications such as reduced parameter count, memory/compute footprint, simplified architecture, or embedded suitability discussion. Finally, the validation label was assigned based on whether a study performed evaluation beyond offline metric reporting, such as using simulation platforms, real vehicles, embedded hardware, or online/real-time testing. Studies relying solely on offline dataset-based metrics were classified as not validated.

6.1 Machine Learning-Based Intrusion Detection for CAN Bus Security

Machine learning based intrusion detection systems represent one of the most established and widely explored approaches for securing CAN bus communications. In this survey, the studies grouped under the ML category are classified as such because they rely on explicit, domain-informed feature engineering rather than learning directly from raw CAN traffic. Specifically, CAN messages are analyzed using protocol knowledge to extract meaningful features such as message identifier patterns, payload-related statistics, and timing or periodicity behavior, which are then provided as structured inputs to classical learning algorithms. Detection is performed using widely adopted ML models, including Support Vector Machines, Decision Trees, Random Forests, gradient-boosting methods such as XGBoost, k-Nearest Neighbors, Naïve Bayes, and isolation-based anomaly detectors [Almehdhar *et al*., 2024]. For example, SVMs effectively handle high-dimensional engineered CAN features [Tanksale, 2019; Wang *et al*., 2018; Bari *et al*., 2023], tree-based models offer transparent and low-latency decision logic [Alfardus and Rawat, 2021; Gundu and Maleki, 2022; Guezzaz *et al*., 2021], and gradient-boosting methods such as XGBoost provide robust generalization across diverse attack patterns while maintaining computational efficiency [Zhou and Wang, 2018; Bari *et al*., 2023; Li *et al*., 2023]. These models operate on fixed feature representations and do not perform automatic hierarchical or end-to-end representation learning, which clearly distinguishes them from deep learning–based approaches. Based on differences in learning strategy, model family, and evaluation or deployment context,while sharing the same feature-engineering paradigm, the reviewed ML-based IDS are further categorized into supervise, one-class and novelty-based detectors, tree-based and gradient-boosting models, ensemble learning strategies, graph-enhanced classical ML approaches, and deployment-oriented implementations.

6.1.1 Supervised Classical Machine Learning

Early ML-based CAN IDS primarily adopt supervised learning, where labeled CAN traffic is used to distinguish normal and malicious behavior. Table 5 highlights the fundamental design considerations in how CAN bus IDSs use labeled versus unlabeled data. Approaches relying on labeled data perform well under controlled conditions but often struggle to scale as attack patterns evolve, whereas unlabeled-data methods better reflect in-vehicle operating environments, but with increased uncertainty in detection outcomes. This trade-off motivates growing interest in semi-supervised and hybrid IDS strategies that aim to balance robustness with practical deployability. Commonly employed models include SVM, DT, Random RF, KNN, MLP, and probabilistic models. These approaches rely on engineered features

Support Vector Machines are commonly adopted in supervised CAN intrusion detection due to their ability to learn maximum-margin decision boundaries and effectively handle high-dimensional handcrafted features through kernel functions. Their strong generalization capability and robustness on small-to-medium-sized datasets make them suitable for security-oriented anomaly detection tasks with relatively low inference cost [Abdullah and Abdulazeez, 2021]. However, empirical evaluations demonstrate that while SVMs achieve high detection accuracy for certain attack types, their training and inference overhead can be prohibitive compared to tree-based alternatives, particularly under fuzzy or highly imbalanced attack scenarios [Tanksale, 2019; Moulahi *et al*., 2021]. Comparative studies using real CAN traffic further show that Decision Trees and Random Forests consistently outperform SVMs in terms of execution time while maintaining comparable or superior detection performance [Bari *et al*., 2023; Alfardus and Rawat, 2021].

Several works emphasize feature engineering as a key factor in improving supervised ML performance. By incorporating temporal features such as time intervals between CAN messages, detection accuracy can be significantly improved without increasing model complexity [Gundu and Maleki, 2022]. Large-scale evaluations confirm that tree-based models and classical ensembles provide a favorable trade-off between accuracy, interpretability, and computa-

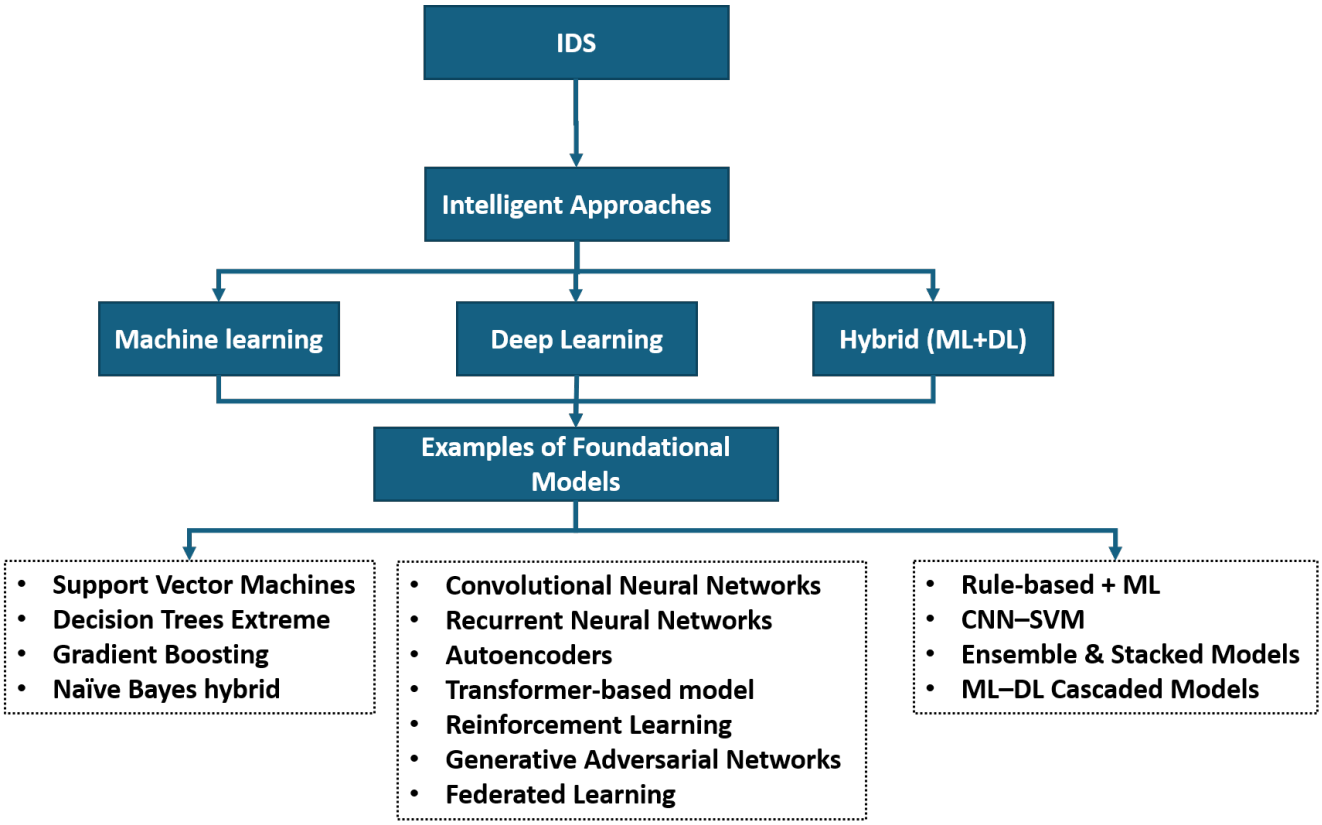

Figure 3. Intelligent IDS broad orgnization alongside foundational examples

tional efficiency, making them suitable for real-world vehicular deployments [Kalkan and Sahingoz, 2020; Islam *et al*., 2024].

6.1.2 One-Class and Novelty-Based Anomaly Detection

To address the scarcity of labeled attack data and the emergence of zero-day threats, many studies adopt one-class learning and unsupervised anomaly detection. These methods model normal CAN behavior and identify deviations as potential intrusions, reducing reliance on attack-specific labeling.

One-Class SVM (OCSVM) is widely used in this context, often enhanced through meta-heuristic optimization techniques to improve boundary selection and detection stability. Optimized OCSVM-based IDS have demonstrated high detection rates for flooding and message injection attacks using real vehicle data, although several studies remain limited to offline evaluation [Al-Saud *et al*., 2019]. Extensions incorporating timing and frequency features report real-time detection latencies on the order of milliseconds, supporting feasibility for in-vehicle deployment [Avatefipour *et al*., 2019].

Alternative one-class approaches such as Support Vector Data Description (SVDD) and Isolation Forest further improve robustness by modeling temporal dependencies and reducing false negatives [Li *et al*., 2021; Sharmin and Mansor, 2021]. Notably, ZeroCAN introduces a per-ECU anomaly modeling strategy using feature-driven SVMs, demonstrating reliable zero-day attack detection across multiple vehicle platforms with very low false-positive rates [Rendel *et al*., 2025]. Intrusion prevention extensions that actively discard malicious frames further highlight the practicality of unsupervised ML for real-time automotive security [De Araujo-Filho *et al*., 2021].

6.1.3 Tree-Based and Gradient-Boosting Models

Tree-based classifiers and gradient-boosting techniques represent one of the most effective ML families for CAN intrusion detection. Models such as Random Forest, Extra Trees, and XGBoost excel at capturing nonlinear relationships among handcrafted CAN features while maintaining fast inference.

Extensive evaluations across multiple real CAN datasets demonstrate that gradient-boosted decision trees achieve consistently high detection accuracy with low false-positive rates across diverse attack types, including spoofing, fuzzing, and replay attacks [Anjum *et al*., 2022]. Feature selection strategies further reduce dimensionality and computational overhead while preserving performance, reinforcing the suitability of tree-based ML for embedded environments [Aksu and Aydin, 2022].

6.1.4 Ensemble Learning Strategies

To improve robustness and generalization, several studies employ ensemble learning through stacking, blending, or voting mechanisms. By combining complementary classifiers, ensemble-based IDS can mitigate weaknesses of individual models while preserving lightweight operation.

Stacked ensembles integrating Random Forest, XGBoost, Decision Trees, and related models achieve high detection accuracy across multiple attack scenarios without relying on deep learning [Alalwany and Mahgoub, 2024; Wu and Tao, 2024]. More advanced frameworks such as BEPCD introduce adaptive blending and confidence-driven voting,

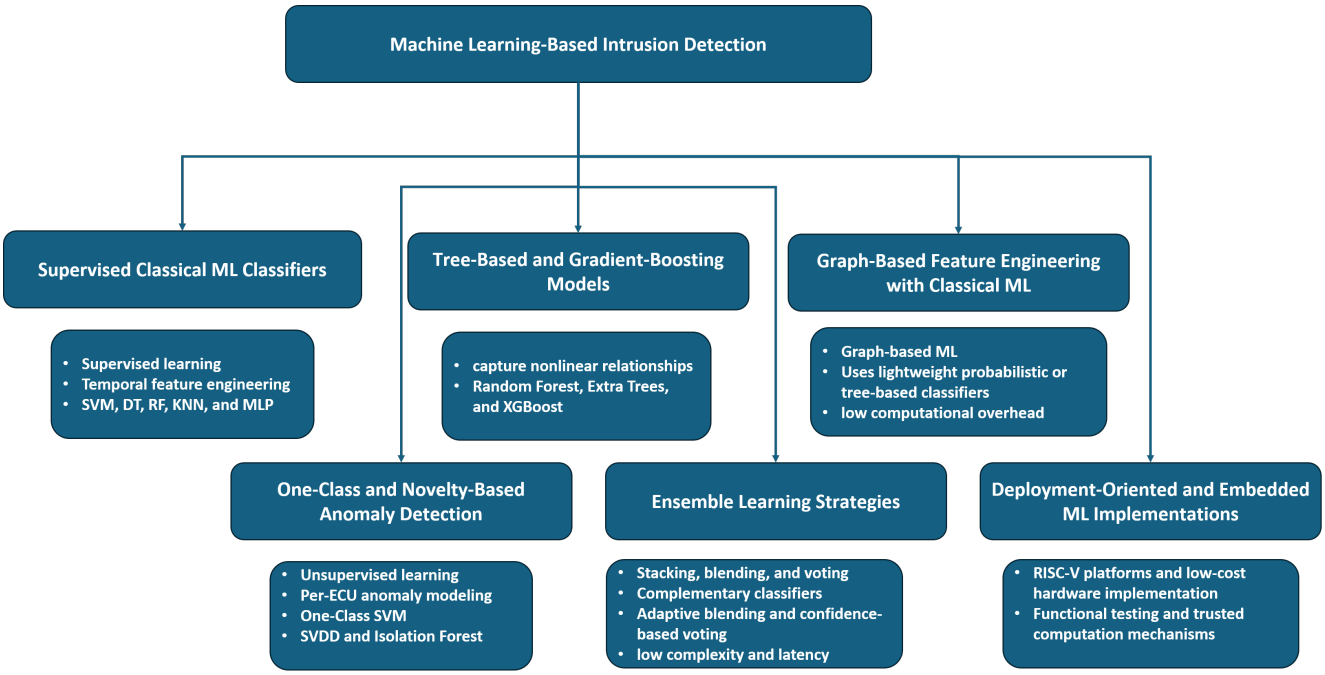

Figure 4. Taxonomy of ML-based IDS for CAN bus security.

Table 5. Comparison of labeled and unlabeled data in CAN bus IDSs.

Type of Data	Definition	Learning Paradigm	Example in CAN Bus IDS	Strengths	Limitations
Labeled	CAN traffic annotated with predefined classes (e.g., normal, attack)	Supervised learning	CAN messages labeled as normal or malicious during training	High detection accuracy; clear decision boundaries	Requires costly, attack-specific labeling and limited generalization to unseen attacks
Unlabeled	CAN traffic without predefined labels; patterns inferred automatically	Unsupervised / semi-supervised learning	Raw CAN traffic where anomalies are detected based on deviations from learned normal behavior	No labeling required; improved detection of unknown attacks	Higher false-positive rates and reduced interpretability of detection results

accompanied by detailed complexity and latency analysis, demonstrating compatibility with automotive-grade ECUs and real-time deployment requirements [Xu *et al*., 2025a].

6.1.5 Graph-Based Feature Engineering with Classical ML

Recent studies enhance classical ML by modeling CAN traffic as temporal or directed graphs, enabling extraction of structural and relational features that capture communication dynamics between message identifiers. These graph-derived features are then classified using lightweight probabilistic or tree-based models.

Graph-based Gaussian Naive Bayes and centrality-driven feature representations achieve high detection accuracy while significantly reducing training data requirements [Walid *et al*., 2024]. By avoiding deep neural architectures, graph-enhanced ML approaches maintain interpretability and low computational overhead, making them particularly attractive for resource-constrained automotive platforms

6.1.6 Deployment-Oriented and Embedded ML Implementations

While many ML-based CAN IDS are evaluated offline, a growing body of work explicitly addresses embedded deployment. Implementations on RISC-V platforms and low-cost hardware demonstrate that classical ML models such as Decision Trees and Random Forests can achieve sub-millisecond inference latency under real CAN traffic conditions [Bonomo *et al*., 2024]. Additional studies integrate functional testing and trusted computation mechanisms, reinforcing the practicality of ML-based IDS for safety-critical automotive subsystems [Al Maruf and Azim, 2024].

Probabilistic modeling approaches using Bayesian Networks further illustrate that interpretable ML can be deployed on embedded hardware for intrusion detection in simulated and real-time vehicular environments, albeit with moderate detection performance compared to ensemble-based methods [Casillo *et al*., 2019].

Table 6 summarizes the surveyed ML–based intrusion de-

Table 6. ML-Based IDS Studies for CAN Bus Security.

Study	Model / Approach	Attack / Scenario	Real-Time & Deployable	Dataset Type	Lightweight	Validation Feasibility
[Tanksale, 2019]	SVM	Injected anomalies / signal manipulation	×	Synthetic anomalies	×	×
[Al-Saud *et al*., 2019]	SVM	DoS	×	Real CAN data	×	×
[Avatefipour *et al*., 2019]	OCSVM, MBA	Injection and frequency manipulation	✓	Private real vehicle (Dodge RAM, DoS injection)	×	×
[Yang *et al*., 2019]	DT, RF, ET, XGBoost	DoS, Fuzzy, Spoofing, Port scan, Brute force	×	Car-Hacking, CI-CIDS2017	×	×
[Nazakat and Khurshid, 2019]	MLP, DT	DoS, Fuzzy	×	HCRL dataset	×	×
[Kalkan and Sahingoz, 2020]	RF, Bagging DT, AdaBoost, NB, LR, ANN	DoS, Fuzzy, Spoofing (RPM, Gear)	×	Public	×	×
[Moulahi *et al*., 2021]	SVM, DT, RF, MLP	DoS, Fuzzy, Impersonation	×	Kia Soul	×	×
[De Araujo-Filho *et al*., 2021]	IPS, iForest, OCSVM	Fuzzing, Gear spoofing, RPM spoofing	✓	Public real-vehicle	×	✓
[Li *et al*., 2021]	M-SVDD, G-SVDD	Data modification, out-of-order messages	×	Private (Luxgen U5, Kia Soul)	×	×
[Sharmin and Mansor, 2021]	iForest, timing-based anomaly detection	Gear spoofing, RPM spoofing, DoS, Fuzzy injection	×	Public	✓	×
[Alfardus and Rawat, 2021]	KNN, RF, SVM, MLP	DoS, Impersonation, Fuzzy	×	Real CAN traffic	×	×
[Anjum *et al*., 2022]	XGBoost	DoS, Fuzzy, Spoofing, Flooding, Malfunction	×	Real: Hyundai Sonata, Kia Soul, Chevrolet Spark; Car-Hacking	×	×
[Gundu and Maleki, 2022]	RF, KNN, XGBoost	DoS, Fuzzy, Impersonation	×	Public real-vehicle	×	×
[Aksu and Aydin, 2022]	MGA + (DTC, SVC, KNC, LRC, LDAC)	DoS, Fuzzing, Spoofing (Gear, RPM)	×	HCRL-Car Hacking, UNSW-NB15, CIC-IDS2017	×	×
[Bari *et al*., 2023]	SVM, DT, KNN	DoS, Impersonation, Fuzzy, Flooding, Malfunction	×	Kia Soul and Chevrolet Spark	×	×
[Alalwany and Mahgoub, 2024]	RF, DT, XGBoost	DoS, Fuzzing, Replay, Spoofing	✓	Real: IEEE DataPort	×	✓
[Wu and Tao, 2024]	DT, ET, XGBoost	DoS, Gear spoofing, RPM spoofing, Fuzzy	×	Car-Hacking	×	×
[Walid *et al*., 2024]	Graph-based + (GNB, LDA, DT, GB, XGBoost)	DoS, Fuzzy, Spoofing, RPM, Gear, mixed attacks	×	Public: Car-Hacking	×	×
[Bonomo *et al*., 2024]	KNN, DT, RF, XGBoost, K-Means	DoS, Impersonation	✓	Public + real vehicle (HCRL, Renault Oroch)	×	✓
[Al Maruf and Azim, 2024]	NB, SVM, DNN	Frequency and data anomalies	×	Synthetic and real (Hyundai)	×	✓
[Xu *et al*., 2025a]	RF, ET, XGBoost, CatBoost	DoS, Spoofing, Fuzzing, Replay	✓	Public: HCRL-CHDC, CIC-IOV2024	✓	✓
[Rendel *et al*., 2025]	SVM	DoS, Fuzzy, Spoofing, data-altering	✓	CAN Train and Test	✓	✓
[Palma *et al*., 2025]	RF, ET, XGBoost, AdaBoost, LR, DNN	DoS and spoofing (gas, speed, steering, RPM)	×	CIC-IOV2024	×	×

tection studies and provides a detailed comparative analysis across multiple dimensions, including the employed model or approach, attack scenario, real-time and deployability claims, dataset type, lightweight characteristics, and validation feasibility. The comparison indicates that strong detection performance can be achieved using established techniques such as SVMs, DTs, and tree-based ensemble methods across diverse attack types and real-world CAN datasets. Lightweight suitability is evaluated based on evidence reported in the literature, including inference or execution time, algorithmic complexity, and explicit discussion of deployment under in-vehicle ECU constraints, as reflected in the table.To clearly summarize the reported detection performance across the reviewed studies, an additional table is provided that presents only the best detection metrics reported in the literature. This information is summarized in Table 7.

Overall, ML-based IDS techniques offer strong detection capabilities, adaptability, and computational efficiency for CAN bus security. Tree-based and ensemble models stand out for their accuracy and runtime performance, while unsu-

Table 7. Best reported detection metrics of ML-based IDSs for CAN bus security.

Study	Best Detection Metrics (As Reported)
[Avatefipour *et al*., 2019]	TPR: up to 97.01%; FPR: as low as 6.45%
[Al-Saud *et al*., 2019]	Detection rate: 96.1%; False alarm rate: 6.45%; Miss rate: 3.9%; Correct reject rate: 93.55%
[Nazakat and Khurshid, 2019]	Accuracy: 100%
[Yang *et al*., 2019]	Accuracy: 100%; Detection rate: 100%; FAR: 0%; F1-score: 1.0
[Kalkan and Sahingoz, 2020]	F1-score: 1.00 (gear spoofing); 0.99 (fuzzy); 0.98 (DoS); 0.94 (combined dataset)
[Moulahi *et al*., 2021]	Not reported
[Sharmin and Mansor, 2021]	AUC: 0.966 (gear spoofing); 0.974 (RPM spoofing)
[Li *et al*., 2021]	Accuracy: up to 99.535%; Recall: up to $\sim$90% (out-of-order attacks); Precision: $>$90%
[De Araujo-Filho *et al*., 2021]	Accuracy: up to 99.92%; Precision: up to 99.57%; Recall: up to 100%; F1-score: up to 99.48%
[Alfardus and Rawat, 2021]	Accuracy: up to 100% (DoS, impersonation); 99.99% (fuzzy); F1-score: 1.0; FPR: 0.0%
[Anjum *et al*., 2022]	Accuracy: 99.99%; Precision: 100%; Recall: 100%; F1-score: 0.9996; FPR: 0.0000
[Gundu and Maleki, 2022]	Accuracy: 97%; Precision: 0.97; Recall: 0.97; F1-score: 0.97
[Aksu and Aydin, 2022]	Accuracy, Precision, Recall, F1-score, and AUC: up to 100%
[Bari *et al*., 2023]	Accuracy: up to 99.9%; Recall: $\approx$1.0
[Wu and Tao, 2024]	Accuracy: 99.21%; Precision: 95.22%; Recall: 98.84%; F1-score: 96.98%; FPR: 0.95%; FNR: 1.95%
[Alalwany and Mahgoub, 2024]	Accuracy: 98.5%; Precision: 98.7%; Recall (Detection Rate): 98.5%; F1-score: 98.5%
[Al Maruf and Azim, 2024]	Accuracy: 99.3%; SVM: 99.07%; Naïve Bayes: 96.96%
[Bonomo *et al*., 2024]	F1-score: 100%; 99.97% (DoS + impersonation, KNN); 99.51% (impersonation, DT/RF/XGBoost)
[Walid *et al*., 2024]	Accuracy: 99%; Recall: 100%; F1-score: 0.99; Training time: $\approx$0.0044 s (GaussianNB, 30% data)
[Palma *et al*., 2025]	Accuracy: 100%; Precision: 100%; Recall: 100%; F1-score: 100%
[Rendel *et al*., 2025]	TPR: 100%; FPR: $< 0.01\%$; AUC: 1.0
[Xu *et al*., 2025a]	Accuracy: 99.96–100%; Precision: 99.99–100%; Recall: 99.99–100%; F1-score: 99.99–100%

pervised and probabilistic methods contribute to robustness against emerging threats. The integration of timing features, multi-vehicle datasets, and hybrid modeling strategies continues to push the boundaries of intelligent intrusion detection in automotive systems.

6.2 Deep Learning-Based Intrusion Detection for CAN Bus Security

Recently, DL models have emerged as powerful tools for enhancing the security of in-vehicle communication systems, particularly the CAN bus. Deep learning techniques are well suited for capturing complex attack patterns and modeling large-scale, high-dimensional CAN traffic data. Inspired by the human brain, DL models utilize interconnected artificial neurons to automatically learn hierarchical feature representations, enabling effective detection of subtle and evolving cyberattacks in modern vehicular networks.

To enable a structured and comparative analysis, the following discussion categorizes DL–based CAN IDSs into four primary classes: supervised deep learning approaches, unsupervised and self-supervised methods, advanced spatiotemporal and hybrid representation models, and lightweight, deployment-oriented solutions. Fig. 5 illustrates the adopted taxonomy and highlights the main architectural themes and representative techniques within each category. This orga-

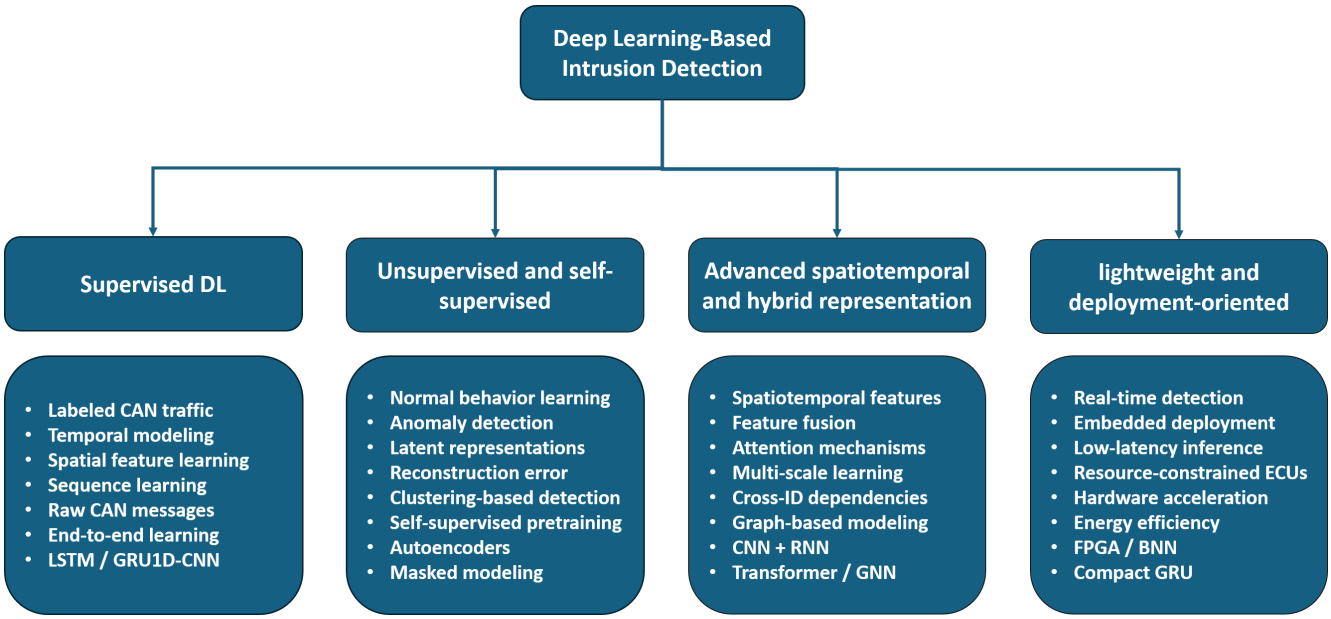

Figure 5. Taxonomy of DL-based IDS for CAN bus security.

nization emphasizes both the methodological characteristics and key implementation considerations of the surveyed approaches. A more detailed quantitative comparison is provided in Table 8 and Table 9 which summarizes key aspects of each study, including the model type, attack scenarios considered, real-time deployment feasibility, datasets used, whether lightweight operation is claimed, and whether any validation beyond offline evaluation is provided. In addition to the best reported evaluation metrics.

6.2.1 Supervised Deep Learning for CAN Bus Intrusion Detection

This section reviews deep learning–based intrusion detection approaches for in-vehicle CAN networks that learn discriminative traffic representations directly from raw or minimally processed CAN messages. The studies in this category primarily employ neural architectures such as recurrent and convolutional networks to capture temporal dependencies, spatial patterns, or their combination in CAN traffic, enabling effective detection of known attack behaviors. While these methods often achieve high detection accuracy under supervised learning settings, most evaluations remain limited to offline or workstation-based experiments, with real-time ECU deployment constraints rarely addressed explicitly.

Hossain et al. introduced a deep learning–based intrusion detection approach for CAN networks using LSTM models. By modeling CAN traffic as temporal sequences, the study demonstrated that recurrent neural networks can effectively capture sequential dependencies inherent in CAN communication [Hossain *et al*., 2020b]. Authors in [Hossain *et al*., 2020a] explored a CNN-based IDS operating directly on raw CAN message fields. Using a one-dimensional convolutional architecture. A study by [Amato *et al*., 2021], investigated the use of feedforward neural networks, including multilayer perceptrons, for CAN bus intrusion detection. By modeling CAN payload bytes as feature vectors, the study showed that relatively simple deep architectures can learn malicious injection patterns without protocol reverse engineering. Aurhors in[Samir *et al*., 2024] presented a comparative deep learning study evaluating LSTM and one-dimensional CNN models for CAN intrusion detection across multiple datasets. The work highlights the complementary roles of temporal and spatial feature learning in CAN traffic analysis, while remaining limited to offline evaluation scenarios. Another study proposed an end-to-end deep learning intrusion detection framework for CAN bus communication in autonomous vehicles. The approach employs a hybrid CNN–BiLSTM architecture to learn local signal patterns and long-term temporal dependencies directly from raw CAN messages, eliminating manual feature engineering and improving robustness against malicious behavior [Siddiqui *et al*., 2025].

Authors in [Rai *et al*., 2025] presented a framework for CAN bus security, evaluating LSTM, GRU, Bi-LSTM, and VGG-16 models across multiple public automotive datasets. The study showed that recurrent models capture temporal dependencies in CAN traffic, while the VGG-16–based CNN achieved stable performance when spatial representations were incorporated, though the evaluation was conducted on desktop hardware without ECU deployment validation. Another study proposed a system for in-vehicle CAN networks that sequentially combines RNN and LSTM models. The framework processes CAN messages as sequential data, using cascaded temporal modeling to improve discrimination between normal traffic and multiple attack behaviors without relying on handcrafted features [Al-Aql and Al-Shammari, 2024]. Authors in [Samir *et al*., 2024] proposed a deep learning–based intrusion detection framework for in-vehicle CAN bus security using recurrent and convolutional neural architectures. The study evaluates LSTM and one-dimensional CNN models across multiple public and simulated datasets.

6.2.2 Unsupervised and Self-Supervised Deep Learning

Unsupervised and self-supervised deep learning approaches have gained increasing attention for CAN bus intrusion detec-

Table 8. DL-Based IDS Studies for CAN Bus Security.

Study	Primary Models / Approaches	Attack / Scenario	Real-Time & Deployable	Dataset Type	Lightweight Claim	Validation Feasibility
Hossain et al. (2020) Hossain *et al.* [2020b]	LSTM (vanilla and stacked LSTM)	DoS, Fuzzing, Spoofing	×	NAIST (real Toyota vehicle with injected attacks)	×	×
Hanselmann et al. (2020) Hanselmann *et al.* [2020]	CANet (multi-ID CANet; LSTM-based unsupervised autoencoder)	Plateau, Continuous change, Playback, Flooding, Suppress	×	Real vehicle (private) + Public synthetic (SynCAN)	×	×
Mehedi et al. (2021) Mehedi *et al.* [2021]	DTL, P-LeNet	Flooding, Fuzzing, Spoofing	×	Real CAN bus traffic	×	×
Amato et al. (2021) Amato *et al.* [2021]	NN, MLP	DoS, Fuzzy, Gear spoofing, RPM spoofing	×	Public	×	×
Narasimhan et al. (2021) Narasimhan *et al.* [2021]	Autoencoder and Gaussian Mixture Model	DoS, Fuzzy	✓	Public	×	×
Rajapaksha (2022) Rajapaksha *et al.* [2022]	CAN-CID (Shallow GRU-based ensemble)	Fuzzing, Injection, Masquerade, DoS, Spoofing (RPM, Gear), Flooding, Malfunction	✓	ROAD + HCRL	✓	✓
Lin et al. (2022) Lin *et al.* [2022]	VGG16 and XBoost	DoS, Fuzzy, Spoofing (Gear and RPM)	×	HCRL	×	×
Desta et al. (2022) Desta *et al.* [2022]	Rec-CNN, Binary and Multi-class classifiers	DoS, Fuzzy, Spoofing (Gear, RPM)	✓	Public and real-vehicle	✓	✓
Alkhatib et al. (2022) Alkhatib *et al.* [2022]	CAN-BERT	Flooding, Fuzzy, Malfunction	✓	Public	×	×
Ma et al. (2022) Ma *et al.* [2022]	GRU, LSTM, AdaBoost, SVM	DoS, Fuzzy, Spoofing	✓	Public	✓	✓
Xun et al. (2022) Xun *et al.* [2021]	VehicleEIDS, Deep SVDD	External attacks – frame forgery, replay, DoS, reverse-engineering	✓	Luxgen U5 SUV and Buick Regal	×	×
Wei et al. (2023) Wei *et al.* [2023]	AMAEID; DT, KNN, LinearSVC	Synthetic byte-flip message anomaly	×	OTIDS	×	×
Devnath et al. (2023) Devnath [2023]	GCN	DoS, Fuzzy, Spoofing, Replay, Mixed	×	OTIDS	×	×
Gao et al. (2023) Gao *et al.* [2023]	CNN-BiLSTM with Multi-Head Attention	DoS, Spoofing (Gear and RPM), Fuzzy	×	Car-Hacking	×	×
Levy et al. (2023) Levy *et al.* [2023]	CAN-LOC (Autoencoder + CNN-based voltage side-channel IDS)	ECU spoofing, ECU insertion/replacement, VBS-D poisoning, and silent intruders	✓	CAN bus prototype + real vehicle	×	✓
Chougule et al. (2024) Chougule *et al.* [2024]	HybridSecNet (two-step LSTM + CNN architecture)	DoS, Fuzzy, Gear, RPM	×	Car-Hacking	×	×
Samir et al. (2024) Samir *et al.* [2024]	CNN, LSTM	DoS, Fuzzy, Spoofing, Replay	×	Public (Car-Hacking, CAN-Intrusion) + Simulated	×	×
Al-Aql and Al-Shammari (2024) Al-Aql and Al-Shammari [2024]	Hybrid RNN + LSTM	DoS, Fuzzy, Impersonation	×	Public – OTIDS	×	×
Kang (2024) Kang *et al.* [2024]	CANival (TIL + LSTM-Autoencoder)	Insertion, Suspension, Masquerade	✓	X-CANIDS + SynCAN	×	✓
Shi et al. (2024) Shi *et al.* [2024]	IDS-DEC (LSTM-CNN Autoencoder + Deep Embedded Clustering)	DoS, Fuzzy, Spoofing (Gear, RPM)	×	Public – HCRL + Car Hacking (Attack and Defense)	×	×
Zhou et al. (2024) Zhou *et al.* [2024]	LSTM + CNN + Transformer	DoS	×	Car-Hacking	×	×
Rangsikunpum et al. (2024) Rangsikunpum *et al.* [2024]	BNN (CNV1w1a backbone), C2F	DoS, Fuzzy, Spoofing (Gear and RPM)	✓	Car-Hacking dataset (HCRL)	✓	✓
Xu (2025) Xu *et al.* [2025b]	MULSAM (MD-LSTM + Self-Attention, FPGA-accelerated)	DoS, Fuzzy, Spoofing, Replay, Delete	✓	Public (4TU.ResearchData) + simulated attacks	✓	✓
Rai et al. (2025) Rai *et al.* [2025]	LSTM, GRU, Bi-LSTM, VGG-16	DoS, Fuzzy, Spoofing (Gear and RPM), Impersonation, Flooding, Malfunction	×	Car-Hacking, OTIDS, Survival Analysis	×	×
El-Fatyany et al. (2025) El-Fatyany *et al.* [2025]	EF-IDS (UNET-LSTM)	DoS, Spoofing, Fuzzy	✓	Tesla Model 3 (2022) and LeapMotor C10 (2024)	✓	✓
Wang et al. (2025) Wang *et al.* [2025]	WGAN-GP + CNN + LSTM	Injection, DoS, Replay, Drop, Tampering	✓	Public CAN datasets + CANOE	✓	✓
Aljabri et al. (2025) Aljabri *et al.* [2025]	MLP	DoS, Fuzzy, Spoofing (Gear, RPM)	✓	Car-Hacking dataset + Car Hacking: Attack and Defense	✓	✓
Xin et al. (2025) Xin *et al.* [2025]	LUFT-CAN (unsupervised frequency–time analysis using FFT and lightweight CNN-based autoencoder)	DoS, Fuzzy, Spoofing (Gear and RPM)	✓	Tesla Model 3 (2022); LeapMotor C10 (2024); Car-Hacking Dataset (2020)	✓	✓

Table 9. Best reported detection metrics of DL-based IDSs for CAN bus security.

Study	Best Detection Metrics (As Reported)
[Hossain *et al*., 2020b]	Accuracy: 99.995%; Recall (TPR): ≈1.0; F1-score: ≈0.9999; AUC: 1.0; FPR: 0.00004; FNR: 0.0002
[Hanselmann *et al*., 2020]	Accuracy: 0.996; AUC: 0.983; TPR: 0.997; TNR: 0.999
[Hossain *et al*., 2020a]	Multiclass: Accuracy up to 99.99%, Recall ≈1.00, F1-score ≈0.999; Binary: Accuracy 100%, F1-score 1.00
[Mehedi *et al*., 2021]	Accuracy: 98.10%; Precision: 98.14%; Recall: 98.04%; F1-score: 97.83%; ROC-AUC: 95.42%
[Narasimhan *et al*., 2021]	Accuracy: 0.803; Precision: 0.747; Recall: 0.863; F1-score: 0.801
[Amato *et al*., 2021]	Precision: 0.974; Recall: 0.965; F1-score: 0.967; AUC: ≈0.989
[Ma *et al*., 2022]	DoS: Precision 0.9993, Recall 0.9991, F1-score 0.9992; Fuzzy: Precision 0.9932, Recall 0.9913, F1-score 0.9922; Spoofing: Precision 0.9995, Recall 0.9931, F1-score 0.9963
[Desta *et al*., 2022]	Accuracy: 0.999; TPR: 0.998; FPR: 0.002; ROC-AUC: ≈1.0; Latency: ≈117 ms
[Rajapaksha *et al*., 2022]	F1-score: 100%; FPR: 0%; FNR: ≈0%; Latency: 10 ms
[Lin *et al*., 2022]	Accuracy: 99.79%; Precision: 99.95%; Recall: 98.58%; F1-score: 99.24%; ROC-AUC: 99.95%
[Wei *et al*., 2023]	AUC: 0.923; AUPR: 0.885; Precision: 0.837
[Devnath, 2023]	Accuracy: 99.89%; Precision: 1.00; Recall: 1.00; F1-score: 1.00
[Levy *et al*., 2023]	Accuracy: up to 100%; ECU ID accuracy: 99.8–99.9%; Precision: >0.99; Recall: 1.00
[Gao *et al*., 2023]	Accuracy: 99.992%; Precision: 99.993%; Recall: 99.97%; F1-score: 99.97%
[Samir *et al*., 2024]	Accuracy: up to 99.20%; F1-score: up to 0.9928; FPR: as low as 0.0018; FNR: as low as 0.0053
[Chougule *et al*., 2024]	Accuracy: 99.595%; Precision: 99.5835%; Recall: 99.5799%; F1-score: 99.5801%
[Rangsikunpum *et al*., 2024]	Accuracy: 99.83%; Precision: 99.88%; Recall: 99.64%; F1-score: 99.76%
[Kang *et al*., 2024]	TPR: 0.960; TNR: 0.997; F1-score: 0.971
[Shi *et al*., 2024]	Accuracy: ≈99%; F1-score: ≈99%; FPR: ≈0.5%
[Zhou *et al*., 2024]	Accuracy: 100%; Precision: 100%; Recall: 100%; F1-score: 1.000; ROC-AUC: ≈0.9999
[Al-Aql and Al-Shammari, 2024]	Accuracy: 93%; Precision/Recall/F1-score (weighted): 0.93; AUC: 0.97–1.00
[Xu *et al*., 2025b]	Accuracy: 98.98% (software), 98.81% (FPGA); Precision/Recall/F1-score: ≈0.99; Latency: 1.88 ms
[Rai *et al*., 2025]	Accuracy: 100%; Precision: 100%; Recall: 100%; F1-score: 1.000
[Wang *et al*., 2025]	Accuracy: 99.9%; Precision: 99.9%; Recall: 99.9%; F1-score: 99.9%; Latency: 0.56 ms
[Aljabri *et al*., 2025]	Accuracy: 99.95%; Recall: 99.97%; F1-score: 99.97%; FPR: 0.01%; Latency: 0.17 ms
[El-Fatyany *et al*., 2025]	Accuracy: 100%; Precision: 100%; Recall: 100%; F1-score: 100%; FPR: 0; Latency: 0.0437 ms / 100 frames

tion because they can learn normal communication patterns directly from data, without relying on labeled attack samples. These methods typically model CAN traffic using representation learning, reconstruction, clustering, or predictive objectives, allowing them to capture normal behavior and identify deviations that may indicate anomalous or previously unseen

attack activity.

Narasimhan et al. proposed an unsupervised deep learning-based CAN intrusion detection framework that learns latent representations using a deep autoencoder and separates normal and malicious traffic via Gaussian Mixture Model (GMM) clustering, without requiring labeled data [Narasimhan *et al*., 2021]. Authors in [Shi *et al*., 2024] proposed IDS-DEC, which jointly learns spatiotemporal representations and performs deep embedded clustering using an LSTM–CNN autoencoder with entropy-regularized clustering. Another stuudy proposed AMAEID, which employs an attention-enhanced denoising autoencoder trained on unlabeled CAN payloads to detect abnormal message injection behavior [Wei *et al*., 2023]. Author proposed CANet, in [Hanselmann *et al*., 2020], which uses ID-specific LSTM subnetworks and an autoencoder-style reconstruction mechanism to model normal CAN behavior. Anomalies are identified based on reconstruction deviations, enabling detection of previously unseen attacks without labeled data. A lightweight study autoencoder-based approach by [Kim *et al*., 2025] operates exclusively on normal traffic, with anomaly detection performed using a single threshold derived via kernel density estimation. Authors in [Xin *et al*., 2025], proposed LUFT-CAN, which jointly analyzes temporal and frequency-domain characteristics of CAN traffic by leveraging spectral patterns of CAN ID sequences to distinguish normal and malicious behavior. Another model introduced by [Alkhatib *et al*., 2022], CAN-BERT, adapts the BERT language model to learn normal CAN identifier sequences using a masked language modeling objective. By treating CAN IDs as tokens and learning bidirectional contextual dependencies without labeled attack data, the framework enables effective detection of message injection anomalies while maintaining low inference latency.

6.2.3 Advanced Representation and Hybrid Deep Learning Models

This section reviews advanced representation learning and hybrid deep learning–based intrusion detection systems for in-vehicle CAN networks. Unlike single-model approaches, the studies in this category integrate multiple deep learning components or heterogeneous data representations, such as temporal dynamics, spatial correlations, attention mechanisms, and structural relationships, to capture complex CAN traffic behaviors.

Chougule et al. proposed HybridSecNet, a two-stage deep learning framework that combines temporal modeling and spatial feature extraction for CAN intrusion detection. An LSTM-based binary detector is followed by a CNN-based multiclass classifier, enabling reliable detection while reducing unnecessary computation [Chougule *et al*., 2024]. Authors in [Kang *et al*., 2024] introduced CANival, a multimodal CAN intrusion detection framework that fuses timing-based analysis with signal-level deep learning. By combining temporal and payload representations through decision-level fusion, CANival improves detection coverage across multiple attack types. Zhou et al. proposed a multi-scale intrusion detection framework that integrates LSTM, CNN, and Transformer models with attention-based feature fusion. The combination of temporal, spatial, and long-range dependency representations enhances robustness against diverse CAN attack behaviors [Zhou *et al*., 2024]. Gao et al. presented an attention-enhanced CNN-BiLSTM model that jointly captures spatial, temporal, and contextual dependencies in CAN traffic, enabling accurate multi-class intrusion detection across diverse attack scenarios [Gao *et al*., 2023]. Authors in [Wang *et al*., 2025] proposed a hybrid CAN intrusion detection framework combining WGAN-GP–based data augmentation with a CNN-LSTM prediction model. By jointly addressing data imbalance and temporal dependencies, the framework achieves high detection performance with low-latency real-time feasibility. Another study by [El-Fatyany *et al*., 2025] introduced EF-IDS, a hybrid deep learning IDS that integrates UNET-based spatial feature extraction with LSTM-based temporal modeling, demonstrating real-time feasibility on embedded automotive hardware. Authors proposed Rec-CNN, in [Desta *et al*., 2022], which reformulates CAN intrusion detection using recurrence plot representations and CNN-based learning, enabling advanced spatiotemporal representation of sequence-level anomalies. Authors proposed an on-the-move intrusion detection framework that transforms CAN features into image representations and applies deep CNN-based feature learning, demonstrating the effectiveness of representation learning for CAN attack detection [Lin *et al*., 2022]. A GCNIDS, introduced by [Devnath, 2023], a graph-based intrusion detection system that models CAN traffic as dynamic graphs and applies graph convolutional networks to capture structural dependencies between messages.

6.2.4 Lightweight and Deployment-Oriented Deep Learning IDS

This section focuses on lightweight and deployment-oriented deep learning intrusion detection systems designed to operate under the strict computational, memory, and latency constraints of in-vehicle environments. The studies reviewed here emphasize practical deployability through compact model architectures, hardware-aware designs, physical-layer analysis, and real-time or near–real-time validation on embedded or automotive-grade platforms. Rather than prioritizing architectural complexity, these approaches demonstrate how deep learning can be adapted for efficient, reliable CAN intrusion detection in resource-constrained vehicular systems.

Ma et al. proposed a GRU-based lightweight intrusion detection system for in-vehicle CAN networks with a strong emphasis on real-time deployability. The framework employs a compact GRU architecture to reduce computational overhead and was physically deployed and evaluated on an in-vehicle embedded platform (NVIDIA Jetson Xavier NX), demonstrating real-time performance under realistic automotive resource constraints [Ma *et al*., 2022]. Authors in [Rajapaksha *et al*., 2022] proposed CAN-CID, a GRU-based intrusion detection system for in-vehicle CAN networks designed for near real-time deployment. The approach combines a shallow GRU model with a time-based detector and is trained exclusively on benign traffic. In another study, authors introduced MULSAM system for vehicular CAN buses with a focus on real-time deployment. To support practical in-vehicle use, the authors implemented a hardware-accelerated FPGA version, demonstrating low inference latency and strong energy

efficiency suitable for resource-constrained automotive platforms [Xu *et al*., 2025b]. A low-power intrusion detection system based on a BNN and implemented on an FPGA was introduced by [Rangsikunpum *et al*., 2024]. By exploiting 1-bit neural operations and a lightweight two-stage architecture, the proposed design significantly reduces computational complexity and power consumption, making it well suited for real-time deployment on cost-constrained automotive hardware platforms. Authors in [Zhang *et al*., 2024] proposed an intusion system for in-vehicle CAN networks that targets strict memory and latency constraints of embedded automotive platforms. By employing a binarized convolutional neural network with binarized weights and activations, the framework enables real-time intrusion detection under severe resource constraints while maintaining robust detection capability. A study proposed a real-time IDS that combines dynamic feature engineering with a lightweight deep neural network. The framework is designed to meet strict latency requirements and remain suitable for resource-constrained automotive platforms [Aljabri *et al*., 2025]. Authors in [Zhao *et al*., 2024] proposed a lightweight GRU-based IDS that maintains low latency and compact model size. Experimental evaluation on real CAN traffic demonstrates real-time detection performance, highlighting the suitability of the approach for deployment in resource-constrained vehicular environments. Authors in [Xun *et al*., 2021] proposed VehicleEIDS, an external IDS that operates at the physical layer using differential voltage signals. By exploiting hardware-level signal inconsistencies and applying a one-class deep learning model, the system enables protocol-independent detection of external attacks across different vehicles and operating conditions. Authors in [Levy *et al*., 2023] introduced CANLOC, a DL-based intrusion detection and localization system for in-vehicle CAN networks based on physical-layer voltage side-channel analysis. The framework enables intrusion detection, localization, and continuous ECU authentication without modifying the CAN protocol, demonstrating robustness to environmental variation and poisoning attempts.

6.3 Hybrid IDS Approaches for CAN Bus Security (ML + DL)

Hybrid approaches combine the strengths of both ML and DL to enhance detection robustness and adaptability. These systems typically apply rule-based filtering or machine learning preprocessing first. The refined inputs are then passed to a deep learning classifier for final decision-making [Chougule *et al*., 2024; Nichelini *et al*., 2023]. Recent developments also include federated and privacy-aware architectures to facilitate secure, distributed training across multiple vehicular nodes without compromising data ownership [Althunayyan *et al*., 2024]. By utilizing complementary detection mechanisms, hybrid IDS are well-suited to identify a broad range of threats, including composite attack chains, adaptive spoofing, coordinated DoS, and multi-modal intrusion scenarios. This layered design enhances the system's capability to detect both known and previously unseen threats in real-time automotive environments [Chougule *et al*., 2024]. Xun et al. proposed a side-channel–based intrusion detection system for in-vehicle CAN networks that combines physical-layer voltage feature extraction with learning-based fingerprinting. Differential voltage features from CAN-H and CAN-L signals are used to construct ECU fingerprints, which are analyzed using a FeatureBagging ensemble integrated with a CNN-based model to detect both external injection attacks and internally compromised ECUs without relying on payload semantics [Xun *et al*., 2023]. Zhang et al. proposed a federated graph neural network–based intrusion detection system for CAN bus security that models CAN traffic as directed message graphs and combines anomaly detection with deep graph-based classification. The two-stage pipeline integrates one-class detection with open-set attack recognition to detect known and unseen attacks while supporting real-time, privacy-preserving multi-vehicle deployment [Zhang *et al*., 2023]. Nichelini et al. proposed CANova, a modular intrusion detection framework for in-vehicle CAN networks that hybridizes rule-based, timing-based, distance-based, and learning-based detection methods. CAN identifiers are dynamically assigned to the most suitable detector based on extracted signal characteristics, enabling flexible and scalable detection of diverse CAN attack behaviors [Nichelini *et al*., 2023]. He et al. proposed HSNRFM, a hybrid intrusion detection model for in-vehicle CAN networks that combines neural feature enhancement with factorization-machine–based interaction modeling. A fully connected network improves raw CAN message representations, while neighborhood similarity and feature interaction modeling jointly enhance anomaly detection capability beyond conventional machine learning approaches [He *et al*., 2021]. Sun et al. proposed CCID-CAN, a hybrid intrusion detection framework that integrates a rule-based VBIN mechanism with Kalman filtering and a Naïve Bayes classifier for CAN bus attack detection. The system was evaluated on real-vehicle CAN traffic under multiple attack types and demonstrated high detection accuracy with low per-message latency suitable for real-time in-vehicle deployment [Sun *et al*., 2024]. Dwivedi proposed an anomaly-based intrusion detection system for in-vehicle CAN networks that combines classical machine learning and deep learning models. By incorporating temporal and attack-frequency features, the framework improves detection performance across multiple attack types, with ensemble and recurrent neural network models achieving the strongest results [Dwivedi, 2022]. Purohit and Govindarasu proposed a two-stage anomaly detection system for in-vehicle CAN networks that combines lightweight rule-based filtering with tree-based machine learning classifiers. Frequent attack patterns are filtered using rules, while Decision Tree, Random Forest, and XGBoost models detect more subtle anomalies, achieving an effective balance between detection performance and computational efficiency suitable for real-time deployment [Purohit and Govindarasu, 2022]. Kumar et al. proposed a hybrid AI-based intrusion detection framework for CAN bus security that integrates representation learning and neural classification. The approach uses a Restricted Boltzmann Machine for feature extraction followed by a dynamically optimized neural network, enabling detection of multiple CAN attack types using real CAN traffic, although evaluation is limited to offline experiments [Kumar *et al*., 2024]. Al Isawi et al. introduced a neuro-fuzzy intrusion detection framework for electric vehicle CAN bus security that combines fuzzy logic

Table 10. Hybrid-Based IDS Studies for CAN Bus Security.

Study	Model / Approach	Attack / Scenario	Real-Time & Deployable	Dataset Type	Lightweight	Validation Feasibility
He *et al*. [2021]	HSNRFM	Data-field manipulation / message anomaly (byte-level perturbation)	×	OTIDS	×	×
Dwivedi [2022]	Logistic Regression, SVC, RF, XGBoost, FFNN, LSTM	DoS, Fuzzy, Spoofing (Gear, RPM)	×	CAR-Hacking + real CAN traffic (NRC)	×	×
Purohit and Govindarasu [2022]	Rule-based + DT, RF, XGBoost	DoS, Fuzzy, Impersonation	×	HCRL (real vehicle CAN traffic)	✓	×
Xun *et al*. [2023]	Feature Bagging + CNN	External injection, ECU masquerading	✓	Real vehicle voltage signals (Buick Regal, Luxgen U5 SUV)	✓	✓
Nichelini *et al*. [2023]	CANova (rule + flow + VAR + RNN-AE)	Spoofing, Fuzzing, DoS, Drop, Replay, Continuous Change	✓	Real vehicle CAN data (ReCAN C-1) + synthetic attack injection	×	✓
Zhang *et al*. [2023]	GNN + SVDD + OpenMax	DoS, Fuzzy, Suspension, Replay, Spoofing	✓	Ford Transit + HCRL CAN data	✓	✓
Im *et al*. [2024]	RF, kNN, LP, SVM + rule-based	DoS, Spoofing (Gear, RPM), Fuzzy	×	Car-Hacking	✓	×
Kumar *et al*. [2024]	MCFO–DANN	DoS, Fuzzy, Impersonation	×	Javed et al. (2021)	×	×
Sun *et al*. [2024]	CCID-CAN (VBIN + Kalman Filter + Naïve Bayes)	Spoofing, Fuzzy, Replay, Suppression, Hybrid	✓	Real CAN traffic from XPeng G9	✓	✓
Driouch *et al*. [2024]	CANSat-IDS: ANN + Kernel SVM	DoS, Fuzzy injection, Replay, Impersonation	✓	Song and Kim + UAVCAN	✓	✓
Khan *et al*. [2024]	DivaCAN: DNN + LGBM + RF + ET + Bagging + kNN	DoS, Fuzzy, Impersonation	×	Lee et al. (2017) CAN dataset	×	×
Al Isawi *et al*. [2025]	ANFIS-SC, ANFIS-FCM	DoS, Impersonation	✓	Real Electric Vehicle	✓	×
Balasubramanian *et al*. [2025]	TF-IDF + DNN	Fuzzing, Spoofing, Masquerade	✓	AD, HCRL (CH and SA), ROAD	✓	✓

with neural learning through an Adaptive Neuro-Fuzzy Inference System. Clustering-based fuzzy rule generation and neural adaptation enable accurate detection of DoS and impersonation attacks while improving interpretability compared to purely deep learning–based IDS approaches [Al Isawi *et al*., 2025]. Driouch et al. proposed CANSat-IDS, a distributed hybrid intrusion detection system that combines machine learning and deep learning models within a space–ground architecture. Lightweight time-based detection is performed on board, while more complex content-based analysis is handled on the ground, enabling effective intrusion detection under strict energy and computational constraints [Driouch *et al*., 2024]. Khan et al. proposed DivaCAN, an ensemble-driven intrusion detection framework for in-vehicle CAN networks that employs a stacked design of heterogeneous base learners and a meta-level classifier. By combining diverse models to capture complementary CAN traffic characteristics, the framework improves robustness against variability in real-world vehicular environments [Khan *et al*., 2024]. Balasubramanian et al. proposed CANLP, an intrusion detection framework that applies natural language processing techniques to raw CAN frames treated as structured sequences. Operating at the per-frame level without relying on signal decoding or timing assumptions, the approach enables precise detection of stealthy CAN attacks in a vehicle-agnostic manner [Balasubramanian *et al*., 2025]. Im et al. proposed a hybrid intrusion detection architecture for in-vehicle CAN networks that integrates machine learning-based detection with lightweight rule-based cross-check filters. By validating IDS outputs using attack-specific rules derived from CAN identifiers and payload characteristics, the framework enhances robustness while remaining suitable for resource-constrained in-vehicle deployment [Im *et al*., 2024]. A more detailed quantitative comparison is provided in Table 10 and Table 11 which summarizes key aspects of each study, including the model type, attack scenarios considered, real-time deployment feasibility, datasets used, whether lightweight operation is claimed, and whether any validation beyond offline evaluation is provided. In ad- dition to the best reported evaluation metrics.

7 Discussion and Summary

A structured, deployment-oriented synthesis of recent intrusion detection systems for in-vehicle CAN networks is presented through a comparative analysis of ML, DL, and hybrid-based IDS approaches. The analysis indicates that high detection performance is widely reported in the literature; however, evaluations are often conducted under settings that do not reflect real in-vehicle operating constraints. Experimental validation typically focuses on a narrow subset of injection-based attacks, with performance primarily summarized using accuracy-oriented metrics such as precision, recall, and F1-score, and commonly assessed through offline experimentation or workstation-class hardware.

While the surveyed studies reflect current research trends and representative evaluation practices, the analysis is inherently shaped by the availability of datasets, reporting stan-

Table 11. Best reported detection metrics of Hybrid-based IDSs for CAN bus security.

Study	Best Detection Metrics (As Reported)
He *et al*. [2021]	Precision: 0.845; Recall: 0.845; AUC: 0.9216; AUPR: 0.9194
Dwivedi [2022]	Accuracy: 100%; F1-score: 1.00; AUC-ROC: 1.00
Purohit and Govindarasu [2022]	Accuracy: 97.62%; F1-score: 0.972
Xun *et al*. [2023]	Accuracy: 98.9%; Precision: 99.7%; Recall: 98.2%; F1-score: 98.9%; FPR: 0.4%; FNR: 1.8%
Nichelini *et al*. [2023]	Accuracy: 0.9997; Precision: 0.9987; TPR: 1.0; FPR: 0.0004; F1-score: 0.9936; Inference latency (TTP): 0.2568 ms
Zhang *et al*. [2023]	Accuracy: 100%; Precision: 100%; Recall (TPR): 100%; F1-score: 100%; AUC-ROC: 1.00; Latency: ≈3.0 ms
Kumar *et al*. [2024]	Accuracy: 98%; Precision: 95%; Recall: 93%; F1-score: 96%
Sun *et al*. [2024]	Accuracy: ≈99–100%; Precision: ≈99–100%; Recall: ≈99%; F1-score: ≈99%; Inference latency: 0.69 ms
Driouch *et al*. [2024]	Accuracy: 99.87%; Precision: 99.9%; Recall (TPR): 99.99%; F1-score: 99.86%
Khan *et al*. [2024]	Precision: 94.93%; Recall: 94.98%; F1-score: 94.97%; FPR: 0.21%; FNR: 0.31%
Im *et al*. [2024]	Accuracy: ≈99.99%; Precision: ≈100%; Recall: 100%; F1-score: ≈99.98%
Al Isawi *et al*. [2025]	Accuracy: 99.68%; Precision: 0.988; Recall (TPR): 0.999; TNR: 0.999; F1-score (binary): 0.993; FPR: 0.001; FNR: 0.001
Balasubramanian *et al*. [2025]	Accuracy: ≈99.7%; F1-score: 0.9974; TPR: ≈1.0; FPR: ≈0; Inference latency: 0.049 ms

dards, and the level of experimental transparency within the literature. These factors influence how IDS performance is reported and interpreted across studies.

A central observation is that lightweight and real-time claims are frequently not matched by the level of evidence provided to support them. Although many studies describe their approaches as lightweight, explicit validation, such as ECU-oriented runtime analysis, embedded deployment, or real-vehicle testing, is often absent, making direct comparison across IDS approaches challenging.

Dataset characteristics also play an important role in shaping reported IDS performance. Studies relying on simulated or narrowly scoped datasets often report higher detection accuracy, whereas evaluations using real-vehicle traffic tend to expose more realistic trade-offs between detection capability and computational feasibility.

From a system-level perspective, no single IDS paradigm satisfies all detection and deployment requirements across scenarios. Machine learning approaches typically offer lower computational overhead and greater interpretability, while DL–based methods enable richer representation learning at higher computational cost. Hybrid approaches seek to combine these strengths, with mixed effectiveness depending on design choices and deployment context.

Across the surveyed studies, a quantitative examination of the reported evaluation metrics reveals several consistent trends. Building on this, a more detailed quantitative synthesis based on the summarized detection metrics further clarifies the performance characteristics of different IDS paradigms. ML-based approaches generally report detection accuracy in the range of approximately 96% to 100%, with most studies clustering around 98–100%, although moderate variability is observed across different datasets and attack scenarios. DL-based approaches exhibit a wider performance range, from approximately 80% in certain unsupervised settings to nearly 100% in supervised configurations, with the majority of studies reporting accuracy above 98%. Hybrid approaches demonstrate similarly high performance, typically ranging from approximately 97% to 100%, with a strong concentration of results near the upper bound. This trend is further reflected in Table 12, where representative models across all categories demonstrate similarly high detection performance despite differences in latency and deployment characteristics.

Despite these high reported values, the quantitative differences between ML, DL, and hybrid approaches are relatively small, often within a narrow margin of 1–3%. This suggests that the choice of model architecture alone does not fully account for performance gains. Instead, dataset characteristics, attack injection methods, and evaluation setups appear to have a more significant impact on reported results. Furthermore, the frequent occurrence of near-perfect detection metrics across multiple studies raises concerns regarding potential dataset bias, limited attack diversity, or evaluation under controlled conditions. These observations indicate that reported performance improvements are not always statistically distinguishable in practical settings and should be interpreted with caution, particularly in the absence of standardized benchmarking and cross-dataset validation. Furthermore, latency and computational efficiency are reported less consistently across the literature, making direct comparison

difficult. While several lightweight IDS designs demonstrate sub-millisecond inference latency on embedded platforms, many studies reporting high detection accuracy do not provide detailed runtime or resource utilization analysis. These observations highlight that detection accuracy alone does not fully capture the practical effectiveness of CAN intrusion detection systems, and that evaluation realism, computational efficiency, and deployment feasibility are equally important factors when comparing IDS approaches.

To provide a consolidated quantitative view of detection performance, latency characteristics, and deployment-related evidence across representative CAN bus intrusion detection systems, Table 12 summarizes key aspects of selected machine learning, deep learning, and hybrid approaches.

8 Deployment Challenges and Practical Limitations

Despite steady progress in CAN bus intrusion detection research, moving IDS solutions from experimental studies into real vehicles remains challenging, largely due to the limited computational and memory capacity of automotive electronic ECUs. While many IDS approaches-particularly DL based models achieve strong performance in offline evaluations, they are commonly assessed on desktop-class hardware, which introduces uncertainty about whether similar performance can be sustained under the strict real-time and resource constraints of in-vehicle environments. Real-time operation presents an additional deployment challenge, as an IDS deployed on a CAN bus must process high-rate message streams with minimal latency to avoid interfering with safety-critical vehicle functions. However, many studies do not report detailed latency measurements, end-to-end timing behavior, or ECU-level execution characteristics, making it difficult to determine whether proposed approaches can operate reliably during continuous, real-world vehicle operation.

However, some studies validate CAN bus IDS using simulation-based and hardware-based experimental environments to assess deployment realism. Software-in-the-loop (SIL) validation is commonly conducted using automotive simulation platforms such as CARLA or industry-grade tools like Vector CANoe, where realistic driving scenarios and protocol-level CAN traffic are generated and attack behaviors are injected in a controlled manner to evaluate runtime stability and detection latency [Dosovitskiy *et al*., 2017; Vector Informatik, 2023]. More advanced validation adopts hardware-in-the-loop (HIL) setups, in which IDS models are deployed on embedded ECUs or low-cost automotive hardware connected to real CAN transceivers, enabling measurement of inference latency, memory usage, and real-time feasibility [Casillo *et al*., 2019; Bonomo *et al*., 2024]. A limited number of studies further employ traffic replay or partial vehicle-level experimentation based on real CAN traces to preserve realism while mitigating safety risks [Miller and Valasek, 2015]. Despite these advances, full vehicle-scale validation remains rare due to safety, legal, and proprietary constraints, highlighting a persistent gap between experimental evaluation and real-world deployment.

9 Open Research Challenges and Future Directions

While CAN bus intrusion detection has advanced considerably, several challenges remain before these systems can be reliably deployed in real vehicles. One recurring issue is the lack of standardized and realistic benchmarking. Current studies often rely on different datasets, attack configurations, and evaluation metrics, which makes it difficult to draw consistent conclusions or fairly compare proposed methods. Developing shared benchmarks that better reflect real driving conditions and diverse attack behaviors would greatly strengthen future research.

Another open challenge lies in reliability across different vehicles and conditions. Many IDS models are trained under controlled conditions and may not perform consistently when applied to different vehicle platforms or evolving traffic patterns. Future research should focus on adaptive detection mechanisms that can maintain reliable performance over time and under changing operational conditions.

Interpretability is becoming increasingly important for DL–based IDSs deployed in safety-critical automotive environments. Although complex models can achieve strong detection performance, their limited transparency reduces trust and limits practical adoption by making detection decisions difficult to understand and analyze. This lack of clarity complicates validation and error analysis and poses challenges for debugging and regulatory evaluation, highlighting the need for greater transparency when applying DL–based IDS in real-world vehicular systems.

Adversarial robustness and evasion resistance remain largely underexplored in current CAN bus intrusion detection research. Most existing IDS models are evaluated under static attack assumptions, where malicious behavior is injected without considering adaptive or model-aware adversaries. As DL–based IDS become more prevalent, understanding how these systems perform under evasion strategies, adversarial perturbations, or stealthy attack patterns designed to bypass detection is increasingly important. Future work should investigate adversarial threat models, robustness evaluation protocols, and defense mechanisms tailored to in-vehicle CAN environments to ensure reliable operation under realistic and evolving attack conditions.

Looking ahead, lightweight model optimization, federated learning, and secure update mechanisms represent promising directions for practical deployment. In addition, emerging paradigms such as foundation models, self-supervised learning, and large language models are expected to play an increasingly important role and will be considered in future work to further enhance the adaptability and intelligence of CAN intrusion detection systems. These approaches can help reduce computational overhead, protect data privacy, and enable continuous improvement across vehicle fleets. Addressing these challenges will be essential for transitioning CAN intrusion detection systems from research prototypes into dependable components of real-world automotive security systems.

Table 12. Unified quantitative comparison of representative CAN-bus IDS models.

Category	Model	Representative Study	Detection Performance (Reported)	Latency / Throughput	Deployment / Resource Evidence
ML	Decision Tree (DT)	Bonomo *et al*. [2024]	Accuracy ≈99.9%; F1-score ≈1.0	$<$ 1.2 ms per message	RISC-V embedded platform; real-time capable
ML	Random Forest (RF)	Alalwany and Mahgoub [2024]	Accuracy ≈98.5%; AUC ≈1.0	∼0.26–0.30 s (system-level)	Real CAN data; system-level evaluation
ML	One-Class SVM (OCSVM)	Avatefipour and Malik [2018]	High TPR; low FPR across datasets	∼1 ms per CAN message	Explicit real-time latency evaluation
ML	XGBoost	Xu *et al*. [2025a]	F1-score ≈99% (most attacks)	∼43.6 ms (reported latency)	Model size, MACs, and complexity reported
DL	LSTM	Hossain *et al*. [2020b]	Accuracy $>$99.9% (binary and multiclass)	Not reported	Offline evaluation (desktop GPU)
DL	GRU (lightweight RNN)	Ma *et al*. [2022]	F1-score $>$99.9%	∼650 messages / 100 ms	Jetson Xavier NX embedded deployment
DL	1D-CNN	Hossain *et al*. [2020a]	Accuracy ≈99.99%	Not reported	Offline evaluation only
DL	Autoencoder (unsupervised)	Kim *et al*. [2025]	Accuracy ≈99.2%	∼0.07 ms per inference	FPGA deployment; low LUT/FF usage
DL	CNN–LSTM	Wang *et al*. [2025]	Accuracy ≈99.8%; F1-score ≈99.7%	∼1.1 ms (embedded controller)	Embedded automotive domain controller
Hybrid	Rule-based + XGBoost	Purohit and Govindarasu [2022]	Accuracy ≈97.6%; F1-score ≈0.97	∼350 s (system-level test)	Hybrid filtering and ML evaluation
Hybrid	CANova (modular hybrid IDS)	Nichelini *et al*. [2023]	Accuracy up to 0.9997; low FPR	Real-time capable (reported)	Real CAN traffic; timing-aware design
Hybrid	Kalman filter + Naïve Bayes	Sun *et al*. [2024]	Accuracy ≈99.5–99.9%	∼0.69 ms per message	Real-vehicle deployment with latency
Hybrid	CANLP (TF–IDF + compact DNN)	Balasubramanian *et al*. [2025]	F1-score ≈0.997	∼0.049 ms per frame	Raspberry Pi 3/4 embedded deployment
Hybrid	Federated GNN + SVDD/OpenMax	Zhang *et al*. [2023]	High accuracy; open-set robustness	∼3 ms per CAN graph	Federated multi-vehicle deployment

10 Conclusion

This survey presented a structured, deployment-oriented review of learning-based intrusion detection systems for in-vehicle CAN networks, covering representative ML,DL, and hybrid approaches reported between 2019 and 2025. Rather than focusing solely on detection accuracy, the survey examined IDS designs using a broader practical perspective, considering reported evaluation realism, computational complexity, inference latency, data requirements, and the extent to which real-time or in-vehicle deployment claims are supported in the literature.

The survey also highlights the strong influence of dataset characteristics and labeling strategies on reported IDS performance. Supervised approaches relying on labeled data typically achieve high accuracy under known attack conditions but face challenges related to scalability, labeling cost, and generalization. Unsupervised and semi-supervised methods better reflect realistic in-vehicle data availability but introduce uncertainty in detection outcomes. These design considerations motivate growing interest in hybrid and semi-supervised designs that seek to balance robustness, adaptability, and practical feasibility.

By organizing existing work within a unified taxonomy

and providing consolidated comparison tables, this survey aims to clarify current evaluation practices, expose common limitations, and support more transparent and reproducible assessment of CAN bus IDSs. The identified gaps suggest that future research should place greater emphasis on realistic validation, standardized reporting of latency and resource usage, and explicit deployment evidence on embedded or vehicle-grade platforms. Addressing these challenges is essential for advancing CAN bus intrusion detection from promising experimental models toward reliable, operationally viable automotive security solutions.

Declarations

Authors' Contributions

AG and TK contributed to the conception of this study. AG collected, organized, and reviewed the relevant articles and served as the main contributor and primary writer of the manuscript. TK reviewed the work and manuscript drafts and provided technical mentorship. All authors read and approved the final manuscript.

Competing interests

The authors declare that they don't have competing interests.

Availability of data and materials

This study does not involve data or code that will be shared.

Funding

This study received no funding.

References

Abdullah, D. M. and Abdulazeez, A. M. (2021). Machine learning applications based on svm classification a review. *Qubahan Academic Journal*, 1(2):81–90. DOI: 10.48161/qaj.v1n2a50.

Aksu, D. and Aydin, M. A. (2022). Mga-ids: Optimal feature subset selection for anomaly detection framework on in-vehicle networks-can bus based on genetic algorithm and intrusion detection approach. *Computers & Security*, 118:102717. DOI: 10.1016/j.cose.2022.102717.

Al-Aql, N. and Al-Shammari, A. (2024). Hybrid rnn-lstm networks for enhanced intrusion detection in vehicle can systems. *Journal of Electrical Systems*, 33(1):1–8. DOI: 10.52783/jes.3318.

Al Isawi, O. A., Al Jaafari, K. A., and Al Sumaiti, A. S. (2025). Electric vehicles can bus cyber attacks detection using adaptive neuro fuzzy inference system. *IEEE Access*. DOI: 10.1109/access.2025.3550970.

Al-Jarrah, O. Y., Maple, C., Dianati, M., Oxtoby, D., and Mouzakitis, A. (2019). Intrusion detection systems for intra-vehicle networks: A review. *Ieee Access*, 7:21266–21289. DOI: 10.1109/access.2019.2894183.

Al Maruf, M. and Azim, A. (2024). Anomaly detection and functional testing for automotive can communication. In *2024 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE. DOI: 10.1109/syscon61195.2024.10553463.

Al-Saud, M., Eltamaly, A. M., Mohamed, M. A., and Kavousi-Fard, A. (2019). An intelligent data-driven model to secure intravehicle communications based on machine learning. *IEEE Transactions on Industrial Electronics*, 67(6):5112–5119. DOI: 10.1109/tie.2019.2924870.

Alalwany, E. and Mahgoub, I. (2024). An effective ensemble learning-based real-time intrusion detection scheme for an in-vehicle network. *Electronics*, 13(5):919. DOI: 10.3390/electronics13050919.

Alfardus, A. and Rawat, D. B. (2021). 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. DOI: 10.1109/uemcon53757.2021.9666745.

Aliwa, E., Rana, O., Perera, C., and Burnap, P. (2021). Cyberattacks and countermeasures for in-vehicle networks. *ACM computing surveys (CSUR)*, 54(1):1–37. DOI: 10.1145/3431233.

Aljabri, W., Hamid, M. A., and Mosli, R. (2025). Enhancing real-time intrusion detection system for in-vehicle networks by employing novel feature engineering techniques and lightweight modeling. *Ad Hoc Networks*, 169:103737. DOI: 10.1016/j.adhoc.2024.103737.

Alkhatib, N., Mushtaq, M., Ghauch, H., and Danger, J.-L. (2022). Can-bert do it? controller area network intrusion detection system based on bert language model. In *2022 IEEE/ACS 19th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–8. IEEE. DOI: 10.1109/aiccsa56895.2022.10017800.

Almehdhar, M., Albaseer, A., Khan, M. A., Abdallah, M., Menouar, H., Al-Kuwari, S., and Al-Fuqaha, A. (2024). Deep learning in the fast lane: A survey on advanced intrusion detection systems for intelligent vehicle networks. *IEEE Open Journal of Vehicular Technology*. DOI: 10.1109/ojvt.2024.3422253.

Althunayyan, M., Javed, A., and Rana, O. (2024). A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning. *arXiv preprint arXiv:2408.08433*. DOI: 10.1016/j.vehcom.2024.100837.

Althunayyan, M., Javed, A., and Rana, O. (2025). A survey of learning-based intrusion detection systems for in-vehicle network. *arXiv preprint arXiv:2505.11551*. DOI: 10.48550/arxiv.2505.11551.

Amato, F., Coppolino, L., Mercaldo, F., Moscato, F., Nardone, R., and Santone, A. (2021). Can-bus attack detection with deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 22(8):5081–5090. DOI: 10.1109/tits.2020.3046974.

Anjum, A., Agbaje, P., Hounsinou, S., and Olufowobi, H. (2022). In-vehicle network anomaly detection using extreme gradient boosting machine. In *2022 11th Mediterranean Conference on Embed-*

ded Computing (MECO), pages 1–6. IEEE. DOI: 10.1109/meco55406.2022.9797224.

Avatefipour, O., Al-Sumaiti, A. S., El-Sherbeeny, A. M., Awwad, E. M., Elmeligy, M. A., Mohamed, M. A., and Malik, H. (2019). An intelligent secured framework for cyberattack detection in electric vehicles' can bus using machine learning. *Ieee Access*, 7:127580–127592. DOI: 10.1109/access.2019.2937576.

Avatefipour, O. and Malik, H. (2018). State-of-the-art survey on in-vehicle network communication (can-bus) security and vulnerabilities. *arXiv preprint arXiv:1802.01725*. DOI: 10.48550/arxiv.1802.01725.

Azzarà, D. (2023). Candemonium: an evaluation test bed for objective can intrusion detection systems benchmarking. Available at:`https://www.politesi.polimi.it/retrieve/74ed111a-8d3f-4ff2-9afa-aae01af7e777/2024_12_Azzar%C3%A0_Executive_Summary.pdf`.

Balasubramanian, K., Baragur, A. G., Donadel, D., Sahabandu, D., Brighente, A., Ramasubramanian, B., Conti, M., and Poovendran, R. (2025). Canlp: Intrusion detection for controller area networks using natural language processing and embedded machine learning. *IEEE Transactions on Dependable and Secure Computing*. DOI: 10.1109/tdsc.2025.3596516.

Bari, B. S., Yelamarthi, K., and Ghafoor, S. (2023). Intrusion detection in vehicle controller area network (can) bus using machine learning: A comparative performance study. *Sensors*, 23(7):3610. DOI: 10.3390/s23073610.

Bonomo, J. P. A., Volpato, J. V., De Carvalho, R. S., and Gracioli, G. (2024). Machine learning-based intrusion detection for automotive can networks on embedded platforms. In *2024 XIV Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–6. IEEE. DOI: 10.1109/sbesc65055.2024.10771926.

Bozdal, M., Samie, M., and Jennions, I. (2018). A survey on can bus protocol: Attacks, challenges, and potential solutions. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pages 201–205. IEEE. DOI: 10.1109/iccecome.2018.8658720.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32. DOI: 10.1023/a:1010933404324.

Casillo, M., Coppolino, L., D'Antonio, S., and Mazzeo, G. (2019). Embedded intrusion detection for controller area network. *IEEE Transactions on Vehicular Technology*, 68(11):11329–11341. DOI: 10.1109/ICSRS48664.2019.8987605.

Cena, G., Seno, L., and Scanzio, S. (2025). Robust multicast origin authentication in macsec and cansec for automotive scenarios. *arXiv preprint arXiv:2502.20555*. DOI: 10.48550/arxiv.2502.20555.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., *et al*. (2015). Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4. DOI: 10.32614/cran.package.xgboost.

Chougule, A., Kulkarni, I., Alladi, T., Chamola, V., and Yu, F. R. (2024). Hybridsecnet: In-vehicle security on controller area networks through a hybrid two-step lstm-cnn model. *IEEE Transactions on Vehicular Technology*. DOI: 10.1109/tvt.2024.3413849.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297. DOI: 10.1007/bf00994018.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27. DOI: 10.1109/tit.1967.1053964.

De Araujo-Filho, P. F., Pinheiro, A. J., Kaddoum, G., Campelo, D. R., and Soares, F. L. (2021). An efficient intrusion prevention system for can: Hindering cyber-attacks with a low-cost platform. *IEEE Access*, 9:166855–166869. DOI: 10.1109/access.2021.3136147.

Desta, A. K., Ohira, S., Arai, I., and Fujikawa, K. (2022). Rec-cnn: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. *Vehicular Communications*, 35:100470. DOI: 10.1016/j.vehcom.2022.100470.

Devnath, M. K. (2023). Gcnids: Graph convolutional network-based intrusion detection system for can bus. DOI: 10.48550/arxiv.2309.10173.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR. DOI: 10.48550/arxiv.1711.03938.

Driouch, O., Bah, S., and Guennoun, Z. (2024). Cansat-ids: An adaptive distributed intrusion detection system for satellites, based on combined classification of can traffic. *Computers & Security*, 146:104033. DOI: 10.1016/j.cose.2024.104033.

Duda, R. O., Hart, P. E., *et al*. (2006). *Pattern classification*. John Wiley & Sons. DOI: 10.1007/978-1-4419-1428-$6_5$195.

Dupont, G., den Hartog, J., Etalle, S., and Lekidis, A. (2019a). 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. DOI: 10.1109/icves.2019.8906465.

Dupont, G., Lekidis, A., Den Hartog, J., and Etalle, S. (2019b). Automotive controller area network (can) bus intrusion dataset v2. *4TU. Centre for Research Data*. DOI: 10.4121/uuid:b74b4928-c377-4585-9432-2004dfa20a5d.

Dwivedi, A. K. (2022). Anomaly detection in intra-vehicle networks. *arXiv preprint arXiv:2205.03537*. DOI: 10.48550/arxiv.2205.03537.

El-Fatyany, A., Wang, X., Lu, L., and Ren, K. (2025). Ef-ids: An efficient intrusion detection system with enriched features for can bus in modern vehicles. *Journal of Systems Architecture*, page 103646. DOI: 10.1016/j.sysarc.2025.103646.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2):179–211. DOI: 10.1207/s15516709cog1402$_1$.

Fakhfakh, F., Tounsi, M., and Mosbah, M. (2022). Cybersecurity attacks on can bus based vehicles: a review and open challenges. *Library hi tech*, 40(5):1179–1203. DOI: 10.1108/lht-01-2021-0013.

Gao, K., Huang, H., Liu, L., Du, R., and Zhang, J. (2023). 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. DOI: 10.1109/ispa-bdcloud-socialcom-sustaincom59178.2023.00138.

Gazdag, A., Ferenc, R., and Buttyán, L. (2023). Crysys dataset of can traffic logs containing fabrication and masquerade attacks. *Scientific Data*, 10(1):903. DOI: 10.1038/s41597-023-02716-9.

Guezzaz, A., Benkirane, S., Azrour, M., and Khurram, S. (2021). A reliable network intrusion detection approach using decision tree with enhanced data quality. *Security and Communication Networks*, 2021:1230593. DOI: 10.1155/2021/1230593.

Gundu, R. and Maleki, M. (2022). Securing can bus in connected and autonomous vehicles using supervised machine learning approaches. In *2022 IEEE International Conference on Electro Information Technology (eIT)*, pages 042–046. IEEE. DOI: 10.1109/eit53891.2022.9813985.

Han, M. L., Kwak, B. I., and Kim, H. K. (2018). Anomaly intrusion detection method for vehicular networks based on survival analysis. *Vehicular communications*, 14:52–63. DOI: 10.1016/j.vehcom.2018.09.004.

Hanselmann, M., Strauss, T., Dormann, K., and Ulmer, H. (2020). Canet: An unsupervised intrusion detection system for high dimensional can bus data. *Ieee Access*, 8:58194–58205. DOI: 10.1109/access.2020.2982544.

He, T., Zhang, L., Kong, F., and Salekin, A. (2020). Exploring inherent sensor redundancy for automotive anomaly detection. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE. DOI: 10.1109/dac18072.2020.9218557.

He, Y., Jia, Z., Hu, M., Cui, C., Cheng, Y., and Yang, Y. (2021). The hybrid similar neighborhood robust factorization machine model for can bus intrusion detection in the in-vehicle network. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):16833–16841. DOI: 10.1109/tits.2021.3113638.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., and Kadobayashi, Y. (2020a). 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. DOI: 10.1109/globecom42002.2020.9322395.

Hossain, M. D., Inoue, H., Ochiai, H., Fall, D., and Kadobayashi, Y. (2020b). Lstm-based intrusion detection system for in-vehicle can bus communications. *Ieee Access*, 8:185489–185502. DOI: 10.1109/access.2020.3029307.

Im, H., Lee, D., and Lee, S. (2024). A novel architecture for an intrusion detection system utilizing cross-check filters for in-vehicle networks. *Sensors*, 24(9):2807. DOI: 10.3390/s24092807.

Islam, S. H., Kumar, S., Immanuel, V. J., Sonthalia, P., Mondal, S., and Pal, A. (2024). Securing connected vehicles: An evaluation framework for can bus-enabled in-vehicle communications. In *Proceedings of the 3rd International Conference on Computing Advancements*, pages 162–168. DOI: 10.1145/3723178.3723200.

Kalkan, S. C. and Sahingoz, O. K. (2020). 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. DOI: 10.1109/icccnt49239.2020.9225442.

Kang, H., Vo, T., Kim, H. K., and Hong, J. B. (2024). Canival: A multimodal approach to intrusion detection on the vehicle can bus. *Vehicular Communications*, 50:100845. DOI: 10.1016/j.vehcom.2024.100845.

Kang, T. U., Song, H. M., Jeong, S., and Kim, H. K. (2018). Automated reverse engineering and attack for can using obd-ii. In *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pages 1–7. IEEE. DOI: 10.1109/vtcfall.2018.8690781.

Karopoulos, G., Kambourakis, G., Chatzoglou, E., Hernández-Ramos, J. L., and Kouliaridis, V. (2022). Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy. *Electronics*, 11(7):1072. DOI: 10.3390/electronics11071072.

Khan, M. H., Javed, A. R., Iqbal, Z., Asim, M., and Awad, A. I. (2024). Divacan: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning. *Computers & Security*, 139:103712. DOI: 10.1016/j.cose.2024.103712.

Kheddar, H. (2025). Transformers and large language models for efficient intrusion detection systems: A comprehensive survey. *Information Fusion*, page 103347. DOI: 10.1016/j.inffus.2025.103347.

Kheddar, H., Dawoud, D. W., Awad, A. I., Himeur, Y., and Khan, M. K. (2024). Reinforcement-learning-based intrusion detection in communication networks: A review. *IEEE Communications Surveys & Tutorials*. DOI: 10.1109/comst.2024.3484491.

Kheddar, H., Himeur, Y., and Awad, A. I. (2023). Deep transfer learning applications in intrusion detection systems: A comprehensive review. *arXiv preprint arXiv:2304.10550*. DOI: 10.48550/arXiv.2304.10550.

Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22. DOI: 10.1186/s42400-019-0038-7.

Kim, D., Im, H., and Lee, S. (2025). Adaptive autoencoder-based intrusion detection system with single threshold for can networks. *Sensors*, 25(13):4174. DOI: 10.3390/s25134174.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., *et al.* (2010). Experimental security analysis of a modern automobile. In *2010 IEEE symposium on security and privacy*, pages 447–462. IEEE. DOI: 10.1109/sp.2010.34.

Kumar, S. V., Singh, B., Batra, R., and Bhagat, A. K. (2024). An ai-powered security system for can bus attacks identification in electric automobiles. *Proceedings on Engineering*, 6(1):221–230. DOI: 10.24874/pes.si.24.02.005.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324. DOI: 10.1109/5.726791.

Lee, H., Jeong, S. H., and Kim, H. K. (2017). 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. DOI: 10.1109/pst.2017.00017.

Levy, E., Shabtai, A., Groza, B., Murvay, P.-S., and Elovici, Y. (2023). Can-loc: Spoofing detection and physical intrusion localization on an in-vehicle can bus based on deep features of voltage signals. *IEEE Transactions on Information Forensics and Security*. DOI: 10.1109/tifs.2023.3297444.

Li, J., Ersotelos, N., Bottarelli, M., and Epiphaniou, G. (2023). Evaluation of machine learning and deep learning-based intrusion detection systems in in-vehicle networks. In *International Conference on AI and the Digital Economy (CADE 2023)*, volume 2023, pages 103–108. IET. DOI: 10.1049/icp.2023.2579.

Li, X., Zhang, H., Miao, Y., Ma, S., Ma, J., Liu, X., and Choo, K.-K. R. (2021). Can bus messages abnormal detection using improved svdd in internet of vehicles. *IEEE Internet of Things Journal*, 9(5):3359–3371. DOI: 10.1109/jiot.2021.3098221.

Lin, H.-C., Wang, P., Chao, K.-M., Lin, W.-H., and Chen, J.-H. (2022). Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks. *Electronics*, 11(14):2180. DOI: 10.3390/electronics11142180.

Lokman, S.-F., Othman, A. T., and Abu-Bakar, M.-H. (2019). Intrusion detection system for automotive controller area network (can) bus system: a review. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):1–17. DOI: 10.1186/s13638-019-1484-3.

Lotto, A., Marchiori, F., Brighente, A., and Conti, M. (2024a). A survey and comparative analysis of security properties of can authentication protocols. *IEEE Communications Surveys & Tutorials*. DOI: 10.1109/comst.2024.3486367.

Lotto, A., Marchiori, F., Brighente, A., and Conti, M. (2024b). A survey and comparative analysis of security properties of can authentication protocols. *arXiv preprint arXiv:2401.10736*. DOI: 10.1109/comst.2024.3486367.

Ma, H., Cao, J., Mi, B., Huang, D., Liu, Y., and Li, S. (2022). A gru-based lightweight system for can intrusion detection in real time. *Security and Communication Networks*, 2022(1):5827056. DOI: 10.1155/2022/5827056.

Mehedi, S. T., Anwar, A., Rahman, Z., and Ahmed, K. (2021). Deep transfer learning based intrusion detection system for electric vehicular networks. *Sensors*, 21(14):4736. DOI: 10.3390/s21144736.

Miller, C. and Valasek, C. (2015). Remote exploitation of an unaltered passenger vehicle. *Black Hat USA*, 2015(S 91):1–91. Available at:`https://www.ioactive.com/wp-content/uploads/pdfs/IOActive_Remote_Car_Hacking.pdf`.

Moulahi, T., Zidi, S., Alabdulatif, A., and Atiquzzaman, M. (2021). Comparative performance evaluation of intrusion detection based on machine learning in in-vehicle controller area network bus. *IEEE Access*, 9:99595–99605. DOI: 10.1109/access.2021.3095962.

Nagarajan, J., Mansourian, P., Shahid, M. A., Jaekel, A., Saini, I., Zhang, N., and Kneppers, M. (2023). Machine learning based intrusion detection systems for connected autonomous vehicles: A survey. *Peer-to-Peer Networking and Applications*, 16(5):2153–2185. DOI: 10.1007/s12083-023-01508-7.

Narasimhan, H., Ravi, V., and Mohammad, N. (2021). Unsupervised deep learning approach for in-vehicle intrusion detection system. *IEEE Consumer Electronics Magazine*, 12(1):103–108. DOI: 10.1109/mce.2021.3116923.

National Highway Traffic Safety Administration (2020). Cybersecurity Best Practices for the Safety of Modern Vehicles. Technical report, U.S. Department of Transportation. Available at:`https://www.nhtsa.gov/technology-innovation/vehicle-cybersecurity`. Accessed: 2025-01-07.

Nazakat, I. and Khurshid, K. (2019). Intrusion detection system for in-vehicular communication. In *2019 15th International Conference on Emerging Technologies (ICET)*, pages 1–6. IEEE. DOI: 10.1109/icet48972.2019.8994327.

Nichelini, A., Pozzoli, C. A., Longari, S., Carminati, M., and Zanero, S. (2023). Canova: a hybrid intrusion detection framework based on automatic signal classification for can. *Computers & Security*, 128:103166. DOI: 10.1016/j.cose.2023.103166.

Palanca, A., Evenchick, E., Maggi, F., and Zanero, S. (2017). A stealth, selective, link-layer denial-of-service attack against automotive networks. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings 14*, pages 185–206. Springer. DOI: 10.1007/978-3-319-60876-1_9.

Palma, Á., Antunes, M., Bernardino, J., and Alves, A. (2025). Multi-class intrusion detection in internet of vehicles: Optimizing machine learning models on imbalanced data. *Future Internet*, 17(4):162. DOI: 10.3390/fi17040162.

Presi, T. (2013). Design and development of pic microcontroller based vehicle monitoring system using controller area network (can) protocol. In *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1070–1076. IEEE. DOI: 10.1109/icices.2013.6508232.

Purohit, S. and Govindarasu, M. (2022). Ml-based anomaly detection for intra-vehicular can-bus networks. In *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 233–238. IEEE. DOI: 10.1109/csr54599.2022.9850292.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106. DOI: 10.1023/a:1022643204877.

Rai, R., Grover, J., Sharma, P., and Pareek, A. (2025). Securing the can bus using deep learning for intrusion detection in vehicles. *Scientific Reports*, 15(1):13820. DOI: 10.1038/s41598-025-98433-x.

Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Madzudzo, G., and Petrovski, A. V. (2022). 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. DOI: 10.23919/cycon55549.2022.9811048.

Rajapaksha, S., Kalutarage, H., Al-Kadri, M. O., Petrovski, A., Madzudzo, G., and Cheah, M. (2023). Ai-based intrusion detection systems for in-vehicle networks: A survey. *ACM Computing Surveys*, 56(2). DOI: 10.1145/3570954.

Rangsikunpum, A., Amiri, S., and Ost, L. (2024). 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. DOI: 10.1109/icit58233.2024.10540960.

Rendel, J., Balte, W., Kurunathan, H., Ali, H. I., Roque, A. D. S., De Morais, W. O., and Fazeli, M. (2025). Zero-can: Anomaly-based zero-day attack detection in vehicular can bus networks. In *2025 33rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pages 121–128. IEEE. DOI: 10.1109/pdp66500.2025.00025.

Samir, S. B. H., Raissa, M., Touati, H., Hadded, M., and Ghazzai, H. (2024). Machine learning-based intrusion detection for securing in-vehicle can bus communication. *SN Computer Science*, 5(8):1082. DOI: 10.1007/s42979-024-03465-1.

Seo, E., Song, H. M., and Kim, H. K. (2018). Gids: Gan based intrusion detection system for in-vehicle network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6. DOI: 10.1109/pst.2018.8514157.

Sharmin, S. and Mansor, H. (2021). Intrusion detection on the in-vehicle network using machine learning. In *2021 3rd International Cyber Resilience Conference (CRC)*, pages 1–6. IEEE. DOI: 10.1109/crc50527.2021.9392627.

Sharmin, S., Mansor, H., Abdul Kadir, A. F., and Aziz, N. A. (2024). Benchmarking frameworks and comparative studies of controller area network (can) intrusion detection systems: A review. *Journal of Computer Security*, 32(5):477–507. DOI: 10.3233/jcs-230027.

Shi, J., Xie, Z., Dong, L., Jiang, X., and Jin, X. (2024). Idsdec: A novel intrusion detection for can bus traffic based on deep embedded clustering. *Vehicular Communications*, 49:100830. DOI: 10.1016/j.vehcom.2024.100830.

Siddiqui, H., Kibriya, H., Khan, W. Z., Siddiqui, S. T., Rana, N., and Tahir, A. (2025). Detecting can bus attacks in autonomous vehicles using deep learning. In *2025 International Conference on Emerging Technologies in Electronics, Computing, and Communication (ICETECC)*, pages 1–6. IEEE. DOI: 10.1109/icetecc65365.2025.11070252.

Song, H. M. and Kim, H. K. (2021). Discovering can specification using on-board diagnostics. *IEEE Design Test*, 38(3):93–103. DOI: 10.1109/MDAT.2020.3011036.

Song, H. M., Woo, J., and Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural network. *Vehicular Communications*, 21:100198. DOI: 10.1016/j.vehcom.2019.100198.

Specification, C. (1991). Bosch. *Robert Bosch GmbH, Postfach*, 50:15. DOI: 10.1365/s40112-014-0662-2.

Sun, H., Huang, W., Weng, J., Liu, Z., Tan, W., He, Z., Chen, M., Wu, B., Li, L., and Peng, X. (2024). Ccid-can: Cross-chain intrusion detection on can bus for autonomous vehicles. *IEEE Internet of Things Journal*, 11(15):26146–26159. DOI: 10.1109/jiot.2024.3393122.

Talebi, S. (2015). A security evaluation and internal penetration testing of the can-bus. Available at:`https://publications.lib.chalmers.se/records/fulltext/212488/212488.pdf`.

Tanksale, V. (2019). Intrusion detection for controller area network using support vector machines. In *2019 IEEE 16th international conference on mobile ad hoc and sensor systems workshops (MASSW)*, pages 121–126. IEEE. DOI: 10.1109/massw.2019.00032.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30. DOI: 10.65215/ctdc8e75.

Vector Informatik (2023). Canoe: Test and simulation tool for automotive networks. Available at:`https://www.vector.com/int/en/products/products-a-z/software/canoe/`. Accessed 2026.

Walid, M. A. A., Devnath, M. K., Muiz, B., Melon, M. M. H., Barman, P. K., and Barman, P. K. (2024). Efficient graph-based intrusion detection for can bus security with minimal training data. In *2024 6th International Conference on Sustainable Technologies for Industry 5.0 (STI)*, pages 1–6. IEEE. DOI: 10.1109/sti64222.2024.10951138.

Wang, Q., Lu, Z., and Qu, G. (2018). An entropy analysis based intrusion detection system for controller area network in vehicles. *arXiv preprint arXiv:1808.04046*. DOI: 10.1109/socc.2018.8618564.

Wang, X., Zhao, J., Liu, P., Yao, N., and Xu, Z. (2025). Can bus intrusion detection based on deep learning with data augmentation for connected autonomous vehicles. *IEEE Transactions on Vehicular Technology*. DOI: 10.1109/tvt.2025.3603056.

Wei, P., Wang, B., Dai, X., Li, L., and He, F. (2023). A novel intrusion detection model for the can bus packet of in-vehicle network based on attention mechanism and autoencoder. *Digital Communications and Networks*, 9(1):14–21. DOI: 10.1016/j.dcan.2022.04.021.

Wu, W., Li, R., Xie, G., An, J., Bai, Y., Zhou, J., and Li, K. (2019). A survey of intrusion detection for in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):919–933. DOI: 10.1109/tits.2019.2908074.

Wu, Y. and Tao, X. (2024). 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. DOI: 10.1109/mlise62164.2024.10674157.

Xin, Y., Wang, X., Lu, L., Zhuo, S., Jiang, Y., Singh, A. K., Ren, K., Yang, M., and Wu, K. (2025). Luft-can: A lightweight unsupervised learning based intrusion detection system with frequency-time analysis for vehicular can bus. *Journal of Systems Architecture*, page 103567. DOI: 10.1016/j.sysarc.2025.103567.

Xu, B., Cao, F., Li, X., Tian, S., Deng, W., and Yue, S. (2025a). Bepcd: an ensemble learning-based intrusion detection framework for in-vehicle can bus. *PeerJ Computer Science*, 11:e3108. DOI: 10.7717/peerj-cs.3108.

Xu, H., Shi, X., Liu, H., Wang, Y., Lu, J., Zeng, H., Li, R., and Wu, D. (2025b). Mulsam: Multidimensional attention

with hardware acceleration for efficient intrusion detection on vehicular can bus. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. DOI: 10.1109/tcad.2025.3541566.

Xun, Y., Deng, Z., Liu, J., and Zhao, Y. (2023). Side channel analysis: A novel intrusion detection system based on vehicle voltage signals. *IEEE Transactions on Vehicular Technology*, 72(6):7240–7250. DOI: 10.1109/tvt.2023.3236820.

Xun, Y., Zhao, Y., and Liu, J. (2021). Vehicleeids: A novel external intrusion detection system based on vehicle voltage signals. *IEEE Internet of Things Journal*, 9(3):2124–2133. DOI: 10.1109/jiot.2021.3090397.

Yang, L., Moubayed, A., Hamieh, I., and Shami, A. (2019). Tree-based intelligent intrusion detection system in internet of vehicles. In *2019 IEEE global communications conference (GLOBECOM)*, pages 1–6. IEEE. DOI: 10.1109/globecom38437.2019.9013892.

Young, C., Zambreno, J., Olufowobi, H., and Bloom, G. (2019). Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test*, 36(6):48–55. DOI: 10.1109/mdat.2019.2899062.

Zhang, H., Zeng, K., and Lin, S. (2023). Federated graph neural network for fast anomaly detection in controller area networks. *IEEE Transactions on Information Forensics and Security*, 18:1566–1579. DOI: 10.1109/tifs.2023.3240291.

Zhang, L., Yan, X., and Ma, D. (2024). 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. DOI: 10.1109/infocom52122.2024.10621400.

Zhao, H., Wang, Y., Cheng, A., Wang, S., Yuan, J., and Wang, H. (2024). Safeguarding gru-based intrusion detection systems from adversarial attacks with dynamic label watermark in can bus communication. *IEEE Internet of Things Journal*. DOI: 10.1109/jiot.2024.3524504.

Zhou, J., Li, S., Cao, Y., Hadi, H. J., and Lin, H. (2024). Robust intrusion detection system in can bus through multi-scale feature fusion. In *ICC 2024-IEEE International Conference on Communications*, pages 1316–1321. IEEE. DOI: 10.1109/icc51166.2024.10623113.

Zhou, Y. and Wang, Y. (2018). Effective intrusion detection system using xgboost. *Information*, 9(7):149. DOI: 10.3390/info9070149.