vehicles

MDPI

Systematic Review

AI-Based Intrusion Detection Beyond CAN: A Systematic Review of Open-Set, Drift-Aware, and Multi-Protocol Security in Software-Defined Vehicles

Mikhail Smolin

Department of Applied Mathematics and Computer Science, Kyrgyz State Technical University Named After I. Razzakov, Bishkek 720044, Kyrgyz Republic; mikhail_smolin@kstu.kg

Abstract

Background: Software-defined and data-driven in-vehicle and connected-vehicle communication systems are growing in complexity and connectivity, thereby increasing the attack surface. This systematic review examines which artificial intelligence (AI)-based intrusion detection system (IDS) approaches have been proposed for automotive networks and how mature the supporting evidence is with respect to unknown attacks, multi-protocol settings, and deployment. **Methods:** IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Wiley Online Library, Scopus, Web of Science, and Google Scholar were searched for studies published between January 2008 and July 2026 (last search: 6 July 2026), complemented by backward and forward snowballing. Primary studies that proposed or evaluated an automotive IDS, introduced an automotive IDS dataset or benchmark, or evaluated attacks against an automotive IDS were eligible; surveys, methodological guidelines, standards, and non-automotive studies were excluded. Evaluation practice was examined across eight dimensions, and the evidence was synthesized narratively and with descriptive statistics. **Results:** Forty-four primary studies published between 2011 and 2025 were included: 36 detectors, four datasets or benchmarks, and four robustness evaluations. Thirty-nine studies (88.6%) addressed CAN only; the five studies covering Automotive Ethernet, SOME/IP, hybrid CAN–Ethernet, or connected-vehicle settings were all published from 2022 onward. Foundational entropy-, timing-, and physical-characteristic-based methods accounted for 58.3% of detectors published up to 2019 but 15.4% of those published in 2023–2025, whereas generative, multi-stage or federated, and adaptive designs rose from 8.3% to 31.6% of all studies. **Conclusions:** The evidence remains concentrated on CAN and on closed-set, offline evaluation, and the heterogeneity of datasets and validation protocols precluded meta-analysis. Practical automotive IDS development should move beyond isolated accuracy reporting toward reproducible, robust, deployment-aware, and lifecycle-aligned evaluation.

Keywords: automotive cybersecurity; intrusion detection system; CAN bus; Automotive Ethernet; Internet of Vehicles; machine learning; deep learning; generative AI; software-defined vehicles; systematic review

Check for updates

Academic Editors: Yong Liao, Chunguo Li and Pei Liu

Received: 9 September 2026
Revised: 8 November 2026
Accepted: 23 November 2026
Published: 26 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Modern vehicles are evolving from largely mechanical transportation platforms into connected, software-defined, and data-driven cyber-physical systems. Contemporary vehicles integrate dozens of electronic control units (ECUs), sensors, actuators, gateways,

communication buses, infotainment units, telematics interfaces, over-the-air update mechanisms, and vehicle-to-everything (V2X) connectivity, enabling advanced driver assistance, cooperative mobility, predictive maintenance, remote diagnostics, and personalized services [1]. However, this connectivity also expands the automotive attack surface by creating additional internal communication paths and external entry points through which malicious actors may influence safety-critical functions [2,3].

Automotive cybersecurity is particularly challenging because in-vehicle communication is heterogeneous, distributed, and resource-constrained. Although the Controller Area Network (CAN) remains one of the most widely studied in-vehicle protocols, modern architectures increasingly combine CAN, CAN FD, Local Interconnect Network (LIN), FlexRay, Media Oriented Systems Transport (MOST), Automotive Ethernet, Scalable Service-Oriented Middleware over IP (SOME/IP), gateways, domain controllers, and zonal or service-oriented architectures [4,5]. Consequently, automotive intrusion detection should not be reduced to the absence of native authentication and encryption in CAN [6]. It should instead address malicious, abnormal, or safety-relevant behavior across distributed, real-time, multi-protocol, and energy-constrained vehicular environments [4,7].

Intrusion Detection Systems (IDSs) have become an important defensive layer because preventive mechanisms alone cannot eliminate all runtime threats. Cryptographic protection, secure boot, access control, and secure software updates are essential, but they may not detect compromised ECUs, injected or replayed messages, abnormal timing, fuzzing, spoofing, denial-of-service (DoS), diagnostic abuse, SOME/IP manipulation, or evolving attack patterns [5,8]. IDSs complement these controls by monitoring vehicle communication and identifying deviations from expected behavior, although they must operate under strict latency, computational, memory, energy, and safety constraints. A high false-positive rate may overload monitoring systems or trigger unnecessary responses, whereas false negatives may allow attacks to persist in systems controlling safety-critical functions [9,10].

Artificial intelligence (AI), machine learning (ML), and deep learning (DL) have consequently gained substantial attention in automotive IDS research. Although traditional rule-based and signature-based IDSs are effective in detecting known attacks, they are inadequate for detecting unknown, evolving, or distribution-shifted attacks [11]. ML models, such as support vector machines, decision trees, random forests, *k*-nearest neighbors, and ensemble models, can learn statistical traffic patterns from engineered features, whereas DL models, including convolutional networks, recurrent networks, long short-term memory networks, gated recurrent units, autoencoders, attention-based models, and hybrid architectures, can learn temporal, payload-level, and sequence-level representations [12–14]. More recent work has examined generative and adaptive approaches, such as generative adversarial networks (GANs), variational autoencoders (VAEs), training-data synthesis, drift-aware retraining, and multi-stage pipelines, for learning from rare attacks, detecting unknown attacks, and controlling false positives [15].

Despite this progress, automotive IDS research is still fragmented. Studies frequently report high accuracy for a single dataset, attack family, or protocol, but inconsistent datasets, attack definitions, feature-extraction pipelines, train–test splits, evaluation metrics, and implementation availability hinder comparison [16,17]. Widely used datasets may also exhibit limited realism, narrow attack diversity, insufficient cross-vehicle validation, or attack-injection procedures that do not fully represent real driving conditions [18,19]. Moreover, many studies emphasize accuracy while providing limited evidence concerning false-positive and false-negative rates, detection latency, resource consumption, adversarial robustness, interpretability, and deployment feasibility, thereby weakening the transferability of laboratory results to production vehicles [10,20].

Existing reviews have examined in-vehicle and CAN-based IDSs, DL-based detection, automotive cyberattacks, and countermeasures for vehicular networks [7,14]. However, few reviews jointly examine fast known-attack detection, unknown or evolving attacks, synthetic-data-supported learning, false-positive control, adversarial robustness, dataset maturity, reproducibility, and deployment readiness under automotive constraints, and few quantify how often the primary studies actually report such evidence.

This review addresses this gap through an evidence-based synthesis of AI-based automotive IDS research. It covers CAN, CAN FD, SOME/IP, Automotive Ethernet, LIN, FlexRay, in-vehicle networks (IVNs), the Internet of Vehicles (IoV), V2X, connected and autonomous vehicles, and software-defined vehicle environments [21,22]. Although the primary emphasis is on AI-based IDSs, foundational non-learning approaches, including entropy-, timing-, frequency-, clock-, voltage-, and physical-characteristic-based methods, are considered where needed for taxonomy development, benchmark positioning, and comparison [23,24].

The objective is not only to identify models with high detection performance but also to determine whether existing approaches are sufficiently mature for realistic automotive deployment. This requires examining known and unknown attack detection, adaptive retraining, generative augmentation, adversarial robustness, dataset realism, reproducibility, latency, memory, CPU and energy requirements, and ECU or gateway feasibility within the lifecycle risk-management and regulatory contexts established by ISO/SAE 21434 and UN Regulation No. 155 [25,26].

Table 1 summarizes the principal research gaps identified in the existing literature and how this review addresses them.

Table 1. Research gaps and scope of this systematic review.

Review Dimension	Common Limitation in Existing Literature	Focus of This SLR
Network scope	Many studies emphasize CAN or isolated in-vehicle traffic settings.	Covers CAN, CAN FD, SOME/IP, Automotive Ethernet, LIN, FlexRay, IVN, IoV, V2X, SDV, and hybrid architectures, and measures how often each is studied.
Attack handling	Known attacks are often evaluated more thoroughly than unknown, shifted, or evolving attacks.	Separates known-attack detection, unknown-attack detection, evolving attacks, and attacks against IDS models.
AI methodology	Prior work often compares individual ML or DL models without connecting adaptive, generative, and multi-stage designs.	Synthesizes traditional ML, DL, generative, adaptive, multi-stage, federated, adversarially robust, and lightweight IDS approaches.
Dataset maturity	Dataset realism, attack diversity, cross-dataset validation, and reproducibility are inconsistently reported.	Assesses real, synthetic, hybrid, simulated, and benchmark datasets, including their coverage and benchmarking limitations.
Deployment readiness	Accuracy is frequently emphasized over false positives, latency, memory, energy, and ECU feasibility.	Evaluates IDS approaches through detection performance, false-positive control, resource limits, real-time feasibility, fleet-level operation, and lifecycle relevance.

The contributions of this review are fivefold. First, it provides a systematic review protocol for AI-based automotive IDSs, including search strategy, eligibility criteria, data extraction, evidence synthesis, and threats-to-validity analysis. Second, it develops a taxonomy that connects foundational automotive IDS categories with ML, DL, generative, adaptive, multi-stage, adversarially robust, and deployable IDS approaches, and it quantifies how the distribution of these families has changed over time. Third, it analyzes the

dataset and benchmarking ecosystem, including real versus synthetic data, attack coverage, feature types, evaluation metrics, reproducibility, and cross-dataset validation. Fourth, it adds a deployment-aware assessment view that takes into account detection time, false positives, memory, CPU and energy constraints, ECU and gateway placement, fleet-level federated operation, computational offloading, and the retraining lifecycle. Fifth, it uncovers research gaps and translates them into concrete steps and a minimum reporting checklist for adaptive, generative, robust, benchmarked, interpretable, and deployable IDS solutions in real automotive settings.

Research Questions

To ensure a systematic and traceable synthesis, this SLR is guided by the following research questions:

RQ1: What AI-based intrusion-detection approaches have been proposed for automotive cybersecurity?
RQ2: Which approaches demonstrate suitability for the rapid detection of known automotive attacks?
RQ3: How do automotive IDS approaches address unknown, evolving, or distribution-shifted attack patterns?
RQ4: Which datasets, input features, attack classes, and evaluation metrics are used to assess automotive IDS models?
RQ5: How do single-stage, multi-stage, adaptive, generative, and adversarially robust IDS approaches differ in their capabilities and limitations?
RQ6: What are the principal reproducibility, benchmarking, and cross-dataset evaluation limitations in existing automotive IDS research?
RQ7: What technical and operational barriers limit the deployment and industrial adoption of AI-based automotive IDSs?
RQ8: To what extent do existing approaches address interpretability, robustness, and trustworthiness in deployment-relevant automotive environments?

These questions connect the study-selection and data-extraction process to the subsequent descriptive, taxonomic, comparative, and deployment-oriented analyses. Appendix B, Table A4, maps each question to the extracted evidence and the synthesis method.

The remainder of this paper is organized as follows. Section 2 presents the background and scope of automotive communication environments, threat models, IDS categories, and known versus unknown attack challenges. Section 3 reviews related surveys and positions this SLR against prior work. Section 4 describes the systematic review methodology, including search strategy, eligibility, screening, data extraction, and synthesis methods. Section 5 presents the descriptive analysis of the primary studies. Section 6 develops the taxonomy of automotive IDS approaches with emphasis on AI-based methods. Section 7 analyzes attack types and detection targets. Section 8 reviews datasets, synthetic data, and evaluation practices. Section 9 compares IDS approaches across performance, robustness, reproducibility, adaptability, and deployment readiness, including fleet-level, offloaded, and adaptive deployment settings. Section 10 presents the open challenges, the future research agenda, and concrete steps to strengthen the evidence base, and Section 11 concludes the paper.

2. Background and Scope

This section defines the technical background and scope of the review. Automotive intrusion detection systems (IDSs) operate in a constrained cyber-physical environment where communication behavior, timing, safety, and resource limits are as important as detection accuracy. Unlike conventional enterprise networks, vehicles contain distributed electronic control units (ECUs), real-time control loops, diagnostic interfaces, gateways, and external connectivity services. Therefore, an automotive IDS cannot be evaluated only as a generic network-security problem. It must be understood in relation to vehicle communication protocols, attack surfaces, deployment locations, and safety-critical constraints.

2.1. In-Vehicle Communication Environment

Modern vehicles employ heterogeneous communication architectures consisting of CAN, CAN FD, LIN, FlexRay, Automotive Ethernet, SOME/IP, gateways, domain controllers, and increasingly software-defined and service-oriented vehicle platforms [27–29]. While CAN remains the most studied protocol because of its widespread deployment and the availability of public datasets, modern IDS solutions must address hybrid network environments spanning message-based, service-oriented, and externally connected communication domains. Consequently, intrusion detection must account for heterogeneous protocols, real-time constraints, distributed ECUs, and evolving attack surfaces across both internal and external vehicle interfaces.

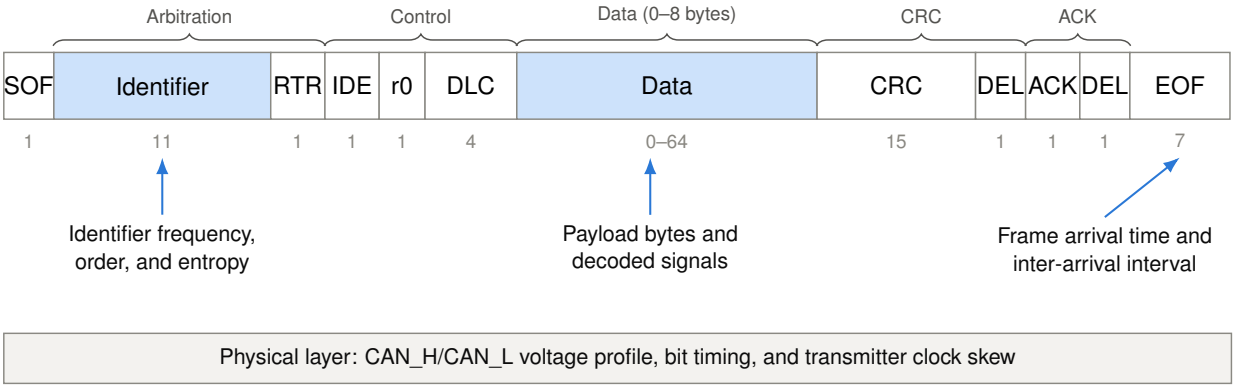
CAN deserves particular attention because it remains the reference case for automotive IDS research. A classical CAN data frame (Figure 1a) carries an 11-bit identifier that determines both the meaning of a message and its arbitration priority, a data length code, up to eight data bytes (up to 64 bytes in CAN FD), and a cyclic redundancy check, but it carries no source address and no authentication field [6,27]. Any node attached to the bus can therefore transmit any identifier, and receivers cannot verify which ECU produced a frame. CAN IDSs compensate for this by inferring misbehavior from the evidence that the bus does expose: identifier statistics such as frequency, order, and entropy [30–32]; payload bytes and decoded signal values [33–35]; frame timing [22,36]; and physical-layer characteristics such as voltage profiles, bit timing, and transmitter clock skew [21,23,37].

In software-defined vehicles (Figure 1b), CAN segments coexist with CAN FD, LIN, and Automotive Ethernet carrying SOME/IP services, and they are interconnected through a central gateway that also links external interfaces such as the telematics unit and the OBD-II diagnostic port [4,29]. The evidence available to an IDS therefore depends on where it is placed. An ECU-level monitor observes one segment with low latency, a gateway-level IDS observes several segments and protocols but may not see the physical source of a frame, and an edge or cloud service can correlate events across a fleet at the cost of connectivity and privacy constraints. This placement view is used throughout the review, particularly in Sections 7.2 and 9.

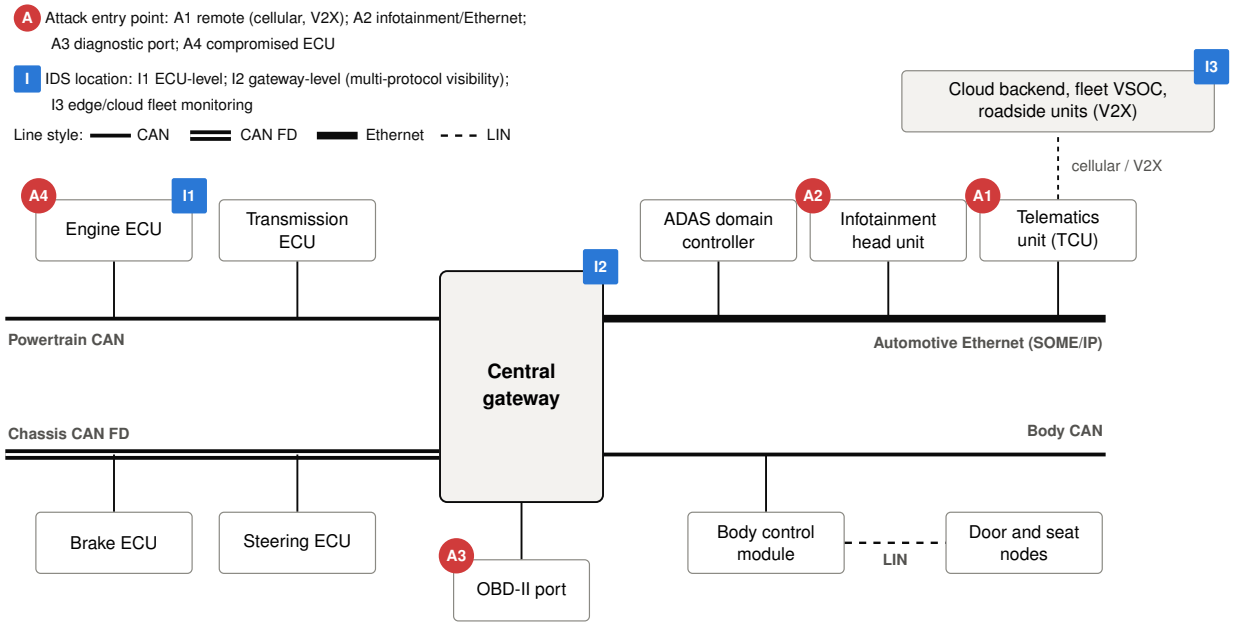
2.2. Automotive Threat Landscape and IDS Categories

Automotive IDS research addresses a broad spectrum of attacks, including DoS, flooding, spoofing, replay, injection, fuzzing, impersonation, diagnostic abuse, and service misuse [38]. These attacks may affect message timing, payloads, ECU identities, service interactions, or higher-level vehicle functions. Automotive IDS approaches are commonly categorized as signature-based, anomaly-based, and timing-based. While traditional approaches focus on predefined rules or communication characteristics, earlier classification-oriented work also applied machine learning to intrusion detection in vehicle systems [39].

A key distinction in this review is between known attacks, which are represented during training or explicitly modeled by the IDS, and unknown attacks, which involve unseen, modified, evolving, or distribution-shifted behaviors. Since real-world attackers continuously adapt their techniques, the ability to detect unknown attacks is increasingly viewed as a critical indicator of deployment readiness.



(**a**) Classical CAN base data frame and the detection evidence extracted from it.



(**b**) Hybrid in-vehicle architecture with attack entry points and IDS deployment locations.

Figure 1. CAN as the reference case for automotive intrusion detection. (**a**) The base data frame carries no sender identity or authentication, so CAN IDSs infer misbehavior from identifier statistics, payload or signal content, timing, and physical-layer characteristics [6,21–23,27,30–37]. (**b**) In a software-defined vehicle, CAN segments sit beside CAN FD, LIN, and Automotive Ethernet behind a central gateway; the observable evidence depends on where the IDS is placed [1–3,24,29,40,41]. Reference numbers indicate representative primary studies and background sources.

2.3. Scope Boundaries of This Review

This review focuses on AI-based IDS approaches for in-vehicle, connected-vehicle, and software-defined vehicle environments. Included studies address CAN, CAN FD, LIN, FlexRay, Automotive Ethernet, SOME/IP, IVN, IoV, V2X, autonomous vehicles, and hybrid communication architectures. The review covers supervised, unsupervised,

semi-supervised, self-supervised, generative, adaptive, federated, adversarially robust, and lightweight IDS approaches, while excluding non-automotive IDS research and security mechanisms unrelated to intrusion detection. Particular emphasis is placed on dataset maturity, robustness, reproducibility, deployment feasibility, and unknown-attack readiness.

The review distinguishes primary studies from background sources. Primary studies are those that propose or evaluate an automotive IDS, introduce or benchmark a dataset for automotive IDS evaluation, or evaluate attacks against an automotive IDS. Surveys, methodological guidelines, standards and specifications, generic machine-learning or security works, and automotive attack or authentication studies without a detection contribution are cited for background, methodology, and positioning, but they are not counted as evidence in the synthesis (Section 4.2).

3. Related Work and Positioning

Prior reviews of automotive intrusion detection can be grouped into six clusters. First, CAN and in-vehicle-network IDS reviews catalogue detection approaches for classical buses and discuss their constraints in vehicles [5,6,8,42]. Second, AI- and deep-learning-focused surveys organize learning-based detectors by architecture, learning paradigm, and evaluation setting [10,14,43]. Third, taxonomies and surveys of surveys consolidate earlier classifications of attacks, defences, and detection mechanisms into broader meta-taxonomies, including cyber-physical detection for vehicles [4,7,9]. Fourth, dataset and synthetic-data reviews examine benchmark coverage, realism, and the use of generated data for training and testing [16,18]. Fifth, connected-vehicle, IoV, and misbehavior-detection reviews address V2X, VANET, cooperative intelligent transportation systems, networked autonomous systems, and generative AI for IoV security [44–47]. Sixth, protocol-security surveys cover in-vehicle protocols and Automotive Ethernet vulnerabilities [29].

Collectively, these studies provide valuable insights into attack types, detection methods, datasets, and evaluation practices. However, most reviews focus on specific protocols, model classes, datasets, or application domains, and they describe evaluation weaknesses qualitatively rather than measuring how often the primary studies report unknown-attack testing, cross-dataset validation, deployment evidence, or multi-protocol coverage. Table 2 summarizes the primary emphasis of each review and the complementary emphasis of the present SLR.

The present review differs from these works in three respects. First, it codes each primary study by contribution type, IDS family, network environment, and publication period, and it reports the resulting distributions, so that statements about trends and coverage rest on measured shares rather than on qualitative impressions (Section 5). Second, it treats classical buses, Automotive Ethernet and SOME/IP, and V2X/IoV within one coded evidence base, which makes the concentration of evidence on CAN visible and comparable across protocols. Third, it relates the evidence to lifecycle requirements (ISO/SAE 21434, ISO 24089, and UN Regulations No. 155 and No. 156 [25]) and to the federated, offloaded, and adaptive deployment settings in which automotive IDSs will operate (Section 9).

Overall, previous reviews provide strong foundations but remain fragmented across CAN security, AI methods, datasets, synthetic data, and connected-vehicle security. This SLR addresses these limitations through a systematic synthesis of 44 primary studies spanning IDS families, attack types, datasets, robustness, reproducibility, trustworthiness, and deployment feasibility.

Table 2. Comparison with representative automotive IDS reviews.

Review	Year	Primary Emphasis	Complementary Emphasis of This Review
CAN and in-vehicle-network IDS reviews			
Al-Jarrah et al. [6]	2019	Intra-vehicle IDS review	Hybrid networks, SOME/IP, generative methods, and deployment-aware AI evaluation.
Lokman et al. [8]	2019	CAN-bus IDS review	Synthetic data, unknown attacks, adaptive learning, adversarial robustness, and deployment feasibility.
Wu et al. [5]	2020	Survey of IDS for in-vehicle networks	Learning-based families introduced after 2020 and their evaluation evidence.
Khan et al. [42]	2025	IVN IDS protocols, applications, and challenges	Study-level coding of evaluation practice and lifecycle alignment.
AI- and deep-learning-focused surveys			
Rajapaksha et al. [10]	2023	AI-based IVN IDS survey	GenAI, federated learning, adversarial evaluation, software-defined vehicles, and benchmark maturity.
Luo et al. [14]	2023	Systematic survey of DL-based IVN IDS	Non-DL baselines, dataset maturity, reproducibility, and real-time deployment constraints.
Althunayyan et al. [43]	2026	Learning-based IVN IDS survey	Fleet-level operation, offloading, and poisoning risks in adaptive retraining.
Taxonomies and surveys of surveys			
Loukas et al. [9]	2019	Taxonomy of cyber-physical IDS for vehicles	AI-based families and evidence maturity per family.
Aliwa et al. [4]	2021	Cyberattacks and countermeasures for IVNs	Detection-specific synthesis and evaluation practice.
Karopoulos et al. [7]	2022	Survey of surveys and meta-taxonomy of IVN IDS	Primary-study evidence, measured temporal trends, and deployment readiness.
Dataset and synthetic-data reviews			
Lee et al. [18]	2023	Analysis of automotive IDS datasets	Full IDS taxonomy, method-level synthesis, and deployment readiness analysis.
Li et al. [16]	2026	Synthetic data for ML-based IVN IDS	The complete AI-based automotive IDS landscape across protocols and methods.
Connected-vehicle, IoV, and misbehavior-detection reviews			
Hasrouny et al. [46]	2017	VANET security challenges and solutions	In-vehicle detection and its link to fleet-level detection.
van der Heijden et al. [47]	2019	Misbehavior detection in cooperative ITS	In-vehicle and gateway-level detection evidence.
Ajibuwa et al. [44]	2023	AI/ML intrusion and misbehavior detection in networked autonomous systems	Automotive protocols, benchmarks, and ECU or gateway constraints.
Mahmoudi et al. [45]	2025	GenAI-based IoV IDS review	IVN, SOME/IP, benchmarking, robustness, and automotive deployment constraints.
Protocol-security surveys			
Douss et al. [29]	2023	In-vehicle protocols and Automotive Ethernet security	AI-based detection across protocols and its evaluation evidence.

4. Systematic Review Methodology

This section describes the methodology used to conduct the systematic literature review (SLR). The review was conducted following the guidelines for systematic literature

reviews in software engineering of Kitchenham and Charters [48], complemented by the backward and forward snowballing procedure of Wohlin [49], and it is reported in accordance with the PRISMA 2020 statement [50]. PRISMA is a reporting guideline; it was used to structure what is reported, not as a method for conducting the review. The completed PRISMA 2020 and PRISMA 2020 for Abstracts checklists are provided as Supplementary Materials. Further guidance on SLR conduct, systematic mapping, and evidence synthesis informed the protocol [51–57]. The review protocol was retrospectively registered in the Open Science Framework (OSF) on 17 September 2026 (Registration ID: u2c73), after the search had been completed; amendments made during revision are reported in Section 4.6.

4.1. Review Protocol and Search Strategy

The review protocol was defined before full screening to reduce selection bias and to ensure consistent decisions across the study-selection process. The protocol specified the research scope, search sources, search terms, inclusion and exclusion criteria, quality assessment rules, extraction fields, and synthesis strategy. The target population was peer-reviewed and technically relevant literature on automotive IDSs, with emphasis on AI-based methods for in-vehicle, connected-vehicle, and software-defined vehicle environments.

The search was conducted across IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Wiley Online Library, Scopus, Web of Science, and Google Scholar to capture automotive IDS literature across cybersecurity, vehicular communication, intelligent transportation, embedded systems, and AI venues. Snowballing was also used to identify additional relevant studies from the references of selected papers and from influential survey articles. Backward and forward snowballing improves coverage when terminology varies across related research communities [49].

The search string combined four groups of terms: automotive context, IDS and cybersecurity, AI and learning methods, and robustness or deployment concerns. Representative automotive terms included “automotive”, “vehicle”, “vehicular”, “in-vehicle network”, “CAN”, “CAN FD”, “SOME/IP”, “Automotive Ethernet”, “V2X”, “IoV”, and “software-defined vehicle”. IDS terms included “intrusion detection”, “anomaly detection”, “attack detection”, and “misbehavior detection”. AI-related terms included “machine learning”, “deep learning”, “neural network”, “CNN”, “RNN”, “LSTM”, “autoencoder”, “GAN”, “VAE”, “federated learning”, and “adversarial robustness”. The search terms were adapted to the syntax of each database. Appendix A, Table A2, reports the exact final query executed in each source, together with the filters applied and the search date; the total number of records retrieved is given in Figure 2.

The search covered studies published from January 2008 to July 2026, and the final search update was completed on 6 July 2026 for all sources. The review considered English-language journal articles, conference papers, and preprints that report primary research with clear technical relevance to automotive IDSs. Because several databases match search terms case-insensitively, short acronyms such as “CAN” also match common words; the resulting irrelevant records were removed during title and abstract screening, and this is discussed as a threat to validity in Section 4.5.

4.2. Eligibility Criteria

Studies were included if they met all of the following criteria (Appendix A, Table A1): (I1) they address an automotive, in-vehicle, connected-vehicle, IoV, V2X, or software-defined vehicle context; (I2) their primary contribution is one of three types, namely (a) proposing or evaluating an IDS, anomaly detector, or attack detector, (b) introducing or benchmarking a dataset for automotive IDS evaluation, or (c) evaluating attacks against an

automotive IDS; (I3) they provide sufficient technical detail to code the extraction fields; and (I4) they are written in English and were published between January 2008 and July 2026.

Studies were excluded if (E1) they are secondary studies, such as surveys, reviews, or surveys of surveys; (E2) they are methodological guidelines for conducting or reporting reviews; (E3) they are standards, regulations, specifications, or frameworks; (E4) they address generic enterprise, cloud, IoT, or network intrusion detection, or general machine-learning methods, without automotive evaluation; (E5) they address automotive security without a detection contribution, such as attack demonstrations, cryptographic protection, authentication, or formal protocol analysis; or (E6) their full text was unavailable. Sources excluded under E1–E5 may still be cited for background, methodology, or positioning, but they do not contribute to the descriptive statistics or the comparative synthesis.

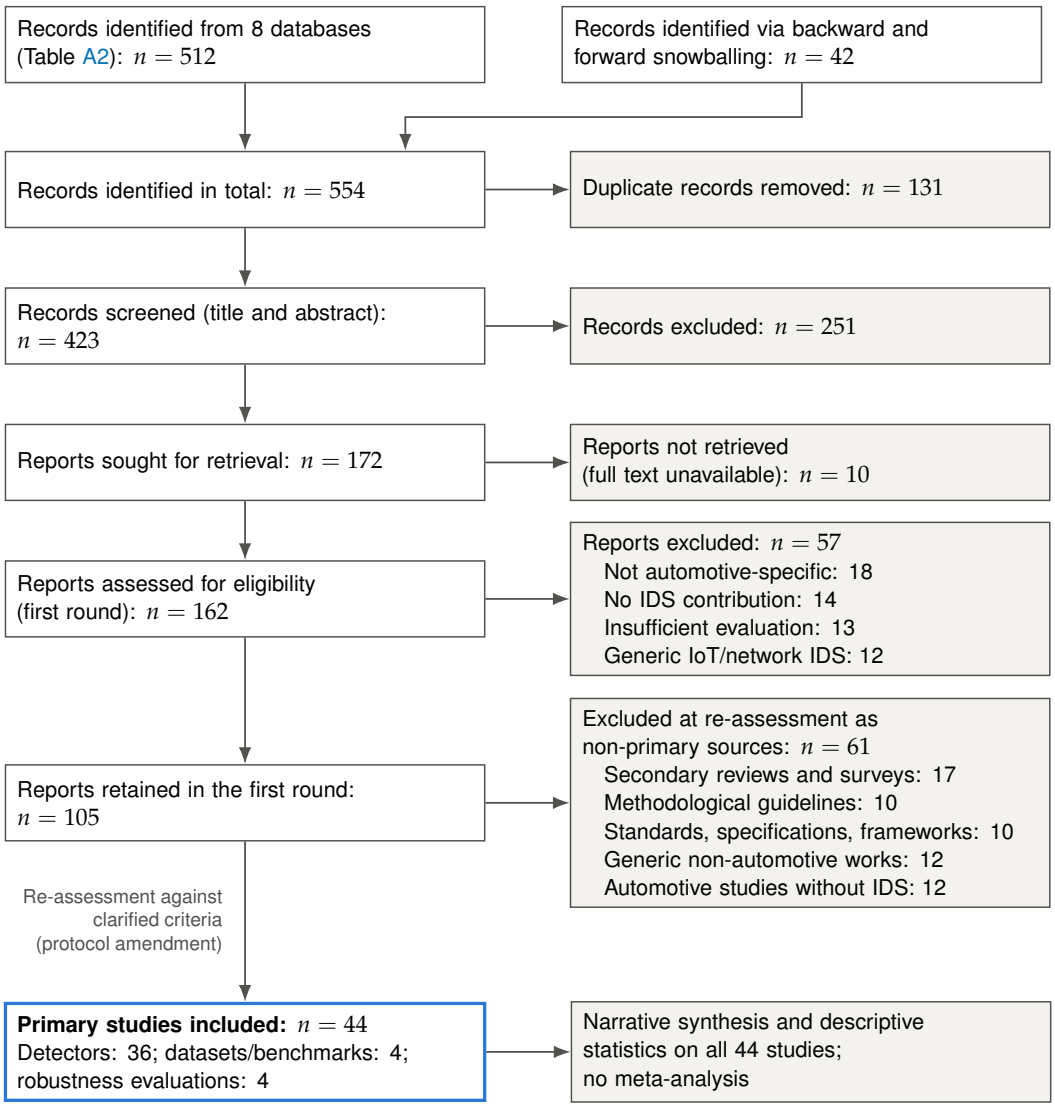


Figure 2. PRISMA 2020 flow diagram. Database and snowballing records were pooled before de-duplication. Full texts that could not be retrieved are reported as "not retrieved" rather than as eligibility exclusions. During revision, the 105 reports retained in the first round were re-assessed against the clarified eligibility criteria (Table A1; Section 4.6); 61 non-primary sources were removed from the evidence base and are cited only as background, methodological, or positioning references.

4.3. Screening and Study Selection

The study-selection process followed a staged screening procedure. First, records from the database searches and from snowballing were pooled, and duplicate records were

removed. Second, titles and abstracts were screened to remove clearly irrelevant papers. Third, full texts were retrieved and assessed against the eligibility criteria. Snowballing was applied to the references and citations of the selected papers and influential surveys, and the resulting records entered the same screening pipeline. Because screening was not independently duplicated by multiple reviewers, inter-rater agreement was not computed; ambiguous cases were resolved by revisiting the eligibility criteria and retaining studies only when automotive IDS relevance was explicit.

Figure 2 summarizes the selection process. The database searches returned 512 records and snowballing a further 42. After the removal of 131 duplicates, 423 records were screened, and 251 were excluded at the title and abstract stage. Of the 172 reports sought for retrieval, 10 could not be retrieved, and 57 of the 162 assessed reports were excluded, leaving 105 reports after the first round. During revision, these 105 reports were re-assessed against the clarified exclusion criteria E1–E5 (Section 4.6). This re-assessment removed 61 non-primary sources, namely 17 secondary reviews, 10 methodological guidelines, 10 standards, specifications, or frameworks, 12 generic non-automotive works, and 12 automotive studies without an IDS contribution (Appendix C, Table A9). The final evidence base comprises 44 primary studies.

4.4. Study Quality and Evaluation Practice

Study quality is addressed through the evaluation practice reported in the primary studies, which is analyzed across the evidence base rather than summarized in a per-study score, because limitations in reporting and evaluation are themselves a central finding of this review. Eight dimensions are considered: automotive relevance, transparency of the datasets used, attack coverage, clarity of the method, completeness of the evaluation, reproducibility, robustness testing, and deployment feasibility.

Table 3 describes the good practice that each dimension denotes, the source on which it is based, and the part of the review in which it is examined. The dimensions adapt the study-quality checklist of Kitchenham and Charters [48] to machine-learning security research, using the pitfalls identified by Arp et al. [58], namely sampling bias, data snooping, inappropriate baselines and performance measures, the base-rate fallacy, lab-only evaluation, and inappropriate threat models, as well as the closed-world critique of Sommer and Paxson [59] and the temporal-bias analysis of Pendlebury et al. [60]. They structure the analysis of datasets and evaluation practice in Section 8, the comparison of reproducibility and deployment readiness in Section 9, and the minimum reporting checklist in Section 10.2.

Table 3. Evaluation-practice dimensions examined across the evidence base.

Dimension	Good Practice	Where Examined	Basis
Automotive relevance	Real vehicle, testbed, or automotive dataset; protocol stated	Section 5.3	[48]
Dataset transparency	Dataset named, with preprocessing, class distribution, and access route	Section 8,	[58]
Attack coverage	Explicit threat model; several attack families; attacker capabilities stated	Sections 7 and 8.2	[58]
Method clarity	Architecture, features, hyperparameters, and windowing reported	Section 9.2	[48]
Completeness of evaluation	Leakage-resistant or cross-dataset split, baselines, and FPR/FNR or per-class results	Section 8.3	[58–60]
Reproducibility	Code or artefacts released, with preprocessing and split definitions	Section 9.2	[58]
Robustness testing	Unseen-attack, drift, adversarial, or cross-vehicle tests	Sections 8.4 and 9.6	[59,60]
Deployment feasibility	Latency and memory or CPU on embedded or automotive-grade hardware; placement stated	Sections 8.4, 9.2, and 9.5	[58]

4.5. Data Extraction, Synthesis, and Validity Control

A structured data-extraction protocol was used to code the included studies. The extracted information comprised bibliographic details, automotive context, network protocol, contribution type, IDS family, AI method, dataset, attack type, evaluation setup, metrics, evidence of reproducibility, and deployment evidence. It was also recorded whether attacks were studied and whether the evaluation covered only known attacks or also the more difficult cases of unknown, shifted, evolving, or adversarial attacks. This difference is crucial because a closed-set classification result may not reflect actual conditions in vehicle security. All study-level fields were coded manually against a fixed codebook (Appendix A, Table A3). Each study was assigned one primary IDS family according to its main detection mechanism, one contribution type, and one network environment.

Table 4 summarizes the extraction framework used in the review, and Table 3 lists the evaluation-practice dimensions examined across the evidence base.

Table 4. Data-extraction framework.

Dimension	Extracted Information
Automotive relevance	Vehicle context, protocol, network layer, IVN, IoV, V2X, SDV, gateway, ECU, diagnostic interface, or deployment relevance.
Contribution type	Detector; dataset or benchmark; robustness evaluation of an IDS.
IDS method	Entropy-based, timing/frequency/sequence-based, physical-characteristic-based, traditional ML, DL, generative, multi-stage or federated, adaptive or drift-aware, or hybrid IDS.
Dataset and attacks	Dataset name, real or synthetic source, attack classes, known or unknown attack setup, class imbalance, availability, and benchmark limitations.
Evaluation quality	Train–test split, cross-validation, cross-dataset testing, baseline comparison, reported metrics, latency, memory, CPU, energy, and false-positive evidence.
Reproducibility and deployment	Code availability, preprocessing details, hyperparameters, implementation details, deployment location, real-time feasibility, and embedded suitability.

The synthesis combined descriptive analysis and taxonomy development with comparative interpretation. The descriptive analysis reports the distribution of the primary studies by publication period, contribution type, IDS family, network environment, and venue. Taxonomy development categorized studies into foundational IDS families and AI-based IDS families. Comparative synthesis then explored the differences in detection capabilities, response to unknown attacks, maturity of the datasets, robustness, reproducibility, and deployment readiness.

No formal statistical meta-analysis was performed because the included studies differ in datasets, attack definitions, splits, metrics, and experimental assumptions. Instead, comparable metrics such as accuracy, precision, recall, F1-score, false-positive rate, false-negative rate, AUC, latency, memory usage, and model size were considered when available. Performance comparisons were made only where evaluation conditions were comparable. This approach avoids misleading rankings of IDS models trained and tested under incompatible conditions.

A number of validity threats were taken into account. Search bias can arise if relevant research is published under different terms or is not indexed in the databases searched; multiple databases and snowballing were used to reduce it. Short protocol acronyms such as “CAN” and “LIN” match common words and surnames in case-insensitive databases, which inflates the number of retrieved records and places more weight on title and abstract screening. Selection bias is possible because screening and coding were performed by a single reviewer; inclusion and exclusion criteria were applied consistently, and the coding

of every included study is reported in Appendix C so that it can be checked. When studies failed to provide complete experimental details, this was reported explicitly rather than inferred, which can introduce data-extraction bias. Eleven of the 44 primary studies (25.0%) are cited in their preprint versions, which may differ from later peer-reviewed versions. There is also a possibility of publication bias, as positive results are more likely to be published. Lastly, external validity is limited because most IDS models are tested on small datasets, a closed set of attacks, or offline experiments rather than on real vehicles. These restrictions are taken into account in the interpretation of results and in the gap analysis; the conclusions remain dependent on the availability and reporting quality of the included studies.

4.6. Protocol Amendments

Five amendments were made to the registered protocol during revision. (i) The eligibility criteria were clarified to separate primary studies from background sources (criteria E1–E5). (ii) The 105 reports retained in the first round were re-assessed against these criteria, and 61 non-primary sources were removed from the evidence base (Figure 2). (iii) The comparable-performance subset of 62 studies reported in the first version had been derived from the first-round set, so it is no longer reported. (iv) All study-level data were re-coded manually against the codebook in Table A3. (v) The per-study scoring scale described in the first version was replaced by an analysis of the same eight dimensions across the evidence base (Section 4.4, Table 3).

5. Descriptive Analysis of the Literature

This section offers a description of the 44 primary studies that form the evidence base. The aim is to present the automotive IDS literature in a concise, quantified summary before the taxonomy, the dataset analysis, and the comparative synthesis. The analysis reports the contribution type, publication period, IDS family, network environment, and venue of each study. These dimensions are used to determine whether the literature continues to be based primarily on CAN benchmark studies, or whether it has spread to modern hybrid vehicle architectures, unknown-attack handling, and deployment-aware AI-based IDSs.

5.1. Overview of the Evidence Base

Of the 44 primary studies, 36 (81.8%) propose or evaluate a detector, four (9.1%) introduce a dataset or benchmark [19,61–63], and four (9.1%) evaluate attacks against automotive IDSs [20,24,64,65]. Although the search window began in 2008, the earliest included study was published in 2011 [31], and 32 studies (72.7%) were published in 2020 or later. Deep learning is the largest family, with 12 studies (27.3%), followed by timing/frequency/sequence-based detectors, traditional ML, multi-stage or federated designs, datasets and benchmarks, and robustness evaluations, with four studies each.

Table 5 provides a primary overview of the studies by IDS family or contribution type, and Appendix C, Table A8, gives the detailed study-level coding. The venue distribution indicates that the evidence is spread across cybersecurity, vehicular communication, embedded systems, intelligent transportation, and applied AI communities: 13 studies (29.5%) appeared in IEEE venues, 11 (25.0%) are cited as arXiv preprints, six (13.6%) appeared in MDPI journals, four (9.1%) in ACM venues, and the remaining 10 (22.7%) in USENIX, NDSS, Elsevier, Springer, PLOS, Wiley, and other venues. This spread confirms that a narrow database search could bias the review toward one research community.

Table 5. Overview of the 44 primary studies by IDS family or contribution type. Period columns give the number of studies published in each period.

Family or Type	*n* (%)	2011–19	2020–22	2023–25	Networks	Studies
Foundational detectors						
Entropy-based	3 (6.8)	2	0	1	CAN	[30,31,66]
Timing, frequency, sequence	4 (9.1)	3	0	1	CAN	[22,32,36,67]
Physical-characteristic	3 (6.8)	2	1	0	CAN	[21,23,37]
Learning-based detectors						
Traditional ML	4 (9.1)	2	1	1	CAN	[11,12,39,68]
Deep learning	12 (27.3)	2	6	4	CAN; Ethernet (1)	[13,33–35,69–76]
Advanced-learning detectors						
Generative	3 (6.8)	1	2	0	CAN	[15,77,78]
Multi-stage, federated	4 (9.1)	0	1	3	CAN; SOME/IP; IoV; connected vehicle	[79–82]
Adaptive, drift-aware	3 (6.8)	0	0	3	CAN; CAN–Ethernet	[17,83,84]
Evaluation infrastructure						
Dataset, benchmark	4 (9.1)	0	1	3	CAN	[19,61–63]
Robustness evaluation	4 (9.1)	0	1	3	CAN	[20,24,64,65]
Total	44 (100)	12	13	19		

5.2. Temporal Evolution of IDS Families

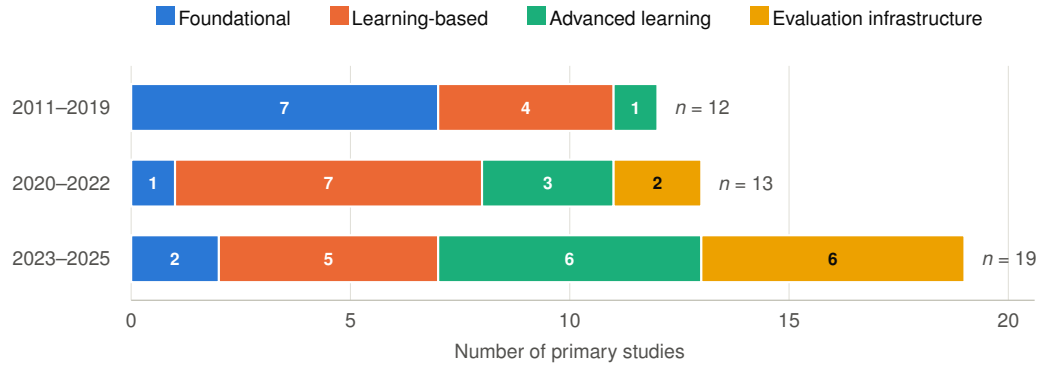
To quantify how automotive IDS research has shifted over time, the primary studies were grouped into three publication periods of comparable size (2011–2019, $n = 12$; 2020–2022, $n = 13$; 2023–2025, $n = 19$) and four method groups. Foundational detectors comprise the entropy-, timing/frequency/sequence-, and physical-characteristic-based families; *learning-based* detectors comprise traditional ML and DL; advanced-learning detectors comprise generative, multi-stage or federated, and adaptive or drift-aware designs; and evaluation infrastructure comprises dataset, benchmark, and robustness-evaluation studies. Two parameters are reported: the share of each group within a period, and the share of detectors that are foundational versus learning-based.

Figure 3 shows the result. Foundational methods accounted for 58.3% of the detectors published in 2011–2019 (7 of 12), but only 9.1% in 2020–2022 (1 of 11) and 15.4% in 2023–2025 (2 of 13). Over the same periods, the combined share of learning-based and advanced-learning detectors rose from 41.7% (5 of 12) to 90.9% (10 of 11) and 84.6% (11 of 13). Advanced-learning designs grew from 8.3% of all studies in 2011–2019 (1 of 12) to 23.1% in 2020–2022 (3 of 13) and 31.6% in 2023–2025 (6 of 19); adaptive or drift-aware detectors appear only from 2024 onward. Evaluation-infrastructure studies were absent before 2020 and represent 15.4% of studies in 2020–2022 (2 of 13) and 31.6% in 2023–2025 (6 of 19). The term "progression" is used in this measured sense only: it describes a change in research emphasis and does not imply that newer families outperform earlier ones under comparable conditions.

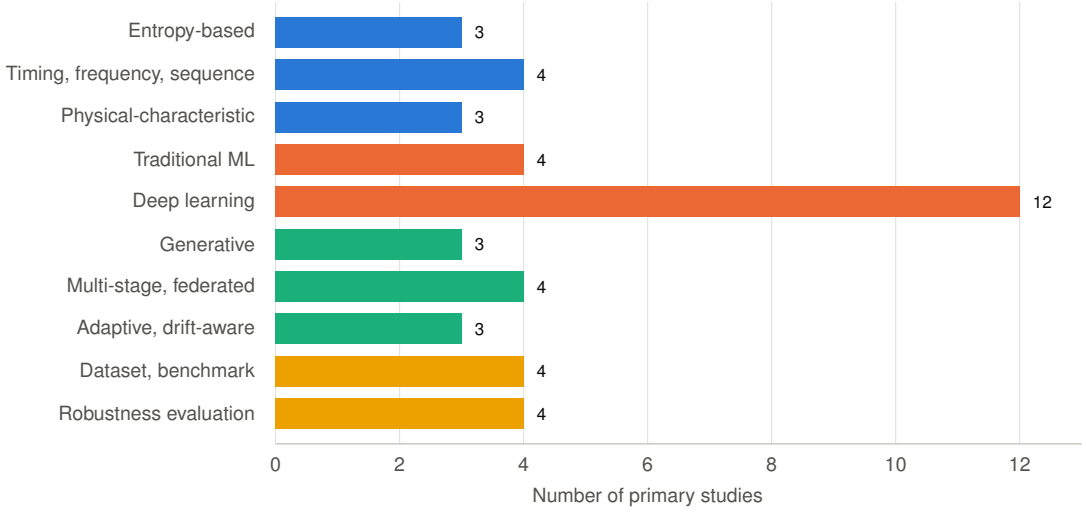
5.3. Network Environment, Method, and Attack Coverage

The network-environment distribution is a key descriptive dimension. Thirty-nine of the 44 primary studies (88.6%) address CAN only, including all four dataset and benchmark contributions and all four robustness evaluations. The five remaining studies cover Automotive Ethernet (AVTP) traffic [74], SOME/IP [82], hybrid CAN–Ethernet networks [84], CAN together with external IoV traffic [80], and federated detection in connected vehicles [81]; all five were published from 2022 onward. No included primary study evaluated detection on LIN or FlexRay traffic, and V2X misbehavior detection is represented in this review only by secondary and background sources [46,47,85,86]. CAN dominates because it is widely used, experimentally accessible, and well represented in public IDS datasets [69]. Nevertheless, modern vehicles increasingly combine CAN, CAN FD, LIN, FlexRay, Auto-

motive Ethernet, SOME/IP, diagnostic interfaces, domain controllers, cloud services, IoV, and V2X communication. A CAN-only IDS may not directly generalize to service-oriented middleware, Ethernet-based communication, or software-defined vehicle architectures. Therefore, each study is coded according to its network setting rather than treated as a generic automotive IDS paper. Dataset-focused reviews have also shown that benchmark coverage strongly affects how IDS results should be interpreted [18].



(**a**) Primary studies per publication period, by method group.



(**b**) Primary studies per IDS family or contribution type (bar colours follow the groups in (**a**)).

Figure 3. Distribution of the 44 primary studies. The share of foundational detectors falls after 2019, while advanced-learning designs and evaluation-infrastructure studies (datasets, benchmarks, and attacks against IDSs) grow in the most recent period. Five studies (11.4%) address a network other than CAN alone; the remaining 39 (88.6%) are CAN-only. Study-level coding is listed in Table A8.

The method distribution separates foundational IDS approaches from AI-centered approaches. Foundational approaches include entropy-based, timing-based, frequency-based, clock-based, voltage-based, and other physical-characteristic-based IDSs (10 studies). These methods are often lightweight and interpretable, making them relevant for gateways and resource-constrained ECUs. AI-based solutions include traditional ML, DL, generative models, adaptive systems, federated learning, and lightweight deployable models (26 detector studies). Traditional ML models are generally built with engineered features, whereas DL models can learn traffic representations, such as temporal, payload, or latent representations, directly from traffic data. The complexity of modeling traffic patterns and attack behavior makes deep learning a prominent research area [14], which is consistent with DL being the largest family in the corpus.

Attack coverage is another descriptive dimension. The studies are categorized by whether they focus on known attacks, unknown attacks, evolving attacks, or attacks on IDS models. Commonly studied known attacks are DoS, fuzzing, spoofing, replay, injection, impersonation, masquerade, and diagnostic abuse. Real vehicles, however, could be exposed to different, new, or altered attack scenarios. Thus, studies that assess unknown-attack detection, concept drift, out-of-distribution behavior, or adversarial manipulation offer more evidence of real-world readiness than studies limited to closed-set classification. In the corpus, explicit unknown-attack designs appear among generative and self-supervised detectors [15,71,77], drift handling appears among adaptive detectors [17,83,84], and attacks on IDS models are the subject of four robustness evaluations [20,24,64,65]. Generative and synthetic-data approaches are particularly pertinent because they may aid rare-attack generation, data augmentation, and unknown-pattern testing [16].

In general, the descriptive analysis shows that automotive IDS research is broad in methods but narrow in evidence. While CAN-based and closed-set evaluations are very prominent, other evaluation topics, such as hybrid-network evaluation, SOME/IP, Automotive Ethernet, cross-dataset validation, low false-positive operation, adversarial robustness, and deployment constraints, are less consistently addressed.

6. Taxonomy of Automotive IDS Approaches with Emphasis on AI-Based Methods

This section provides a compact taxonomy of automotive intrusion detection system (IDS) approaches, focusing on AI-based approaches. The taxonomy does not view automotive IDS as merely a machine-learning classification problem. Rather, it first identifies established IDS families grounded in information theory, communication parameters, and physical properties, and then situates ML, DL, generative, adaptive, robust, and deployment-oriented approaches in the broader automotive context. This matters because modern vehicles are hybrid cyber-physical systems that include CAN, CAN FD, LIN, FlexRay, Automotive Ethernet, SOME/IP, diagnostic interfaces, gateways, domain controllers, cloud connectivity, and V2X services.

A useful way to distinguish automotive IDS methods is by the evidence on which the IDS relies to make a detection decision. Some approaches track characteristics of traffic, the frequency of messages, timing behavior, or specific characteristics of the ECU. Others learn decision boundaries, payload patterns, temporal dependencies, latent representations, or fleet-level anomalies from data. A practical automotive IDS may combine several of these mechanisms rather than relying on one model. For example, a lightweight timing monitor can detect abnormal message rates, while a learning-based module can analyze suspicious payload or sequence patterns. This layered view is consistent with earlier automotive IDS surveys, which show that IDS design depends on network type, attack model, data availability, and deployment location [8].

6.1. Taxonomy Rationale

A taxonomy is needed because the automotive IDS literature is fragmented across protocols, datasets, attack types, and evaluation assumptions. A CAN-based IDS trained on known attacks is not directly comparable to an IDS for SOME/IP traffic, unknown-attack detection, federated learning, or embedded deployment. Similarly, an offline model reporting high accuracy on a balanced dataset provides different evidence from a model tested under class imbalance, unseen attacks, real-time latency limits, and ECU resource constraints.

This review organizes IDS approaches into two broad groups. The first group contains foundational automotive IDS families: information-theory-based IDSs, parametric IDSs, and physical-characteristic-based IDSs. These approaches are important because they are

often lightweight, interpretable, and closer to the communication behavior of in-vehicle networks. The second group contains AI-centered approaches: traditional ML-based IDSs, deep learning-based IDSs, generative and GenAI-assisted IDSs, multi-stage and adaptive IDSs, adversarially robust IDSs, and deployable lightweight IDSs.

Table 6 summarizes the principal strengths, limitations, unknown-attack readiness, and deployment readiness of the IDS families identified in this review, together with the number of primary studies in each family. Because the included studies use different datasets, attack definitions, preprocessing pipelines, and validation protocols, the table compares design trade-offs rather than ranking individual models. The readiness ratings follow the explicit rules stated in the table note.

Table 6. Comparative synthesis of automotive IDS families.

IDS Family	n	Main Strength	Main Limitation	Unknown-Attack Readiness	Deployment Readiness	Studies
Entropy- and frequency-based	3	Lightweight detection of abrupt traffic-distribution changes.	May miss stealthy or low-rate attacks that preserve aggregate statistics.	Low	High	[30,31,66]
Sequence- and timing-based	4	Effective for detecting disruptions in periodic message behavior.	Sensitive to window size, driving context, and dataset-specific timing.	Low to moderate	High for CAN monitoring	[22,32,36,67]
Physical-characteristic-based	3	Supports source identification and spoofing or masquerade detection.	Requires stable hardware-level measurements; may generalize poorly.	Moderate	Moderate	[21,23,37]
Traditional ML-based	4	Efficient classification using engineered traffic and payload features.	Depends on feature quality, preprocessing, and dataset representativeness.	Low to moderate	Moderate to high	[11,12,39,68]
Deep learning-based	12	Learns complex temporal, payload-level, or latent attack patterns.	Data hungry, less interpretable, and resource intensive.	Moderate to high	Moderate	[13,70,75,76]
Generative or synthetic-data-assisted	3	Supports rare-attack augmentation, imbalance handling, and unknown-pattern exploration.	Depends on the realism and protocol validity of synthetic data.	Moderate to high	Moderate	[15,77,78]
Multi-stage, hybrid, and federated	4	Combines fast screening with deeper analysis to balance speed and detection depth.	Requires careful thresholding and control of error propagation across stages.	Moderate to high	Moderate to high	[79–82]
Adaptive and drift-aware	3	Can adjust to changing vehicle behavior, software versions, and operating contexts.	Uncontrolled adaptation may absorb malicious behavior into the normal profile.	High	Moderate	[17,83,84]
Adversarial robustness evaluation	4	Evaluates resilience against evasion, perturbation, poisoning, or model-targeting attacks.	Increases evaluation complexity and may add computational overhead.	Moderate	Moderate	[20,24,64,65]

Unknown-attack readiness: High if detection relies on a learned model of normal behavior, uncertainty, or out-of-distribution scoring and the family includes evaluation on attacks unseen during training; Moderate if it models normal behavior for a single type of evidence, or combines supervised detection with partial open-set evaluation; Low if it applies fixed thresholds to aggregate statistics or relies on closed-set classifiers only. Deployment readiness: High if per-message computation is small and constant enough for ECU or gateway execution; Moderate if gateway- or domain-controller-class computation is needed; Low if server-class computation is required. Ratings are the author's judgment under these rules, not measured scores; *n* is the number of primary studies in the family.

Table 7 summarizes the compact taxonomy used in this review, linking each IDS family to its main detection evidence and typical role in automotive intrusion detection. Figure 4 shows the same taxonomy as a tree, together with the number of primary studies in each family.

Table 7. Taxonomy of automotive intrusion detection system approaches.

IDS Family	Main Detection Evidence	Typical Role	Studies
Information-theory-based IDS	Entropy, uncertainty, message distribution, and traffic randomness.	Lightweight anomaly monitoring.	[30,31,66]
Parametric IDS	Message frequency, timing intervals, periodicity, sequence behavior, and traffic rate.	Fast detection of abnormal communication behavior.	[22,32,36,67]
Physical-characteristic-based IDS	Clock skew, voltage profile, bit timing, signal behavior, and ECU fingerprint.	Source identification and spoofing detection.	[21,23,37]
Traditional ML-based IDS	Engineered traffic, payload, timing, entropy, or statistical features.	Efficient classification and baseline comparison.	[11,12,39,68]
Deep learning-based IDS	Temporal sequences, payload patterns, decoded signals, latent features, and learned representations.	Complex attack-pattern learning.	[33–35,69]
Generative and GenAI-assisted IDS	Synthetic attacks, data augmentation, rare-event generation, and latent distributions.	Data-scarcity handling and unknown-pattern testing.	[15,77,78,83]
Adaptive, robust, and deployable IDS	Drift handling, staged or federated detection, adversarial testing, latency, memory, energy, and model size.	Practical readiness for real vehicle environments.	[17,64,72,79]

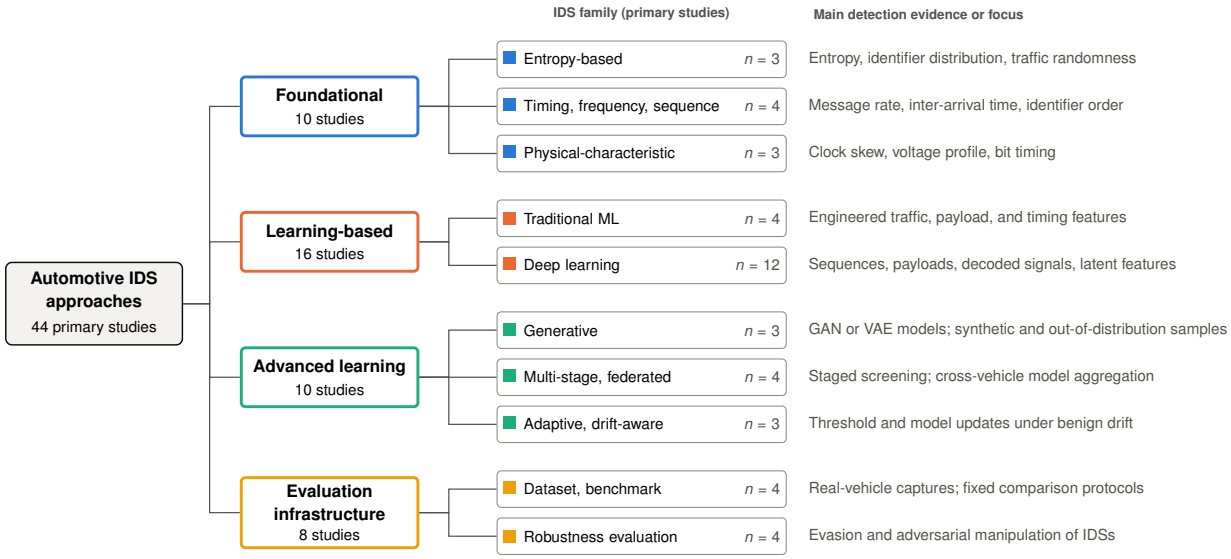


Figure 4. Taxonomy of automotive IDS approaches used in this review, with the number of primary studies in each family. Group colours match Figure 3; detailed strengths, limitations, and readiness ratings are given in Table 6.

6.2. Foundational Automotive IDS Families

Information-theory-based IDSs detect anomalies through changes in traffic entropy, identifier distribution, or communication randomness [30,31,66]. These approaches are computationally efficient and well suited for lightweight monitoring, but they may struggle against stealthy attacks that preserve normal statistical characteristics [30].

Parametric IDSs detect deviations in message frequency, timing intervals, periodicity, sequence order, and traffic rate. Because many CAN messages follow predictable periodic patterns, these methods can efficiently detect flooding, DoS, injection, and some masquerade attacks [32,36]. Their effectiveness depends on stable communication profiles and realistic evaluation under varying driving conditions [22].

Physical-characteristic-based IDSs use clock skew, voltage behavior, bit timing, signal shape, and ECU fingerprints to support source authentication and spoofing detection [21,23,37]. Their effectiveness depends on stable hardware measurements and

may be affected by environmental variation, device heterogeneity, and the limited availability of signal-level evidence in public datasets [21,37]. They can also be evaded by attackers who manipulate the physical signal itself [24].

6.3. AI-Based IDS Families

Traditional ML-based IDSs use engineered traffic, payload, timing, entropy, or diagnostic features with classifiers such as support vector machines, decision trees, random forests, and ensemble methods [11,12,39]. They offer an attractive balance between efficiency, interpretability, and detection performance, but they remain highly dependent on feature quality and dataset representativeness. Many studies also rely on closed-set evaluation, potentially overstating generalization to unseen attacks [10,39].

Deep learning-based IDSs learn temporal, payload-level, and latent traffic representations using architectures such as CNNs, RNNs, LSTMs, GRUs, autoencoders, transformers, and hybrid neural networks [13,69,70,75]. Their principal advantage is the ability to model complex attack patterns that are difficult to capture through engineered features. However, they are often data intensive, less interpretable, and more resource demanding than traditional ML approaches, requiring evaluation beyond classification performance alone [14,72,73]. Related online anomaly-detection research outside the automotive domain has also demonstrated the use of ensemble autoencoders for network intrusion detection [87].

Generative and GenAI-assisted IDSs address data scarcity, class imbalance, and unknown-attack evaluation using techniques such as GANs and VAEs [15,77,78]. Their effectiveness depends heavily on synthetic-data realism, requiring validation of protocol conformity, timing behavior, payload semantics, and downstream detection performance [16,77].

6.4. Adaptive, Robust, and Deployable IDS

Multi-stage and adaptive IDS approaches are important because real vehicle environments are dynamic. A multi-stage IDS may use a fast first stage for known-attack detection and a second stage for uncertain, unknown, or suspicious cases [79,80]. This design is useful because known attacks can often be detected quickly, while unknown or shifted attacks require deeper analysis. Adaptive IDSs support changing vehicle behavior through drift handling and model updates [17,83,84]. While adaptation can reduce false positives under evolving conditions, uncontrolled updating may incorporate malicious behavior into the normal profile, creating an important trade-off between flexibility and security [79,88]. Section 9.6 examines this trade-off in detail.

Adversarially robust IDSs consider attacks against the detection model itself, including evasion, poisoning, adversarial perturbation, and backdoor manipulation. Robustness testing is therefore an important complement to conventional detection metrics when assessing deployment readiness [64,89].

Deployment readiness depends not only on detection effectiveness but also on latency, throughput, memory consumption, energy requirements, model size, update strategy, and deployment location. These considerations are particularly important for ECU-, gateway-, edge-, and cloud-assisted IDS deployments, which operate under different resource constraints (Figure 1b; Section 9.5).

6.5. Interpretability and Trustworthiness of AI-Based IDS

Interpretability and trustworthiness are important for automotive IDS deployment because detection outputs may support safety-relevant monitoring, maintenance, or mitigation decisions. High classification performance alone is not sufficient if the model

cannot explain why traffic was classified as malicious, whether the decision is stable under distribution shift, or how reliable the output is under unseen attack conditions.

Recent adaptive and generative IDS designs further show why trustworthiness should be evaluated beyond ordinary accuracy. Adaptive autoencoders recalibrate their decision threshold as traffic changes [17], multi-stage federated designs separate fast screening from deeper analysis across vehicles [79], and generative adaptive designs synthesize new training samples when deviations from known patterns are detected, including protocol-aware variants for hybrid CAN–Ethernet settings with adversarial-robustness evaluation [83,84]. These studies are relevant because they connect synthetic-data-supported learning, concept-drift handling, adversarial resilience, and deployment-oriented constraints within the same IDS design direction.

For automotive IDSs, explanation should support auditability by showing whether a model relies on meaningful timing, payload, protocol, or contextual evidence rather than benchmark artifacts. General XAI literature frames this need around transparency, interpretability, and explanation quality, while high-stakes ML research also cautions that inherently interpretable models may be preferable to post-hoc explanations when decisions have operational consequences [90].

Explainability can support analyst review by identifying the features, message identifiers, timing patterns, payload structures, or protocol behaviors that influence a detection decision [81,91]. Signal-aware explainable detection has been demonstrated for CAN [34]. This is especially relevant for deep learning and generative IDS approaches, where internal representations are often difficult to inspect. Interpretable evidence can also help distinguish genuine attacks from unusual but benign vehicle behavior, reducing the risk of unnecessary alerts.

Trustworthiness also requires robustness against adversarial manipulation, concept drift, dataset bias, and unrealistic evaluation settings. An automotive IDS should therefore be assessed not only by F1-score or accuracy, but also by false-positive behavior, detection latency, resource usage, cross-dataset performance, adversarial resilience, and stability under changing vehicle or network conditions. These factors are necessary for determining whether an IDS is suitable for practical deployment in ECUs, gateways, edge monitors, or software-defined vehicle architectures.

7. Attack Types and Detection Targets

This section analyzes attack types and detection targets in automotive IDS research. Automotive attacks may target timing, identifier sequences, payload values, ECU identity, diagnostic services, Ethernet services, V2X messages, or the IDS model itself. Therefore, attack coverage should be interpreted in relation to the communication layer, deployment location, and evaluation assumptions.

7.1. CAN and IVN Attacks

CAN and other in-vehicle network attacks remain central because many public datasets and experimental IDS studies are based on CAN traffic. Common attacks include DoS, flooding, fuzzing, replay, spoofing, injection, impersonation, and masquerade. DoS and flooding attacks usually increase message frequency or occupy bus arbitration with high-priority identifiers. Their detection targets are therefore timing intervals, message rates, arbitration identifiers, and bus-load changes. Timing-based and frequency-based IDSs can detect many of these attacks quickly [36,67], but they may miss carefully paced attacks that preserve normal periodicity.

Replay, spoofing, injection, and impersonation attacks are more difficult because the message may look syntactically valid. A replay attack resends previously observed traffic,

while an injection attack introduces unauthorized frames into the bus. A spoofing or impersonation attack attempts to make a malicious message appear to originate from a legitimate ECU. Early CAN-security work showed that practical CAN attacks could exploit weak authentication and predictable behavior [41]. Wireless attack work further showed that external connectivity can become a path toward in-vehicle CAN abuse [40]. These attacks require IDS evidence beyond simple bus load, such as payload consistency, source fingerprinting, sequence validity, and context-aware plausibility.

Fuzzing and diagnostic-abuse attacks represent another IVN category. Fuzzing sends malformed or random messages to discover unexpected ECU behavior, while diagnostic abuse uses maintenance interfaces or UDS/OBD-II services outside expected operating conditions. A practical IDS should separate normal driving traffic from service, repair, and firmware-related traffic, because treating all diagnostic behavior as malicious would increase false positives.

7.2. Ethernet, SOME/IP, and Hybrid-Network Attacks

Modern vehicles increasingly use Ethernet, SOME/IP, domain controllers, gateways, and service-oriented communication. This changes the attack surface from message-level bus behavior to mixed protocol, service, and application behavior. Ethernet-oriented attacks may include ARP poisoning, SYN flooding, packet replay, man-in-the-middle behavior, scanning, service misuse, and malformed packet injection. In a hybrid vehicle architecture, an attacker may compromise an external or Ethernet-facing interface and then influence CAN, diagnostic, or service-level behavior through a gateway. Automotive Ethernet has also been studied directly in intrusion-detection settings, including AVTP traffic [74].

SOME/IP attacks are important because SOME/IP supports service discovery, remote-procedure-style communication, event notification, and application-level data exchange. Relevant attacks include service spoofing, replay, DoS, fuzzing, malformed serialization, de-serialization abuse, and unauthorized service invocation. Formal analysis of SOME/IP services shows that protocol-level weaknesses can affect service integrity and availability [92]. For IDS design, the detection target is not only an abnormal packet count but also service identity, method identifier, event group behavior, message type, payload structure, and caller–provider consistency. Related SOME/IP security work has examined service protection, authentication, and secure communication [93–95], and a multi-layer IDS has been proposed for SOME/IP-based in-vehicle networks [82].

Hybrid-network attacks require cross-layer interpretation. An Ethernet anomaly may be harmless in isolation but suspicious if it is followed by abnormal CAN traffic or diagnostic requests. Similarly, a gateway IDS may observe only part of the traffic and may not see the physical source of a message. The same attack may require different evidence depending on whether the IDS is deployed at an ECU, gateway, in-vehicle monitor, or edge service (Figure 1b). Protocol-aware adaptive detection has been proposed for hybrid CAN–Ethernet networks [84], but it remains one of very few primary studies in this setting.

7.3. V2X and IoV Attacks

V2X and IoV environments extend the detection target beyond the vehicle boundary [46]. Attacks in this layer include Sybil attacks, false data injection, message replay, jamming, routing attacks, man-in-the-middle attacks, bogus event reporting, and location manipulation. Potential cyberattacks against automated vehicles include threats against sensors, control logic, map information, and cooperative automation assumptions [85].

False data injection and Sybil attacks are particularly important because the attacker may use syntactically valid messages with misleading semantic content. A V2X IDS therefore needs semantic and context-aware evidence, such as position plausibility, speed

consistency, map consistency, neighbor agreement, trajectory feasibility, and trust history. Misbehavior-detection surveys emphasize that connected-vehicle security must detect insider behavior, not only outsiders without credentials [47]. This differs from IVN settings, where detection often focuses on timing, payload, or identifier anomalies.

Connected and cooperative driving also introduces control-level consequences. Attacks on connected-vehicle streams can disturb cooperative adaptive cruise control and platooning behavior, even when the attack is expressed through communication rather than direct physical access [86]. For this reason, V2X and IoV attack detection should be evaluated both by packet-level accuracy and by the protection of safety-relevant behavior. Among the primary studies, only one multi-tiered design evaluates external IoV traffic alongside CAN [80], and one addresses federated detection across connected vehicles [81].

7.4. Unknown, Evolving, and Model-Targeting Attacks

A key distinction in this SLR is between known-attack detection and unknown-attack readiness. Known attacks are represented in training or explicitly defined in the threat model. Unknown attacks are unseen, shifted, rare, modified, or evolving attack patterns. In practice, attackers may change message rate, payload values, target ECUs, service identifiers, timing, or attack sequence to avoid a detector trained on fixed benchmark behavior. This creates a gap between closed-set IDS accuracy and real deployment readiness.

Unknown attacks require different detection targets from known attacks. Rather than only learning labels, an IDS may need to model normality, uncertainty, reconstruction error, temporal consistency, protocol validity, or context plausibility [15,71,76]. Generative and synthetic-data approaches can support rare-attack exploration, but synthetic attacks must preserve protocol semantics and timing realism. Adaptive IDSs can update to new benign behavior, but uncontrolled adaptation may learn malicious behavior as normal. Multi-stage IDSs can reduce this risk by detecting known attacks quickly and routing uncertain cases to deeper analysis.

Attacks against IDS models add another layer. Evasion attacks attempt to craft inputs that bypass detection. Poisoning attacks contaminate training data. Backdoor attacks create hidden behaviors that activate under specific triggers. Adversarial manipulation has been shown to be relevant for automotive IDS evaluation [20,64,65], and physical-layer detectors can also be evaded [24]. Therefore, attack coverage should include not only attacks against the vehicle network, but also attacks against the detection pipeline, training process, model update mechanism, and decision threshold.

For synthesis, this review treats detection targets as five connected categories:

- communication evidence, including timing, frequency, identifiers, and payload structure;
- source evidence, including ECU fingerprints, clock behavior, voltage profiles, and sender consistency;
- service evidence, including SOME/IP services, diagnostic requests, and gateway behavior;
- semantic evidence, including vehicle state, location plausibility, trajectory consistency, and cooperative context;
- model evidence, including uncertainty, drift, adversarial sensitivity, and update robustness.

This mapping supports later comparison because an IDS that performs well against CAN flooding is not necessarily suitable for SOME/IP service misuse, V2X false data injection, or adversarial evasion. Automotive IDS evaluation should report which attack target is detected, which network layer is observed, and whether the test covers only known attacks or also unknown, evolving, and model-targeting threats.

8. Datasets, Synthetic Data, and Evaluation Practices

Datasets and evaluation practices strongly shape how automotive IDS results should be interpreted. A model may report high accuracy on a controlled CAN dataset, but the same result may not generalize to another vehicle, protocol, driving condition, gateway architecture, or unseen attack pattern. Table 8 summarizes the principal dataset categories used in automotive IDS research, together with their typical sources, attack coverage, realism, and major benchmarking limitations.

Table 8. Dataset maturity in automotive IDS research.

Dataset Type	Typical Source	Attack Coverage	Realism and Main Limitation	Studies
Real vehicle datasets	Captured from vehicles, ECUs, CAN buses, gateways, or testbeds.	Usually covers selected attacks such as DoS, spoofing, replay, fuzzing, or injection.	High realism, but often limited in size, vehicle diversity, protocol coverage, and public availability.	[19,21,62]
Synthetic datasets	Generated through simulation, traffic injection, synthetic attack generation, or model-based augmentation.	Can cover rare, unknown, or controlled attack variants.	Useful for coverage expansion, but realism depends on whether timing, payload, protocol, and driving context are preserved.	[16,77,96]
Hybrid datasets	Combine real benign traffic with injected, simulated, or generated attacks.	Often broader than purely real datasets and more controlled than uncontrolled field data.	Moderate realism; attack injection may still simplify attacker behavior or vehicle dynamics.	[13,36,75]
Benchmark datasets	Public datasets reused across multiple IDS studies.	Supports comparison across methods, but often focuses on a narrow set of attacks or protocols.	Useful for reproducibility, but repeated use may encourage dataset-specific optimization rather than generalization.	[18,61,63]

Automotive IDS datasets can be grouped into real, synthetic, hybrid, simulated, and benchmark datasets. Real datasets are collected from physical vehicles, ECUs, CAN buses, gateways, diagnostic tools, or laboratory testbeds. However, real vehicle datasets are often limited in size, vehicle diversity, attack diversity, protocol coverage, and public availability. They may also focus on one vehicle platform or one communication layer, which limits generalization across modern hybrid architectures. All four dataset and benchmark contributions among the primary studies are CAN-based [19,61–63].

Benchmark datasets are valuable because they allow repeated comparison across IDS methods. For example, early deep-learning IDS evaluation on in-vehicle network traffic helped establish the usefulness of data-driven detection, but it also demonstrated the dependence of reported performance on dataset structure and attack representation [75]. Benchmark reuse creates a risk of dataset-specific optimization, where a model learns fixed injected identifiers, repeated payload patterns, or preprocessing artifacts rather than generalizable intrusion behavior.

More recent dataset work has attempted to address these limitations by improving documentation, attack realism, and comparability. The ROAD dataset was introduced to support more realistic CAN IDS benchmarking, including ambient driving data and attacks with different stealth levels [19]. Similarly, the can-train-and-test dataset was designed to support evaluation across multiple vehicles and attack captures, which is useful for studying whether IDS models generalize beyond a single vehicle configuration [62]. Benchmarks of time-based detectors [61] and of unsupervised online detectors for masquerade attacks [63] further enable like-for-like comparison. These datasets are important because

automotive IDS evaluation should not only show that a model works on one benchmark, but also whether it remains reliable across vehicles, attack types, and traffic distributions.

Hybrid and simulated datasets remain necessary because real attack data are difficult to collect safely and ethically. Hybrid datasets combine real benign traffic with injected, simulated, or generated attacks, while simulation environments can support controlled scenario generation and repeatable testing. Driving simulators such as CARLA are not IDS datasets by themselves, but they illustrate how simulation can support controlled vehicle-behavior and sensor-context generation when real-world data collection is limited [96]. The main limitation is realism: injected or simulated attacks may not fully reflect the timing, stealth, intent, or adaptive behavior of real attackers. Dataset maturity should therefore be judged by protocol coverage, attack diversity, realism, documentation, accessibility, and suitability for cross-dataset testing, not only by the number of samples.

8.1. Synthetic Data and Generative Augmentation

Because real automotive attack data are scarce, imbalanced, and often narrow in scope, synthetic-data generation has become increasingly important. Generative adversarial networks, variational autoencoders, simulation-based traffic generation, and model-based augmentation can create rare attacks, balance classes, or test unknown attack behavior. However, generated samples should not automatically be assumed to represent realistic automotive traffic. Many IDS models are trained under closed-set conditions, where attack types in the test set are also present during training. In practice, attackers may modify message timing, payload values, service identifiers, attack rates, diagnostic requests, or target ECUs. GAN-based in-vehicle IDS work shows how generative learning can support unknown-attack detection from normal traffic, but generated or learned representations still require careful validation against realistic vehicle behavior [77].

Synthetic data should not be treated as automatically realistic. For example, synthetic CAN frames may match identifier frequency but not represent physically plausible vehicle states. Similarly, generated SOME/IP traffic may imitate packet distributions without preserving service logic, method identifiers, event groups, or caller–provider relationships. If unrealistic synthetic samples are used for training or testing, reported performance may become misleading. Useful checks include protocol validity, payload plausibility, timing consistency, class diversity, similarity to real traffic, and effect on downstream detection [16].

8.2. Features, Labels, and Attack Coverage

Feature extraction is another major source of variation across studies. Traditional ML-based IDSs commonly rely on engineered features such as CAN identifiers, payload bytes, message frequency, inter-arrival time, entropy, diagnostic services, and traffic-window statistics. Deep learning models may use raw payloads, temporal sequences, reconstructed representations, or learned embeddings. Unsupervised LSTM-autoencoder approaches such as CANnolo, for example, model message behavior across identifiers and time and flag deviations through reconstruction error [76], and recurrent autoencoders have been designed with embedded automotive platforms in mind [72].

Sequence-based and timing-based features remain important because many in-vehicle messages follow regular periodic patterns. IDS methods based on CAN identifier sequences can detect abnormal message-order behavior, but their reliability depends on traffic stability, window selection, and attack stealth. Work on CAN ID-sequence analysis shows that sequence-level evidence can be useful for anomaly detection, especially when attacks disturb normal communication order [32]. However, these features may be less effective

when an attacker preserves frequency and timing while changing the signal-level meaning of payload values.

Signal-level features provide a different type of evidence. Instead of observing only identifiers or binary payload bytes, signal-aware IDS approaches attempt to model translated vehicle signals and their temporal consistency. CANShield is an example of a signal-level deep-learning IDS that evaluates CAN data at different temporal granularities, which is relevant for detecting attacks that appear normal at the raw message level but abnormal in physical or semantic behavior [35]. This distinction matters because future IDS evaluation should not treat all feature types as equivalent: identifier-level, payload-level, timing-level, signal-level, and physical-fingerprint evidence each require different data access and deployment assumptions (Figure 1a).

Labels also require careful interpretation. Binary labels support simple evaluation but hide differences between DoS, fuzzing, spoofing, replay, injection, impersonation, masquerade, diagnostic abuse, and service misuse. Multi-class labels provide more detail but may still represent only known attacks. Open-set, out-of-distribution, and anomaly-based evaluation are therefore important for assessing whether a model can detect attacks not seen during training [15]. Attack coverage should also reflect modern vehicle architectures. CAN attacks remain important, but automotive IDS evaluation should not be limited to CAN-only datasets when Ethernet, SOME/IP, diagnostics, gateways, V2X, and IoV introduce additional detection targets.

8.3. Evaluation Metrics and Validation Protocols

Accuracy, precision, recall, F1-score, and AUC are widely used in automotive IDS studies. These metrics are useful for classification assessment, but they are insufficient for deployment readiness. A high F1-score does not prove that an IDS can run on an ECU, gateway, or edge device. For safety-relevant monitoring, false-positive and false-negative rates are especially important, because unnecessary alerts can overload response workflows, while missed attacks can allow malicious behavior to persist.

Latency and resource metrics are equally important. Automotive IDSs may need to operate under strict timing, memory, CPU, and energy constraints. Detection latency, throughput, inference time, model size, memory footprint, CPU load, and energy consumption should therefore be reported when deployment is claimed. Without these metrics, it is difficult to determine whether an IDS is suitable for ECU-level, gateway-level, edge-assisted, or cloud-assisted deployment.

Validation protocol is another key issue. Random train–test splits can overestimate performance when similar traffic windows from the same vehicle or attack sequence appear in both training and testing. More reliable evaluation may require time-based splits, vehicle-level splits, cross-dataset validation, unseen-attack testing, or protocol-level generalization tests [60]. Broader intrusion-detection literature has warned that machine-learning IDS evaluation can become misleading when models are tested under closed-world assumptions that do not reflect deployment conditions [59], and general security-ML guidance identifies data snooping and the base-rate fallacy as recurring pitfalls [58]. This warning is directly relevant to automotive IDSs, where closed-set benchmark accuracy may not represent unknown attacks, unseen vehicles, shifted traffic, or adversarial manipulation.

Class imbalance also affects evaluation because automotive datasets often contain far more benign traffic than malicious traffic, or an uneven representation of attack families. Accuracy can be misleading under imbalance; precision, recall, F1-score, false-positive rate, false-negative rate, and per-class reporting should therefore be interpreted together. Preprocessing transparency is also necessary for fair comparison. Studies should report filtering rules, window sizes, feature normalization, label construction, class balancing,

train–test separation, hyperparameter selection, and baseline configuration. Without these details, it is difficult to determine whether a reported improvement comes from the IDS model itself or from differences in data preparation.

8.4. Robustness and Deployment-Oriented Evaluation

Robustness evaluation should include unknown attacks, shifted attacks, adversarial manipulation, concept drift, and changing driving conditions. An IDS trained on one vehicle, one dataset, or one attack configuration may fail when deployed in another environment. This is especially relevant for AI-based IDSs, because models can be sensitive to distribution shift. Out-of-distribution testing, adversarial evaluation, and drift-aware validation provide stronger evidence of real-world readiness than a single closed-set benchmark score.

Deployment-oriented evaluation should also consider where the IDS is placed. ECU-level IDSs require low memory, low latency, and minimal computational overhead. Gateway IDSs can monitor multiple network segments but may still face strict timing constraints. Edge-assisted IDSs may support heavier models, while cloud-assisted IDSs can enable fleet-level analytics but introduce privacy, connectivity, and latency concerns. Physical-fingerprint methods, such as clock-based ECU fingerprinting, also show that deployment location and the available measurement signals affect what evidence an IDS can use [21]. A fair evaluation should therefore connect metrics to the intended deployment location.

Overall, dataset and evaluation maturity remains uneven. The strongest evidence comes from studies that use transparent datasets, realistic attacks, clear preprocessing, appropriate baselines, reproducible validation, cross-dataset testing, false-positive analysis, robustness testing, and deployment-relevant metrics. Future automotive IDS research should move beyond isolated accuracy reporting toward evaluation designs that reflect hybrid vehicle networks, unknown attacks, synthetic-data limitations, resource constraints, and operational reliability.

9. Comparative Synthesis, Reproducibility, and Deployment Readiness

This section compares automotive IDS approaches according to the criteria most relevant for practical use: detection speed, false-positive control, robustness, adaptability, interpretability, reproducibility, and deployability. The goal is not to rank individual studies by accuracy, because the reviewed works use different datasets, attack definitions, preprocessing pipelines, validation protocols, and hardware assumptions. Instead, the synthesis identifies recurring design trade-offs and evidence gaps across IDS families. This consolidated view addresses RQ5–RQ8 by comparing single-stage, multi-stage, adaptive, generative, and adversarially robust IDS approaches while also considering benchmarking quality, deployment barriers, trustworthy operation, and the fleet-level, offloaded, and adaptive settings in which IDSs will be deployed. The comparative discussion builds on Table 6.

9.1. Comparative Analysis of IDS Families

Overall, no IDS family dominates across all evaluation dimensions. Lightweight statistical IDSs provide fast and interpretable detection of traffic anomalies, but threshold-based approaches may miss low-rate attacks that preserve normal aggregate statistics [31]. Sequence- and timing-based IDSs exploit temporal communication patterns and are effective for known-attack detection, but evidence for unknown-attack and cross-vehicle generalization remains limited [68,70].

Traditional ML-based IDSs combine diverse features efficiently, but they may overfit dataset artifacts. Their evaluation should therefore include feature transparency, false-

positive rates, and cross-dataset validation. Deep learning-based IDSs capture complex temporal and payload patterns, but they often require deployment on gateways, domain controllers, or edge-assisted platforms because of their computational overhead [80]. Generative IDSs address data scarcity and class imbalance, but their effectiveness depends on synthetic-data realism and should be evaluated using both detection performance and data-quality metrics.

Multi-stage IDSs combine fast filtering with deeper analysis [79,80], but they are vulnerable to error propagation. Evaluations should therefore report stage-level errors, overall detection performance, and latency. Adaptive IDSs address evolving vehicle behavior but risk incorporating malicious activity during retraining. Consequently, drift monitoring, controlled updates, and rollback mechanisms are essential, particularly in software-defined vehicles (Section 9.6). Adversarially robust IDSs require evaluation beyond conventional metrics, because high accuracy may not translate into resilience against evasion, poisoning, or model-targeting attacks [97]. This challenge also affects large-model-assisted IDS designs [98].

9.2. Reproducibility, Benchmarking, and Deployment Readiness

Deployment readiness requires more than classification metrics. Accuracy, precision, recall, F1-score, and AUC are useful, but they do not show whether an IDS can operate under automotive timing, memory, CPU, energy, and communication constraints. ECU-level IDSs require low latency and minimal overhead, while gateway and domain-controller IDSs may support richer analysis but must still avoid delaying safety-relevant communication. Edge-assisted and cloud-assisted IDSs can support heavier models and fleet-level analytics, but they introduce additional latency, privacy, connectivity, and data-governance constraints.

Lifecycle controls are equally important. Automotive cybersecurity is managed across development, production, operation, maintenance, and decommissioning [99]. ISO/SAE 21434 frames cybersecurity engineering as a lifecycle process, while UN Regulation No. 155 requires cybersecurity management evidence for vehicle approval [25,26]. Software-update engineering and management are additionally addressed by ISO 24089 and UN Regulation No. 156 [100,101]. Therefore, an IDS evaluated only as an offline classifier provides incomplete deployment evidence. A deployable IDS should specify how alerts are monitored, thresholds are adjusted, drift is detected, false positives are reviewed, and unsafe model updates are prevented.

9.3. Trustworthiness and Practical Interpretation

Trustworthy automotive IDS deployment also depends on robustness and interpretability. A model may perform well on known attacks but fail under shifted benign traffic, unseen attacks, software updates, adversarial perturbations, or changing operational conditions. Interpretability is therefore not only a usability feature; it supports debugging, analyst review, root-cause analysis, and regulatory explanation when IDS outputs affect safety-relevant monitoring or response workflows [102].

The strongest evidence comes from studies that combine transparent datasets, leakage-resistant validation, realistic baselines, cross-dataset testing, robustness evaluation, resource measurements, deployment-context assumptions, and lifecycle controls. This does not mean every IDS must be tested on production hardware, but claims about practical readiness should be supported by evidence that reflects the intended operating environment.

9.4. Distributed and Federated IDS Architectures for Vehicle Fleets

Fleet-level detection can identify attacks that are invisible to a single vehicle, such as a campaign that exploits the same telematics vulnerability across many vehicles, and it allows

a model learned in one vehicle to protect others. Three architectures can be distinguished. Centralized aggregation sends traffic features or alerts to a cloud backend or vehicle security operations centre. It is simple, but it creates a single point of failure, a backhaul bottleneck, and privacy exposure, and its latency depends on connectivity. Hierarchical federated learning trains locally in each vehicle and aggregates model updates first at roadside units or edge servers and then in the cloud [79]. Raw traffic stays in the vehicle, the backhaul load falls, and regional models can follow local conditions; federated learning has also been combined with explainability for connected-vehicle IDSs [81]. Decentralized (peer-to-peer) learning exchanges model updates between vehicles over V2V links without a central aggregator. It removes the bottleneck, but convergence becomes difficult under high mobility and intermittent links. Figure 5 contrasts the three architectures.

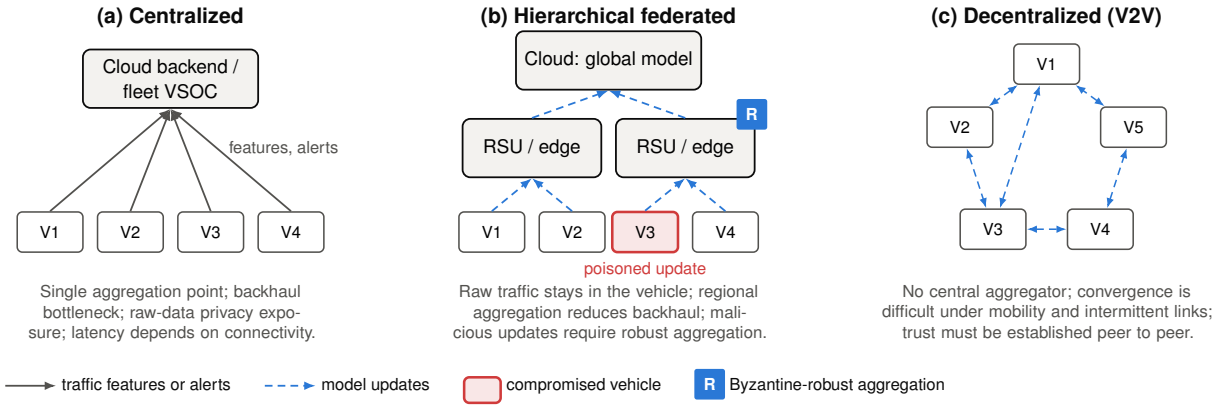


Figure 5. Fleet-level IDS architectures. (**a**) Centralized aggregation of features or alerts. (**b**) Hierarchical federated learning with aggregation at roadside or edge servers before the cloud; a compromised vehicle can submit poisoned updates, which robust aggregation must tolerate. (**c**) Decentralized exchange of model updates between vehicles. Only two primary studies evaluate federated detection [79,81], and none reports communication cost or robustness to malicious clients.

Synchronizing threat intelligence across a fleet raises four challenges [88]. First, data are not independently and identically distributed: vehicle models, ECU configurations, and software versions produce different benign traffic, so a single global model may fit no vehicle well, and clustered or personalized aggregation is needed. Second, connectivity is intermittent, so aggregation must tolerate stale or missing updates through asynchronous schemes and partial participation. Third, communication cost on V2X and cellular links must be limited through update compression and sparse participation. Fourth, trust cannot be assumed: a compromised vehicle can submit poisoned updates or implant a backdoor, so Byzantine-robust aggregation, secure aggregation, and the misbehavior-reporting mechanisms developed for cooperative intelligent transportation systems are required [47,88]. As an alternative to exchanging model updates, fleets can share compact threat intelligence, such as indicators, anomaly scores, or misbehavior reports, which lowers bandwidth and makes provenance easier to verify.

The evidence base for these architectures is thin. Only two of the 44 primary studies use federated detection [79,81], and none of the primary studies reports communication cost per round, convergence under client dropout, or detection performance when a stated fraction of vehicles is malicious. Future fleet-level IDS studies should report these quantities alongside detection metrics, because they determine whether a federated design avoids the bottleneck it is meant to remove.

9.5. Computational Offloading for In-Vehicle Gateways

Software-defined vehicles generate large data streams from sensors, ECUs, and services. However, an IDS does not need to process raw sensor data; it monitors communication and service behavior, so data volume can first be reduced at the source by computing windowed features at the gateway and forwarding features instead of raw frames. Detection can then be placed across tiers: the ECU, the zonal or central gateway, the in-vehicle high-performance computer or domain controller, roadside or multi-access edge computing, and the cloud (Figure 1b). Each tier further from the bus offers more computation but less timing determinism and less availability.

Five offloading strategies are relevant for deploying deep learning models at resource-constrained in-vehicle gateways. (i) Cascaded detection handles most traffic with a lightweight first stage on the gateway and escalates only uncertain windows to a heavier model [79,80]. (ii) Early-exit networks stop inference at an intermediate layer when the prediction is already confident. (iii) Model partitioning runs the first layers on the gateway and sends compact intermediate features to the in-vehicle computer or edge server. (iv) Model compression, such as quantization, pruning, and knowledge distillation, fits deep models within gateway memory and compute limits; lightweight recurrent autoencoders designed for automotive embedded systems illustrate this direction [72]. (v) Opportunistic edge offloading moves analysis to edge or roadside infrastructure when connectivity allows, with a local fallback.

Offloading is worthwhile only if transfer, remote inference, and return together are faster than local inference and still meet the detection deadline:

$$\frac{D}{B} + \frac{C}{f_{\text{edge}}} + T_{\text{rtt}} \; < \; \min\left(\frac{C}{f_{\text{local}}}, \; T_{\text{deadline}}\right), \tag{1}$$

where D is the size of the transmitted data, B the link bandwidth, C the computational load of the model, f the processing capability of each tier, and T_{rtt} the round-trip latency. Because connectivity is not guaranteed, the detection of safety-relevant attacks must not depend on offloading. Offloading suits analysis that is not time-critical, such as forensics, fleet correlation, and retraining, whereas detection on the bus must meet its timing locally. Among the primary studies, staged designs [79,80] and embedded-oriented models [72] address parts of this problem, but no study evaluates detection performance and offloading latency together under realistic connectivity.

9.6. Poisoning Risks in the Retraining Lifecycle of Adaptive IDS

Adaptive IDSs follow benign drift by retraining on recent traffic, recalibrating thresholds, pseudo-labeling new data, or generating new training samples [17,83,84]. The same mechanism allows an attacker to shift the learned notion of normal behavior. An attacker who can inject traffic gradually, for example through a compromised ECU or an aftermarket device, can move the profile in small steps until the target attack is classified as benign. Because each step resembles ordinary drift, such slow poisoning is harder to detect than an abrupt attack.

Six entry points can be identified along the retraining lifecycle (Figure 6): (1) data collection, where low-rate malicious traffic is absorbed into the benign training set; (2) labeling, where the IDS labels its own retraining data, so errors reinforce themselves; (3) generative augmentation, where synthetic samples inherit a poisoned distribution; (4) federated aggregation, where malicious vehicles submit poisoned or backdoored updates [88]; (5) threshold recalibration, where decision thresholds are relaxed gradually; and (6) model distribution, where a tampered model is delivered through the update channel. The resulting attacks may target availability (raising false positives until operators disable the IDS), integrity

(making a specific attack appear normal), or the model itself through a backdoor triggered by a specific pattern.

Benign drift can often be separated from poisoning by tying it to authorized events. Changes that follow a signed software update, a new ECU configuration, or a newly deployed service are expected, whereas drift outside such events should be quarantined for analyst review. Seven controls follow from this view: retraining only on traffic from vehicles and time windows with a verified software state (provenance); testing every updated model on a fixed set of known attacks and benign cases before deployment and rejecting updates that reduce detection (golden-set gating); capping how far a model or threshold may move in a single update (bounded updates); filtering outliers from retraining data (sanitization); running a frozen, non-adaptive reference model in parallel and investigating disagreements; using Byzantine-robust aggregation for federated updates (Section 9.4); and keeping previous models available for rollback, with human approval for updates. Treating IDS model updates as software updates places them under the integrity and management requirements of ISO 24089 and UN Regulation No. 156 [100,101] and under the operations-phase activities of ISO/SAE 21434 [25].

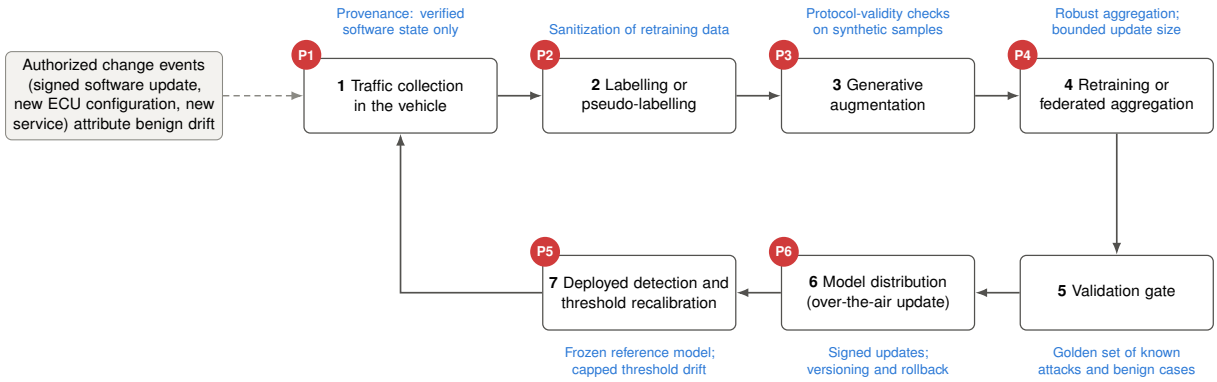


Figure 6. Retraining lifecycle of an adaptive IDS with poisoning entry points (P1–P6) and corresponding controls. Drift that coincides with an authorized change event is expected; drift outside such events is quarantined for review. In the reviewed corpus, adaptation is evaluated on benign drift only [17,83,84], and no primary study evaluates poisoning of its own update mechanism.

In the corpus, adaptation is evaluated on benign drift [17,83,84], and adversarial robustness is evaluated against evasion of fixed models [20,64,65]. No primary study evaluates poisoning of its own update mechanism. This gap matters most for software-defined vehicles, where frequent software updates make benign drift routine and therefore provide cover for gradual poisoning.

10. Open Challenges and Future Research Directions

10.1. Open Challenges

Table 9 summarizes the main open challenges and the corresponding research directions.

The reviewed literature shows clear progress in AI-based automotive IDSs, but several challenges still limit practical maturity. The main issue is not the absence of accurate models; many studies already report high detection performance under controlled conditions. The more important gap is whether these results remain valid across vehicles, protocols, datasets, attack variants, deployment locations, and lifecycle changes.

One major direction is cross-protocol and cross-layer detection. Automotive IDS research remains strongly shaped by CAN datasets, but modern vehicles increasingly combine classical buses with Ethernet, SOME/IP, diagnostics, gateways, cloud interfaces, and cooperative communication. Connected-vehicle communication further expands IDS requirements beyond traditional in-vehicle networks [103]. Future IDS designs should therefore connect in-vehicle evidence with service, gateway, edge, and cooperative-context evidence rather than treating each network layer as an isolated detection problem.

Lifecycle-aware updating is another unresolved challenge. Software-defined vehicles will receive updates after deployment, and normal traffic may change as features, services, and configurations evolve. These developments imply that an automotive IDS should not be viewed as a fixed classifier. Future work should define how IDS models are updated, validated, rolled back, monitored, and audited across the vehicle lifecycle, and how the update pipeline itself is protected against poisoning (Section 9.6).

Table 9. Open challenges and future research directions for automotive IDSs.

Open Challenge	Why It Matters	Future Research Direction
Cross-protocol detection	CAN-focused models may not generalize to Ethernet, SOME/IP, diagnostics, V2X, or mixed gateway traffic; 88.6% of primary studies are CAN-only.	Develop IDS designs that jointly analyze message-level, service-level, gateway-level, and cooperative-vehicle evidence.
Lifecycle-aware model updates	Vehicle software, traffic patterns, and benign behavior change after deployment.	Design controlled model-update pipelines with drift monitoring, poisoning safeguards, rollback, auditability, and software-update governance.
Fleet-level operation	Federated and distributed IDSs promise fleet-wide detection but are rarely evaluated under realistic connectivity and adversarial participants.	Report communication cost, convergence under dropout, and robustness to malicious vehicles.
Trustworthy AI evaluation	Accuracy alone does not establish reliability, explainability, or robustness.	Report uncertainty, explanation evidence, adversarial testing, calibration, false-positive burden, and safety-relevant failure modes.
Deployment evidence	Many IDS studies are evaluated offline without realistic hardware or timing constraints.	Measure latency, throughput, CPU, memory, energy, model size, and deployment location under realistic traffic loads.
Benchmark realism	Public datasets often cover limited vehicles, attacks, protocols, and driving contexts; all dataset contributions in the corpus are CAN-based.	Create shared benchmarks with realistic attack timing, class imbalance, cross-vehicle validation, and reproducible preprocessing.
Standard and regulatory alignment	IDS outputs may support cybersecurity monitoring, incident response, and software-defined vehicle operations.	Map IDS evidence to cybersecurity engineering, software-update management, AI-risk management, and safety assurance processes.

Trustworthy AI evaluation also needs stronger treatment. Future studies should therefore report not only detection metrics but also explanation evidence, uncertainty estimates, threshold sensitivity, calibration, adversarial robustness, and failure analysis.

Deployment evidence must also become more systematic. AUTOSAR has introduced intrusion-detection and security-event concepts into automotive software architecture, which shows that IDS functionality is moving closer to production-oriented engineering practice [104]. However, many academic IDS studies still report offline accuracy without latency, CPU, memory, energy, or model-size analysis. Future evaluations should explicitly state whether the IDS is intended for an ECU, gateway, domain controller, edge node, cloud service, or fleet-security platform. Future research should also clarify how IDS outputs interact with safety-relevant monitoring, escalation, and response processes.

Benchmark realism remains one of the strongest limitations. A useful benchmark should not only provide large numbers of labeled frames. It should document the vehicle

context, driving scenario, protocol layer, attack method, injection timing, class distribution, preprocessing, and recommended evaluation splits. It should also include difficult benign cases, because false positives often emerge from unusual but legitimate traffic. Cross-dataset and cross-vehicle tests should become routine when a study claims generalization. Without such evaluation, it is difficult to know whether a model has learned automotive intrusion behavior or simply memorized dataset-specific patterns [71].

10.2. Steps to Strengthen the Evidence Base

The unevenness identified in this review is largely a problem of evaluation design and reporting rather than of model capability. The following steps address it, grouped by the actors who can take them.

For study authors. (1) Use leakage-resistant splits, such as time-based, vehicle-held-out, or capture-held-out splits, instead of random frame-level splits, and state the split procedure; random splits place near-identical windows in both training and test sets and inflate performance [58–60]. (2) Report a standard unknown-attack protocol in which each attack family is held out in turn, giving the detection rate and false-positive rate for each held-out family. (3) Test across datasets and vehicles by training on one dataset and testing on another with compatible features, for example ROAD [19] or can-train-and-test [62], and report the resulting performance drop. (4) Report the false-positive rate at a fixed detection rate, or false alarms per hour of driving, alongside F1-score and AUC, because the base rate of attacks is low in real operation [58]. (5) Report per-message or per-window latency, throughput at realistic bus load, memory, model size, CPU load, and energy on named hardware, and state the intended deployment location. (6) Evaluate against adaptive attackers whose perturbations respect protocol timing, arbitration, and payload semantics [20,24,64,65]. (7) Re-evaluate after a simulated software update or the addition of a new service, and describe how retraining is controlled (Section 9.6). (8) Publish code, preprocessing scripts, split definitions, and trained models.

For dataset creators. (9) Capture CAN or CAN FD, Automotive Ethernet, SOME/IP, and diagnostic traffic simultaneously, with gateway-level ground truth and difficult benign cases such as diagnostic sessions and software updates. (10) Document each dataset with the vehicle, driving scenarios, attack-injection method and timing, class distribution, and recommended splits.

For venues and reviewers. (11) Adopt a minimum reporting checklist, such as Table 10, and maintain shared benchmarks with fixed splits so that results can be compared directly.

Table 10. Minimum evidence-reporting checklist for automotive IDS studies.

#	Item to Report	Why It Matters	Gap Addressed
1	Split procedure (time-, vehicle-, or capture-held-out)	Prevents train–test leakage	Section 8.3
2	Held-out attack-family results	Measures unknown-attack readiness	Section 7.4
3	Cross-dataset or cross-vehicle results	Measures generalization	Section 8
4	FPR at a fixed detection rate, or alarms per hour	Measures operational alert burden	Section 8.3
5	Latency, throughput, memory, CPU, and energy on named hardware	Measures deployability	Section 9.5
6	Intended deployment location	Puts resource figures in context	Section 2.1
7	Protocol-constrained adversarial evaluation	Measures robustness to adaptive attackers	Section 7.4
8	Post-update (benign-drift) and poisoning evaluation	Measures lifecycle stability	Section 9.6
9	For federated designs: communication cost, dropout, malicious clients	Measures fleet-level feasibility	Section 9.4
10	Code, preprocessing, split definitions, and dataset documentation	Allows reproduction and correct reuse	Section 9.2

Overall, future automotive IDS research should move from isolated model development toward evidence packages that combine realistic benchmarks, cross-protocol coverage, lifecycle-aware updating, trustworthy AI evaluation, and deployment-specific resource measurements. This direction is necessary for moving AI-based IDSs from promising laboratory prototypes toward systems that can support cybersecurity monitoring in connected, software-defined, and safety-relevant vehicle environments [78].

11. Conclusions

This systematic literature review examined AI-based intrusion detection systems for automotive cybersecurity across 44 primary studies, selected from 105 first-round reports after re-assessment against clarified eligibility criteria. The review considered in-vehicle, connected-vehicle, and software-defined vehicle environments, including CAN, CAN FD, SOME/IP, Automotive Ethernet, IVN, IoV, V2X, and hybrid network architectures. It organized the literature according to IDS families, attack types, datasets, evaluation practices, robustness evidence, deployment constraints, reproducibility, and trustworthiness.

The synthesis shows a measurable shift in research emphasis. Foundational entropy-, timing-, and physical-characteristic-based methods accounted for 58.3% of the detectors published up to 2019, but only 9.1% and 15.4% of those published in 2020–2022 and 2023–2025, respectively, while generative, multi-stage or federated, and adaptive designs grew from 8.3% to 31.6% of all studies (Section 5.2). These methods provide useful capabilities for detecting known attacks, modeling complex traffic patterns, supporting rare-attack learning, and handling evolving conditions. However, the evidence base remains uneven. Thirty-nine of the 44 primary studies (88.6%) address CAN only, every dataset and benchmark contribution is CAN-based, studies of datasets, benchmarks, and attacks against IDSs appear only from 2021 onward, and no primary study evaluates poisoning of an adaptive update mechanism or the communication cost of federated detection. Many studies still rely on closed-set attack assumptions, limited cross-dataset validation, and offline evaluation settings. High accuracy or F1-score alone is therefore insufficient to establish practical readiness.

Several gaps are especially important for future research. First, automotive IDS evaluation should better reflect hybrid vehicle architectures by covering CAN, Ethernet, SOME/IP, diagnostics, gateways, V2X, and software-defined functions together. Second, datasets should improve realism, documentation, attack diversity, class balance, and reproducibility. Third, evaluation should include false-positive and false-negative behavior, latency, CPU, memory, energy, model size, and deployment location. Fourth, IDS robustness should be tested against unknown attacks, distribution shift, adversarial manipulation, concept drift, poisoning of the retraining pipeline, and unrealistic synthetic data. Finally, interpretability, uncertainty, lifecycle-aware model updates, and alignment with automotive cybersecurity engineering should become standard parts of IDS assessment. Section 10.2 translates these gaps into concrete steps and a minimum reporting checklist. Overall, the review indicates that AI-based IDSs are promising but not yet uniformly deployment-ready. The strongest future direction is not a single model family, but a reproducible and deployment-aware IDS pipeline that combines realistic benchmarks, cross-protocol evidence, robust learning, explainable decisions, resource-aware implementation, and lifecycle controls. Such evidence is necessary for moving automotive IDS research from controlled laboratory demonstrations toward practical cybersecurity monitoring in connected, software-defined, and safety-relevant vehicle environments.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/vehicles8120315/s1, Table S1: Study-level coding of the 44 primary studies; Table S2: Re-assessment decisions for the 105 first-round reports; the PRISMA 2020 checklist; and the PRISMA 2020 for Abstracts checklist.

Funding: This research received no external funding.

Data Availability Statement: The data supporting this review, namely the study-level coding and the re-assessment decisions, are provided in Supplementary Tables S1 and S2. The review protocol is available on the Open Science Framework (Registration ID: u2c73).

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Search Strategy, Eligibility Criteria, and Codebook

Table A1. Search scope and study-selection criteria.

Category	Description
Search sources	IEEE Xplore, ACM Digital Library, ScienceDirect, SpringerLink, Wiley Online Library, Scopus, Web of Science, Google Scholar, and backward and forward snowballing.
Core search concepts	Automotive or vehicular context; IDS or anomaly detection; AI, ML, DL, generative, adaptive, federated, or adversarial methods; deployment or robustness constraints.
Inclusion criteria	(I1) Automotive, IVN, IoV, V2X, SDV, CAN, CAN FD, SOME/IP, Automotive Ethernet, OBD-II, or hybrid vehicle network context. (I2) Primary contribution of one of three types: proposing or evaluating an IDS, anomaly detector, or attack detector; introducing or benchmarking a dataset for automotive IDS evaluation; or evaluating attacks against an automotive IDS. (I3) Sufficient technical detail for coding. (I4) English; published January 2008 to July 2026; journal article, conference paper, or preprint.
Exclusion criteria	(E1) Secondary studies: surveys, reviews, and surveys of surveys. (E2) Methodological guidelines for conducting or reporting reviews. (E3) Standards, regulations, specifications, and frameworks. (E4) Generic enterprise, cloud, IoT, or network IDS studies and general ML methods without automotive evaluation. (E5) Automotive security studies without a detection contribution, such as attack demonstrations, cryptography, access control, authentication, firmware updates, secure boot, privacy mechanisms, or formal protocol analysis. (E6) Full text unavailable. Sources excluded under E1–E5 may be cited as background.
Final evidence base	After database search, snowballing, duplicate removal, screening, full-text eligibility assessment, and re-assessment against E1–E5, 44 primary studies were retained for synthesis.

Table A2. Final search strings as executed in each database, with filters and search dates.

Database	Final Search String (as Executed)	Filters	Search Date
IEEE Xplore	(“automotive” OR “vehicle” OR “in-vehicle” OR “CAN” OR “CAN FD” OR “SOME/IP” OR “Automotive Ethernet” OR “V2X” OR “IoV”) AND (“intrusion detection” OR “anomaly detection” OR “attack detection”) AND (“machine learning” OR “deep learning” OR “AI” OR “autoencoder” OR “GAN” OR “federated learning” OR “adversarial”)	English; journal and conference papers; 2008–2026	6 July 2026
ACM Digital Library	(“automotive” OR “vehicular” OR “in-vehicle network” OR “controller area network”) AND (“intrusion detection” OR “anomaly detection”) AND (“machine learning” OR “deep learning” OR “generative” OR “adaptive”)	English; proceedings and journal articles; 2008–2026	6 July 2026
ScienceDirect	(“automotive cybersecurity” OR “in-vehicle network” OR “connected vehicle”) AND (“intrusion detection” OR “anomaly detection”) AND (“machine learning” OR “deep learning” OR “artificial intelligence”)	English; article and review papers; 2008–2026	6 July 2026
SpringerLink	(“vehicle” OR “automotive” OR “CAN bus” OR “V2X”) AND (“intrusion detection” OR “attack detection”) AND (“machine learning” OR “deep learning” OR “adaptive IDS”)	English; conference and journal papers; 2008–2026	6 July 2026
Wiley Online Library	(“automotive” OR “vehicular network” OR “connected vehicle”) AND (“intrusion detection” OR “cybersecurity”) AND (“machine learning” OR “AI-based”)	English; journal papers; 2008–2026	6 July 2026

Table A2. *Cont.*

Database	Final Search String (as Executed)	Filters	Search Date
Scopus	TITLE-ABS-KEY((automotive OR vehicular OR "in-vehicle" OR CAN OR "SOME/IP" OR "Automotive Ethernet" OR V2X OR IoV) AND ("intrusion detection" OR "anomaly detection" OR "attack detection") AND ("machine learning" OR "deep learning" OR AI OR generative OR federated OR adversarial))	English; articles, conference papers, and reviews; 2008–2026	6 July 2026
Web of Science	TS=((automotive OR vehicular OR "in-vehicle network" OR CAN OR "Automotive Ethernet" OR V2X) AND ("intrusion detection" OR "anomaly detection") AND ("machine learning" OR "deep learning" OR "artificial intelligence"))	English; articles and proceedings; 2008–2026	6 July 2026
Google Scholar	"automotive intrusion detection" OR "CAN bus intrusion detection" OR "AI-based IDS in-vehicle network" OR "machine learning automotive cybersecurity"	English; relevance screening; 2008–2026	6 July 2026
Total database records retrieved: 512 (Figure 2)			
Snowballing	Backward and forward snowballing from selected surveys, benchmark papers, standards, and highly cited automotive IDS studies (42 records).	References and citing papers screened manually; 2008–2026	6 July 2026

Note. Unquoted acronyms such as CAN are matched case-insensitively by several databases and therefore also retrieve records containing the word "can"; such records were removed during title and abstract screening.

Table A3. Codebook for the study-level coding.

Field	Codes and Decision Rule
Contribution type	Detector: proposes or evaluates a detection method. Dataset/benchmark: the main contribution is a dataset or a comparative benchmark. Robustness evaluation: the main contribution is an attack against, or a resilience analysis of, an automotive IDS.
IDS family (one per detector)	Entropy-based; timing/frequency/sequence; physical-characteristic; traditional ML; deep learning; generative (GAN, VAE, or adversarial autoencoder at the core of detection); multi-stage/federated (staged pipeline or federated training is the central contribution); adaptive/drift-aware (model or threshold adaptation is the central contribution). When several mechanisms are combined, the family of the mechanism presented as the main contribution is coded.
Method group	Foundational: entropy, timing/frequency/sequence, physical-characteristic. Learning-based: traditional ML, deep learning. Advanced learning: generative, multi-stage/federated, adaptive/drift-aware. Evaluation infrastructure: dataset/benchmark, robustness evaluation.
Network environment	CAN; Automotive Ethernet; SOME/IP; CAN + Automotive Ethernet; CAN + external IoV; connected vehicle. Coded from the traffic on which detection is evaluated.
Publication period	2011–2019; 2020–2022; 2023–2025 (by year of the cited version).

Appendix B. Supporting Review Matrices

Table A4. Research-question traceability matrix. References are representative primary studies; background sources are given in parentheses where they frame the analysis.

RQ	Primary Analytical Focus	Extracted Evidence	Synthesis Method	References
RQ1	Automotive IDS approaches and methodological taxonomy	IDS family, model architecture, learning paradigm, detection evidence, and target network	Taxonomic synthesis and frequency-based descriptive analysis	[13,21,30,31,36,77,79,83]
RQ2	Rapid detection of known attacks	Attack type, detection mechanism, reported performance, computational complexity, and latency	Structured qualitative performance comparison	[21,22,30–32,36,66,67,72]
RQ3	Unknown, evolving, and shifted attacks	Open-set detection, anomaly modeling, adaptation, concept drift, uncertainty, and synthetic attack generation	Conceptual and taxonomy-driven synthesis	[15,17,71,76–78,83,84]

Table A4. *Cont.*

RQ	Primary Analytical Focus	Extracted Evidence	Synthesis Method	References
RQ4	Datasets, features, attacks, and metrics	Dataset source, network protocol, attack classes, input features, evaluation metrics, and validation design	Descriptive and comparative evidence synthesis	[19,34,35,61–63]
RQ5	Comparison of major IDS designs	Detection stage, adaptation mechanism, generative component, robustness strategy, and architectural limitations	Multidimensional comparative synthesis	[15,77,79,80,82–84]
RQ6	Reproducibility and benchmarking	Dataset and code availability, preprocessing, validation protocol, baseline selection, and cross-dataset testing	Cross-study synthesis of reported evaluation practice	[19,61–63] ([58–60])
RQ7	Deployment and industrial adoption	Latency, memory, CPU usage, energy consumption, false-positive burden, deployment location, and embedded feasibility	Deployment-oriented evidence synthesis	[21,23,66,72,79,80] ([25,100,104])
RQ8	Interpretability, robustness, and trustworthiness	Explanation methods, uncertainty, adversarial resilience, model drift, reliability, and safety-related evidence	Risk-oriented qualitative synthesis	[20,24,34,64,65,81] ([90,91,102])

Table A5. Evaluation metrics and deployment relevance for automotive IDSs.

Metric Category	Common Metrics	Deployment Relevance	Limitation if Used Alone	References
Detection accuracy	Accuracy, precision, recall, F1-score, and AUC.	Measures classification quality under the tested dataset and attack setup.	A high F1-score does not prove real-time feasibility, robustness, or low operational alert burden.	[12,70,73,75]
False-alarm behavior	False-positive rate, false-negative rate, alert rate, and missed-detection rate.	Determines whether the IDS can operate without overwhelming monitoring systems or missing safety-relevant attacks.	Often depends on dataset imbalance, threshold selection, and attack prevalence.	[21,32,36]
Timing performance	Detection latency, inference time, throughput, and response time.	Critical for real-time CAN, Ethernet, gateway, and ECU-level deployment.	Frequently omitted or measured on non-automotive hardware.	[23,66,72,80]
Resource usage	CPU load, memory footprint, model size, and energy consumption.	Determines feasibility for embedded ECUs, gateways, and edge devices.	Difficult to compare unless hardware and implementation details are reported.	[66,72,80]
Robustness and generalization	Cross-dataset accuracy, adversarial robustness, drift sensitivity, and unknown-attack detection.	Indicates whether the IDS can remain reliable under changing vehicles, attacks, and traffic distributions.	Still under-reported compared with standard classification metrics.	[15,20,62,64,65]
Deployment-context evidence	ECU, gateway, edge/cloud, fleet monitoring, IDS manager, software update, and lifecycle monitoring.	Connects model evaluation to practical automotive cybersecurity engineering.	Deployment-context evidence is scattered and often discussed conceptually.	[79,81] ([25,100,101,104])

Table A6. Deployment-readiness evidence categories.

Deployment Context	Required Evidence	Typical Constraint	Use in Synthesis	References
ECU-level IDS	Latency, memory, CPU, model size	Strict timing and resource limits	Used to assess embedded feasibility.	[21,23,72]
Gateway IDS	Throughput, multi-protocol visibility, latency	Multiple network segments	Used to assess practical monitoring scope.	[79,80,82]
Edge/cloud-assisted IDS	Connectivity, privacy, storage, response delay, communication cost	Network dependence	Used to assess fleet-level monitoring feasibility.	[79–81]

Note. References are representative examples from the primary studies. The complete study-level coding is provided in Appendix C, Table A8.

Table A7. Evidence-based research gaps and future research directions.

Research Gap	Evidence From Reviewed Studies	Consequence	Future Research Direction	References
Limited cross-protocol validation	39 of 44 primary studies (88.6%) are CAN-only; Automotive Ethernet, SOME/IP, hybrid, IoV, and connected-vehicle settings appear in one study each.	IDS performance may not generalize to hybrid and software-defined vehicle architectures.	Develop cross-protocol benchmarks covering CAN, CAN FD, Automotive Ethernet, SOME/IP, diagnostics, gateways, and V2X/IoV interactions.	[74,80–82,84]
Dataset realism and benchmark maturity	All four dataset and benchmark contributions are CAN-based; many datasets remain narrow in vehicle diversity, attack diversity, protocol scope, and driving context.	High reported performance may reflect dataset-specific patterns rather than robust vehicle-level intrusion detection.	Create better documented, multi-vehicle, multi-protocol, and scenario-rich datasets with realistic benign traffic and attack traces.	[19,61–63]
Weak unknown-attack and open-set evaluation	Evaluation predominantly targets known attacks, with limited assessment of unknown attacks and distribution shifts.	IDS models may perform well on benchmark attacks but fail against novel or evolving intrusions.	Use open-set, cross-dataset, leave-one-attack-out, and unknown-attack evaluation protocols as standard practice.	[15,71,77]
Limited deployment-readiness evidence	Latency, throughput, CPU use, memory use, model size, and hardware constraints are not reported consistently.	It remains difficult to judge whether IDS models can operate on ECUs, gateways, or real-time monitoring units.	Report runtime, model size, memory, CPU, throughput, deployment location, and hardware configuration with detection metrics.	[23,66,72]
Insufficient adversarial robustness testing	Four robustness evaluations, all on CAN; adversarial evasion and poisoning are less common than accuracy-based evaluation.	Closed-set performance may overestimate security when attackers adapt to the IDS model.	Evaluate IDS models against evasion, poisoning, backdoor, perturbation, and adaptive attack scenarios.	[20,24,64,65]
Limited interpretability	Explainability is increasingly discussed, but few IDS studies provide decision explanations useful for analysts.	Operators may not understand why an alert was triggered or how to connect it to vehicle-level response.	Integrate explainable IDS methods that identify influential message IDs, payload patterns, timing, signals, or protocol-level causes.	[34,81] ([90])
Lifecycle and software-update alignment	Adaptation is evaluated on benign drift only; no primary study evaluates poisoning of its update mechanism.	An IDS that works at training time may degrade, or be poisoned, after software updates or changed traffic behavior.	Develop lifecycle-aware IDS pipelines with drift monitoring, poisoning safeguards, update validation, and standards compliance.	[17,83,84] ([100,101])
Fleet-level operation	Two federated primary studies; none reports communication cost, dropout tolerance, or malicious-client robustness.	Fleet-wide detection claims remain untested under realistic connectivity and adversaries.	Report communication cost, convergence under dropout, and robustness to malicious vehicles.	[79,81]
Reproducibility and comparability	Methodological differences in datasets, preprocessing, validation, and metrics limit direct comparison; 25.0% of studies are cited as preprints.	Model rankings may be misleading when evaluation settings are not standardized.	Publish preprocessing steps, code, dataset splits, attack definitions, and evaluation protocols.	[61,63] ([58])

Appendix C. Study-Level Coding

Table A8. Study-level coding of the 44 primary studies, ordered by publication year. Supplementary Table S1 gives the same coding together with full titles.

ID	Ref.	Year	Contribution	IDS Family	Network	Main Detection Evidence or Focus
S01	[31]	2011	Detector	Entropy-based	CAN	Entropy of the message stream
S02	[13]	2016	Detector	Deep learning	CAN	LSTM prediction of data-field sequences
S03	[21]	2016	Detector	Physical-characteristic	CAN	ECU clock-skew fingerprinting
S04	[36]	2016	Detector	Timing/frequency/sequence	CAN	Message inter-arrival intervals
S05	[75]	2016	Detector	Deep learning	CAN	Deep neural network on payload features
S06	[32]	2017	Detector	Timing/frequency/sequence	CAN	Identifier transition sequences

Table A8. *Cont.*

ID	Ref.	Year	Contribution	IDS Family	Network	Main Detection Evidence or Focus
S07	[39]	2018	Detector	Traditional ML	CAN	Supervised classifiers on frame features
S08	[30]	2018	Detector	Entropy-based	CAN	Entropy analysis of CAN traffic
S09	[68]	2018	Detector	Traditional ML	CAN	Shape of decoded CAN data
S10	[22]	2019	Detector	Timing/frequency/sequence	CAN	Message frequencies across driving modes
S11	[23]	2019	Detector	Physical-characteristic	CAN	Bit-time fingerprint and attacker identification
S12	[77]	2019	Detector	Generative	CAN	GAN discriminators for known and unknown attacks
S13	[70]	2020	Detector	Deep learning	CAN	LSTM on frame sequences
S14	[72]	2020	Detector	Deep learning	CAN	Recurrent-autoencoder reconstruction error for embedded systems
S15	[11]	2021	Detector	Traditional ML	CAN	Machine-learning classifiers
S16	[24]	2021	Robustness evaluation	–	CAN	Evasion of voltage-based detection
S17	[61]	2021	Dataset/ benchmark	–	CAN	Benchmark of time-based detectors
S18	[37]	2021	Detector	Physical-characteristic	CAN	Deep features of voltage signals; intrusion localization
S19	[71]	2021	Detector	Deep learning	CAN	Self-supervised learning with noised pseudo-normal data
S20	[76]	2021	Detector	Deep learning	CAN	LSTM-autoencoder reconstruction
S21	[15]	2022	Detector	Generative	CAN	Auxiliary-classifier GAN with out-of-distribution detection
S22	[69]	2022	Detector	Deep learning	CAN	BERT language model on identifier sequences
S23	[80]	2022	Detector	Multi-stage/federated	CAN + external IoV	Multi-tiered hybrid detection for intra-vehicle and external networks
S24	[78]	2022	Detector	Generative	CAN	Semi-supervised convolutional adversarial autoencoder
S25	[74]	2022	Detector	Deep learning	Automotive Ethernet	Unsupervised detection on AVTP streams
S26	[12]	2023	Detector	Traditional ML	CAN	Statistical characteristics of attacks
S27	[62]	2023	Dataset/ benchmark	–	CAN	Curated multi-vehicle CAN dataset
S28	[33]	2023	Detector	Deep learning	CAN	Payload-based detection
S29	[64]	2023	Robustness evaluation	–	CAN	Resilience of ML-based IDSs to adversarial attacks
S30	[82]	2023	Detector	Multi-stage/federated	SOME/IP	Multi-layer detection for SOME/IP traffic
S31	[34]	2023	Detector	Deep learning	CAN	Signal-aware detection with explanations
S32	[35]	2023	Detector	Deep learning	CAN	Signal-level detection at multiple time scales
S33	[19]	2024	Dataset/ benchmark	–	CAN	Dataset with ambient driving and stealthy attacks (ROAD)
S34	[79]	2024	Detector	Multi-stage/federated	CAN	Multi-stage detection with hierarchical federated learning
S35	[67]	2024	Detector	Timing/frequency/sequence	CAN	Frequency anomalies with arbitration-based defence

Table A8. *Cont.*

ID	Ref.	Year	Contribution	IDS Family	Network	Main Detection Evidence or Focus
S36	[65]	2024	Robustness evaluation	–	CAN	Adversarial attacks on in-vehicle IDSs
S37	[83]	2024	Detector	Adaptive/drift-aware	CAN	DNN–VAE adaptive sample generation
S38	[63]	2024	Dataset/ benchmark	–	CAN	Benchmark of unsupervised online detectors for masquerade attacks
S39	[17]	2025	Detector	Adaptive/drift-aware	CAN	Adaptive autoencoder with a single threshold
S40	[20]	2025	Robustness evaluation	–	CAN	Adversarial manipulation of automotive IDSs
S41	[66]	2025	Detector	Entropy-based	CAN	Lightweight entropy-based algorithm
S42	[73]	2025	Detector	Deep learning	CAN	Deep learning detection for autonomous vehicles
S43	[84]	2025	Detector	Adaptive/drift-aware	CAN + Automotive Ethernet	Protocol-aware adaptive detection with adversarial-robustness evaluation
S44	[81]	2025	Detector	Multi-stage/federated	Connected vehicle	Federated detection with explainable AI

Table A9. First-round reports excluded at re-assessment as non-primary sources. These sources remain cited for background, methodology, or positioning.

Exclusion Reason	n	References
E1: Secondary reviews and surveys	17	[4–10,14,16,18,29,42–47]
E2: Methodological guidelines	10	[48–57]
E3: Standards, specifications, and frameworks	10	[25–28,99–104]
E4: Generic non-automotive works	12	[38,58–60,87–91,96–98]
E5: Automotive attack demonstrations without IDS	7	[1–3,40,41,85,86]
E5: Automotive security mechanisms or formal analysis without IDS	5	[92–95,105]
Total	61	

References

1. Koscher, K.; Czeskis, A.; Roesner, F.; Patel, S.; Kohno, T.; Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; et al. Experimental security analysis of a modern automobile. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 447–462. https://doi.org/10.1109/SP.2010.34.
2. Checkoway, S.; McCoy, D.; Kantor, B.; Anderson, D.; Shacham, H.; Savage, S.; Koscher, K.; Czeskis, A.; Roesner, F.; Kohno, T. Comprehensive experimental analyses of automotive attack surfaces. In Proceedings of the 20th USENIX Security Symposium, San Francisco, CA, USA, 8–12 August 2011; pp. 77–92.
3. Miller, C.; Valasek, C. Remote exploitation of an unaltered passenger vehicle. In Proceedings of Black Hat USA, Las Vegas, NV, USA, 1–4 August 2015.
4. Aliwa, E.; Rana, O.; Perera, C.; Burnap, P. Cyberattacks and countermeasures for in-vehicle networks. *ACM Comput. Surv.* **2021**, *54*, 1–37. https://doi.org/10.1145/3431233.
5. Wu, W.; Li, R.; Xie, G.; An, J.; Bai, Y.; Zhou, J.; Li, K. A survey of intrusion detection for in-vehicle networks. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 919–933. https://doi.org/10.1109/TITS.2019.2908074.
6. Al-Jarrah, O.Y.; Maple, C.; Dianati, M.; Oxtoby, D.; Mouzakitis, A. Intrusion detection systems for intra-vehicle networks: A review. *IEEE Access* **2019**, *7*, 21266–21289. https://doi.org/10.1109/ACCESS.2019.2894183.
7. Karopoulos, G.; Kambourakis, G.; Chatzoglou, E.; Hernández-Ramos, J.L.; Kouliaridis, V. Demystifying in-vehicle intrusion detection systems: A survey of surveys and a meta-taxonomy. *Electronics* **2022**, *11*, 1072. https://doi.org/10.3390/electronics11071072.
8. Lokman, S.F.; Othman, A.T.; Abu-Bakar, M.H. Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 184. https://doi.org/10.1186/s13638-019-1484-3.
9. Loukas, G.; Karapistoli, E.; Panaousis, E.; Sarigiannidis, P.; Bezemskij, A.; Vuong, T. A taxonomy and survey of cyber-physical intrusion detection approaches for vehicles. *Ad. Hoc Netw.* **2019**, *84*, 124–147. https://doi.org/10.1016/j.adhoc.2018.10.002.

10. Rajapaksha, S.; Kalutarage, H.; Al-Kadri, M.O.; Petrovski, A.; Madzudzo, G.; Cheah, M. AI-based intrusion detection systems for in-vehicle networks: A survey. *ACM Comput. Surv.* **2023**, *55*, 237. https://doi.org/10.1145/3570954.
11. Alfardus, A.; Rawat, D.B. Intrusion detection system for CAN bus in-vehicle network based on machine learning algorithms. In Proceedings of the IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference, New York, NY, USA, 1–4 December 2021; pp. 944–949. https://doi.org/10.1109/UEMCON53757.2021.9666745.
12. Khan, J.; Lim, D.W.; Kim, Y.S. Intrusion detection system CAN-bus in-vehicle networks based on the statistical characteristics of attacks. *Sensors* **2023**, *23*, 3235. https://doi.org/10.3390/s23073554.
13. Taylor, A.; Leblanc, S.; Japkowicz, N. Anomaly detection in automobile control network data with long short-term memory networks. In Proceedings of the IEEE International Conference on Data Science and Advanced Analytics, Montreal, QC, Canada, 17–19 October 2016; pp. 130–139. https://doi.org/10.1109/DSAA.2016.20.
14. Luo, F.; Wang, J.; Zhang, X.; Jiang, Y.; Li, Z.; Luo, C. In-vehicle network intrusion detection systems: A systematic survey of deep learning-based approaches. *PeerJ Comput. Sci.* **2023**, *9*, e1648. https://doi.org/10.7717/peerj-cs.1648.
15. Zhao, Q.; Chen, M.; Gu, Z.; Luan, S.; Zeng, H.; Chakraborty, S. CAN bus intrusion detection based on auxiliary classifier GAN and out-of-distribution detection. *ACM Trans. Embed. Comput. Syst.* **2022**, *21*, 45. https://doi.org/10.1145/3540198.
16. Li, J.; Ersotelos, N.; Tsompanas, M.-A.; Epiphaniou, G. Challenges and opportunities of synthetic data generation for machine learning-based intrusion detection systems in in-vehicle networks. *Veh. Commun.* **2026**, *58*, 100998. https://doi.org/10.1016/j.vehcom.2025.100998.
17. Kim, D.; Im, H.; Lee, S. Adaptive autoencoder-based intrusion detection system with single threshold for CAN networks. *Sensors* **2025**, *25*, 4174. https://doi.org/10.3390/s25134174.
18. Lee, S.; Choi, W.; Kim, I.; Lee, G.; Lee, D.H. A comprehensive analysis of datasets for automotive intrusion detection systems. *Comput. Mater. Contin.* **2023**, *76*, 3413–3442. https://doi.org/10.32604/cmc.2023.039583.
19. Verma, M.E.; Bridges, R.A.; Iannacone, M.D.; Hollifield, S.C.; Moriano, P.; Hespeler, S.C.; Kay, B.; Combs, F.L. A comprehensive guide to CAN IDS data and introduction of the ROAD dataset. *PLoS ONE* **2024**, *19*, e0296879. https://doi.org/10.1371/journal.pone.0296879.
20. Longari, S.; Cerracchio, P.; Carminati, M.; Zanero, S. Assessing the resilience of automotive intrusion detection systems to adversarial manipulation. *ACM Trans. Cyber-Phys. Syst.* **2025**, *9*, 31. https://doi.org/10.1145/3737294.
21. Cho, K.-T.; Shin, K.G. Fingerprinting electronic control units for vehicle intrusion detection. In Proceedings of the 25th USENIX Security Symposium, Austin, TX, USA, 10–12 August 2016; pp. 911–927.
22. Young, C.; Zambreno, J.; Olufowobi, H.; Bloom, G. Automotive intrusion detection based on constant CAN message frequencies across vehicle driving modes. In *Proceedings of the ACM Workshop on Automotive Cybersecurity*; Association for Computing Machinery (ACM): New York, NY, USA, 2019; pp. 9–14. https://doi.org/10.1145/3309171.3309179.
23. Zhou, J.; Joshi, P.; Zeng, H.; Li, R. BTMonitor: Bit-time-based intrusion detection and attacker identification in controller area network. *ACM Trans. Embed. Comput. Syst.* **2019**, *18*, 117. https://doi.org/10.1145/3362034.
24. Bhatia, R.; Kumar, V.; Serag, K.; Celik, Z.B.; Payer, M.; Xu, D. Evading voltage-based intrusion detection on automotive CAN. In Proceedings of the Network and Distributed System Security Symposium, Online, 21–25 February 2021. https://doi.org/10.14722/ndss.2021.23013.
25. *ISO/SAE 21434:2021*; Road Vehicles—Cybersecurity Engineering. International Organization for Standardization: Geneva, Switzerland, 2021.
26. United Nations Economic Commission for Europe. *UN Regulation No. 155: Uniform Provisions Concerning the Approval of Vehicles with Regard to Cyber Security and Cyber Security Management System*; United Nations Economic Commission for Europe: Geneva, Switzerland, 2021.
27. *ISO 11898-1:2024*; Road Vehicles—Controller Area Network (CAN)—Part 1: Data Link Layer and Physical Signalling. International Organization for Standardization: Geneva, Switzerland, 2024.
28. AUTOSAR. SOME/IP Protocol Specification. AUTOSAR Foundation Release R22-11, 2022. Available online: https://www.autosar.org/fileadmin/standards/R22-11/FO/AUTOSAR_PRS_SOMEIPProtocol.pdf (accessed on 22 November 2026).
29. Douss, A.B.C.; Abassi, R.; Sauveron, D. State-of-the-art survey of in-vehicle protocols and automotive Ethernet security and vulnerabilities. *Math. Biosci. Eng.* **2023**, *20*, 17057–17095. https://doi.org/10.3934/mbe.2023761.
30. Wang, Q.; Lu, Z.; Qu, G. An entropy analysis based intrusion detection system for controller area network in vehicles. In Proceedings of the 2018 31st IEEE International System-on-Chip Conference (SOCC), Arlington, VA, USA, 4–7 September 2018; pp. 90–95. https://doi.org/10.1109/SOCC.2018.8618564.
31. Müter, M.; Asaj, N. Entropy-based anomaly detection for in-vehicle networks. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 1110–1115. https://doi.org/10.1109/IVS.2011.5940552.
32. Marchetti, M.; Stabili, D. Anomaly detection of CAN bus messages through analysis of ID sequences. In Proceedings of the IEEE Intelligent Vehicles Symposium, Los Angeles, CA, USA, 11–14 June 2017; pp. 1577–1583. https://doi.org/10.1109/IVS.2017.7995934.

33. Longari, S.; Pozzoli, C.A.; Nichelini, A.; Carminati, M.; Zanero, S. CANdito: Improving payload-based detection of attacks on controller area networks. In Proceedings of the 7th International Symposium on Cyber Security, Cryptology, and Machine Learning (CSCML 2023), Be'er Sheva, Israel, 29–30 June 2023; pp. 135–150. https://doi.org/10.1007/978-3-031-34671-2_10.
34. Jeong, S.; Lee, S.; Lee, H.; Kim, H.K. X-CANIDS: Signal-aware explainable intrusion detection system for Controller Area Network-based in-vehicle network. *arXiv* **2023**, arXiv:2303.12278. https://doi.org/10.48550/arXiv.2303.12278.
35. Shahriar, M.H.; Xiao, Y.; Moriano, P.; Lou, W.; Hou, Y.T. CANShield: Deep learning-based intrusion detection framework for controller area networks at the signal-level. *IEEE Internet Things J.* **2023**, *10*, 22111–22127. https://doi.org/10.1109/JIOT.2023.3303271.
36. Song, H.M.; Kim, H.R.; Kim, H.K. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In Proceedings of the International Conference on Information Networking, Kota Kinabalu, Malaysia, 13–15 January 2016; pp. 63–68. https://doi.org/10.1109/ICOIN.2016.7427089.
37. Levy, E.; Shabtai, A.; Groza, B.; Murvay, P.-S.; Elovici, Y. CAN-LOC: Spoofing detection and physical intrusion localization on an in-vehicle CAN bus based on deep features of voltage signals. *arXiv* **2021**, arXiv:2106.07895. https://doi.org/10.48550/arXiv.2106.07895.
38. García-Teodoro, P.; Díaz-Verdejo, J.; Maciá-Fernández, G.; Vázquez, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Comput. Secur.* **2009**, *28*, 18–28. https://doi.org/10.1016/j.cose.2008.08.003.
39. Alshammari, A.; Zohdy, M.A.; Debnath, D.; Corser, G. Classification approach for intrusion detection in vehicle systems. *Wirel. Eng. Technol.* **2018**, *9*, 79–94. https://doi.org/10.4236/wet.2018.94007.
40. Woo, S.; Jo, H.J.; Lee, D.H. A practical wireless attack on the connected car and security protocol for in-vehicle CAN. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 993–1006. https://doi.org/10.1109/TITS.2014.2351612.
41. Hoppe, T.; Kiltz, S.; Dittmann, J. Security threats to automotive CAN networks: Practical examples and selected short-term countermeasures. In *Computer Safety, Reliability, and Security; Lecture Notes in Computer Science*; Springer: Berlin, Germany, 2008; Volume 5219, pp. 235–248. https://doi.org/10.1007/978-3-540-87698-4_21.
42. Khan, J.A.; Khan, M.A.; Saeed, N.; Cayrel, P.L.; Hahn, C. Intrusion detection systems for in-vehicle networks: Protocols, applications, and challenges. *IEEE Access* **2025**, *13*, 215219–215250. https://doi.org/10.1109/ACCESS.2025.3645058.
43. Althunayyan, M.; Javed, A.; Rana, O.F. A survey of learning-based intrusion detection systems for in-vehicle networks. *Comput. Netw.* **2026**, *277*, 112031. https://doi.org/10.1016/j.comnet.2026.112031.
44. Ajibuwa, O.; Hamdaoui, B.; Yavuz, A.A. A survey on AI/ML-driven intrusion and misbehavior detection in networked autonomous systems: Techniques, challenges and opportunities. *arXiv* **2023**, arXiv:2305.05040. https://doi.org/10.48550/arXiv.2305.05040.
45. Mahmoudi, I.; Boubiche, D.E.; Athmani, S.; Toral-Cruz, H.; Chan-Puc, F.I. Toward generative AI-based intrusion detection systems for the Internet of Vehicles (IoV). *Future Internet* **2025**, *17*, 310. https://doi.org/10.3390/fi17070310.
46. Hasrouny, H.; Samhat, A.E.; Bassil, C.; Laouiti, A. VANet security challenges and solutions: A survey. *Veh. Commun.* **2017**, *7*, 7–20. https://doi.org/10.1016/j.vehcom.2017.01.002.
47. van der Heijden, R.W.; Dietzel, S.; Leinmüller, T.; Kargl, F. Survey on misbehavior detection in cooperative intelligent transportation systems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 779–811. https://doi.org/10.1109/COMST.2018.2873088.
48. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering, Version 2.3*; Keele University: Staffordshire, UK; Durham University: Durham, UK, 2007.
49. Wohlin, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; pp. 1–10. https://doi.org/10.1145/2601248.2601268.
50. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, n71. https://doi.org/10.1136/bmj.n71.
51. Brereton, P.; Kitchenham, B.A.; Budgen, D.; Turner, M.; Khalil, M. Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* **2007**, *80*, 571–583. https://doi.org/10.1016/j.jss.2006.07.009.
52. Kitchenham, B.; Brereton, O.P.; Budgen, D.; Turner, M.; Bailey, J.; Linkman, S. Systematic literature reviews in software engineering—A systematic literature review. *Inf. Softw. Technol.* **2009**, *51*, 7–15. https://doi.org/10.1016/j.infsof.2008.09.009.
53. Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. Systematic mapping studies in software engineering. In Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, Bari, Italy, 26–27 June 2008; pp. 68–77. https://doi.org/10.14236/ewic/EASE2008.8.
54. Okoli, C. A guide to conducting a standalone systematic literature review. *Commun. Assoc. Inf. Syst.* **2015**, *37*, 879–910. https://doi.org/10.17705/1CAIS.03743.
55. Cooper, H.; Hedges, L.V.; Valentine, J.C. (Eds.) *The Handbook of Research Synthesis and Meta-Analysis*, 3rd ed.; Russell Sage Foundation: New York, NY, USA, 2019. https://doi.org/10.7758/9781610448864.

56. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G.; The PRISMA Group. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Med.* **2009**, *6*, e1000097. https://doi.org/10.1371/journal.pmed.1000097.
57. Higgins, J.P.T.; Thomas, J.; Chandler, J.; Cumpston, M.; Li, T.; Page, M.J.; Welch, V.A. (Eds.) *Cochrane Handbook for Systematic Reviews of Interventions*, 2nd ed.; Wiley: Chichester, UK, 2019. https://doi.org/10.1002/9781119536604.
58. Arp, D.; Quiring, E.; Pendlebury, F.; Warnecke, A.; Pierazzi, F.; Wressnegger, C.; Cavallaro, L.; Rieck, K. Dos and don'ts of machine learning in computer security. In Proceedings of the 31st USENIX Security Symposium, Boston, MA, USA, 10–12 August 2022; pp. 3971–3988.
59. Sommer, R.; Paxson, V. Outside the closed world: On using machine learning for network intrusion detection. In Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 305–316. https://doi.org/10.1109/SP.2010.25.
60. Pendlebury, F.; Pierazzi, F.; Jordaney, R.; Kinder, J.; Cavallaro, L. TESSERACT: Eliminating experimental bias in malware classification across space and time. In Proceedings of the 28th USENIX Security Symposium, Santa Clara, CA, USA, 14–16 August 2019; pp. 729–746. https://doi.org/10.48550/arXiv.1807.07838.
61. Blevins, D.H.; Moriano, P.; Bridges, R.A.; Verma, M.E.; Iannacone, M.D.; Hollifield, S.C. Time-based CAN intrusion detection benchmark. In Proceedings of the Workshop on Automotive and Autonomous Vehicle Security, Virtual, 25 February 2021. https://doi.org/10.14722/autosec.2021.23013.
62. Lampe, B.; Meng, W. Can-train-and-test: A curated CAN dataset for automotive intrusion detection. *arXiv* **2023**, arXiv:2308.04972. https://doi.org/10.48550/arXiv.2308.04972.
63. Moriano, P.; Hespeler, S.C.; Li, M.; Bridges, R.A. Benchmarking unsupervised online IDS for masquerade attacks in CAN. *arXiv* **2024**, arXiv:2406.13778. https://doi.org/10.48550/arXiv.2406.13778.
64. Zenden, I.; Wang, H.; Iacovazzi, A.; Vahidi, A.; Blom, R.; Raza, S. On the resilience of machine learning-based IDS for automotive networks. *arXiv* **2023**, arXiv:2306.14782. https://doi.org/10.48550/arXiv.2306.14782.
65. Aloraini, F.; Althunayyan, M.; Javed, A.; Rana, O. Adversarial attacks on intrusion detection systems in in-vehicle networks of connected and autonomous vehicles. *Sensors* **2024**, *24*, 3848. https://doi.org/10.3390/s24123848.
66. Chen, J.; Wang, Z. Design and validation of a lightweight entropy-based intrusion detection algorithm for automotive CANs. *World Electr. Veh. J.* **2025**, *16*, 334. https://doi.org/10.3390/wevj16060334.
67. Dong, Q.; Song, Z.; Shao, H. Intrusion detection and defence for CAN bus through frequency anomaly analysis and arbitration mechanism. *Electron. Lett.* **2024**, *60*, e13112. https://doi.org/10.1049/ell2.13112.
68. Tyree, Z.; Bridges, R.A.; Combs, F.L.; Moore, M.R. Exploiting the shape of CAN data for in-vehicle intrusion detection. *arXiv* **2018**, arXiv:1808.10840. https://doi.org/10.48550/arXiv.1808.10840.
69. Alkhatib, N.; Mushtaq, M.; Ghauch, H.; Danger, J.-L. CAN-BERT do it? Controller Area Network Intrusion Detection System based on BERT Language Model. *arXiv* **2022**, arXiv:2210.09439. https://doi.org/10.48550/arXiv.2210.09439.
70. Hossain, M.D.; Inoue, H.; Ochiai, H.; Fall, D.; Kadobayashi, Y. LSTM-based intrusion detection system for in-vehicle CAN bus communications. *IEEE Access* **2020**, *8*, 185489–185502. https://doi.org/10.1109/ACCESS.2020.3029307.
71. Song, H.M.; Kim, H.K. Self-supervised anomaly detection for in-vehicle network using noised pseudo normal data. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1098–1108. https://doi.org/10.1109/TVT.2021.3051026.
72. Kukkala, V.; Thiruloga, S.V.; Pasricha, S. INDRA: Intrusion detection using recurrent autoencoders in automotive embedded systems. *arXiv* **2020**, arXiv:2007.08795. https://doi.org/10.48550/arXiv.2007.08795.
73. Rai, R.; Sharma, P.; Grover, J.; Pareek, A. Securing the CAN bus using deep learning for intrusion detection in autonomous vehicles. *Sci. Rep.* **2025**, *15*, 16811. https://doi.org/10.1038/s41598-025-98433-x.
74. Alkhatib, N.; Mushtaq, M.; Ghauch, H.; Danger, J.-L. Unsupervised network intrusion detection system for AVTP in automotive Ethernet networks. *arXiv* **2022**, arXiv:2202.00045. https://doi.org/10.48550/arXiv.2202.00045.
75. Kang, M.-J.; Kang, J.-W. Intrusion detection system using deep neural network for in-vehicle network security. *PLoS ONE* **2016**, *11*, e0155781. https://doi.org/10.1371/journal.pone.0155781.
76. Longari, S.; Valcarcel, D.H.N.; Zago, M.; Carminati, M.; Zanero, S. CANnolo: An anomaly detection system based on LSTM autoencoders for Controller Area Network. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 1913–1924. https://doi.org/10.1109/TNSM.2020.3038991.
77. Seo, E.; Song, H.M.; Kim, H.K. GIDS: GAN based intrusion detection system for in-vehicle network. *arXiv* **2019**, arXiv:1907.07377. https://doi.org/10.48550/arXiv.1907.07377.
78. Hoang, T.-N.; Kim, D. Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *arXiv* **2022**, arXiv:2204.01193. https://doi.org/10.48550/arXiv.2204.01193.
79. Althunayyan, M.; Javed, A.; Rana, O. A robust multi-stage intrusion detection system for in-vehicle network security using hierarchical federated learning. *Veh. Commun.* **2024**, *50*, 100837. https://doi.org/10.1016/j.vehcom.2024.100837.
80. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A multi-tiered hybrid intrusion detection system for Internet of Vehicles. *IEEE Internet Things J.* **2022**, *9*, 616–632. https://doi.org/10.1109/JIOT.2021.3084796.

81. Taheri, R.; Jafari, R.; Gegov, A.; Arabikhan, F.; Ichtev, A. Explainable AI for federated learning-based intrusion detection systems in connected vehicles. *Electronics* **2025**, *14*, 4508. https://doi.org/10.3390/electronics14224508.
82. Luo, F.; Yang, Z.; Zhang, Z.; Wang, Z.; Wang, B.; Wu, M. A multi-layer intrusion detection system for SOME/IP-based in-vehicle network. *Sensors* **2023**, *23*, 4376. https://doi.org/10.3390/s23094376.
83. Smolin, M. GenCoder: A generative AI-based adaptive intra-vehicle intrusion detection system. *IEEE Access* **2024**, *12*, 150651–150663. https://doi.org/10.1109/ACCESS.2024.3476177.
84. Smolin, M. GenCoder++: A protocol-aware and adversarially robust adaptive intrusion detection framework for hybrid CAN-Ethernet vehicular networks. *IEEE Access* **2025**, 13, *138381–138399*. https://doi.org/10.1109/ACCESS.2025.3595879.
85. Petit, J.; Shladover, S.E. Potential cyberattacks on automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 546–556. https://doi.org/10.1109/TITS.2014.2342271.
86. Amoozadeh, M.; Raghuramu, A.; Chuah, C.-N.; Ghosal, D.; Zhang, H.M.; Rowe, J.; Levitt, K. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Commun. Mag.* **2015**, *53*, 126–132. https://doi.org/10.1109/MCOM.2015.7120028.
87. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. In Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 18–21 February 2018. https://doi.org/10.14722/ndss.2018.23204.
88. Kairouz, P.; McMahan, H.B. Advances and open problems in federated learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. https://doi.org/10.1561/2200000083.
89. Alhajjar, E.; Maxwell, P.; Bastian, N.D. Adversarial machine learning in network intrusion detection systems. *arXiv* **2020**, arXiv:2004.11898. https://doi.org/10.48550/arXiv.2004.11898.
90. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. https://doi.org/10.1038/s42256-019-0048-x.
91. Adadi, A.; Berrada, M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* **2018**, *6*, 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052.
92. Zelle, D.; Lauser, T.; Kern, D.; Krauß, C. Analyzing and securing SOME/IP automotive services with formal and practical methods. In Proceedings of the 16th International Conference on Availability, Reliability and Security, Vienna, Austria, 17–20 August 2021; pp. 1–10. https://doi.org/10.1145/3465481.3465748.
93. Iorio, M.; Reineri, M.; Risso, F.; Sisto, R.; Valenza, F. Securing SOME/IP for in-vehicle service protection. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13450–13466. https://doi.org/10.1109/TVT.2020.3028880.
94. Lee, S.; Choi, W.; Lee, D.H. Protecting SOME/IP communication via authentication ticket. *Sensors* **2023**, *23*, 6293. https://doi.org/10.3390/s23146293.
95. Ma, B.; Wu, Y.; Yang, X.; Guan, X.; Zhang, W. An authentication and secure communication scheme for in-vehicle networks based on SOME/IP. *Sensors* **2022**, *22*, 647. https://doi.org/10.3390/s22020647.
96. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; pp. 1–16. https://doi.org/10.48550/arXiv.1711.03938.
97. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015. https://doi.org/10.48550/arXiv.1412.6572.
98. Aghaei, E.; Niu, X.; Shadid, W.; Al-Shaer, E. SecureBERT: A domain-specific language model for cybersecurity. In Proceedings of the 18th EAI International Conference on Security and Privacy in Communication Networks (SecureComm 2022), Virtual Event, 17–19 October 2022; pp. 39–56. https://doi.org/10.1007/978-3-031-25538-0_3.
99. *J3061_201601*; Cybersecurity Guidebook for Cyber-Physical Vehicle Systems. SAE International Recommended Practice: Warrendale, PA, USA, 2016. https://doi.org/10.4271/J3061_201601.
100. *ISO Standard 24089:2023*; Road Vehicles—Software Update Engineering. International Organization for Standardization: Geneva, Switzerland, 2023.
101. United Nations Economic Commission for Europe. *UN Regulation No. 156: Uniform Provisions Concerning the Approval of Vehicles with Regards to Software Update and Software Updates Management System*; UNECE: Geneva, Switzerland, 2021.
102. Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; NIST Trustworthy and Responsible AI; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. https://doi.org/10.6028/NIST.AI.100-1.
103. *ETSI EN 302 637-2 V1.4.1*; Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. ETSI: Sophia Antipolis, France, 2019.

104. AUTOSAR. *Specification of Intrusion Detection System Manager*; AUTOSAR Adaptive Platform, Release R23-11; AUTOSAR: Hörgertshausen, Germany, 2023.
105. Du, J.; Tang, R.; Sun, Y.; Feng, T. Security analysis and improvement of vehicle Ethernet SOME/IP protocol. *Sensors* **2022**, *22*, 6792. https://doi.org/10.3390/s22186792.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.