

Review

Operational Architectures for Hybrid AI-Based Network Intrusion Detection: A Critical Review and Multi-Axial Taxonomy

Alaa Haddad ¹, Sarvesh August Wajs ¹, Mohamed Hadi Habaebi ^{2,*}
and Mohammed Abdullah Salem Al-Hussaini ^{3,*}

¹ Faculty of Computing and Informatics, Multimedia University, Cyberjaya 63100, Malaysia; alaa.haddad@mmu.edu.my (A.H.)

² Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur 53100, Malaysia

³ Faculty of Computer Studies, Arab Open University Oman, Muscat 130, Oman

* Correspondence: habaebi@iiu.edu.my (M.H.H.); mohammed.h@aou.edu.om (M.A.S.A.-H.)

Abstract

Artificial intelligence is widely used in network intrusion detection, but evidence regarding operational architecture remains fragmented across model families, fusion patterns, deployment environments, and evaluation practices. We conducted a PRISMA 2020-compliant systematic review of secondary AI-NIDS reviews published from 1 January 2020 to 22 September 2026. OpenAlex and arXiv were searched on 22 September 2026; deterministic DOI/title deduplication and prespecified text rules preceded title/abstract screening, and a single reviewer assessed the retrieved full texts. Review-level limitations were appraised across eligibility, search reproducibility, selection/data collection, evidence traceability, and synthesis appropriateness; no aggregate quality score or statistical pooling was used. Of 8519 identified records, 1091 duplicates and 2256 automatically ineligible records were removed, 5172 records were screened, 101 reports were sought, 25 full texts were assessed, and 22 reviews were included. The synthesis supports a five-axis taxonomy spanning observation, AI intelligence, fusion, deployment, and trust/operations, but also shows weak search reproducibility in 12 of 22 included reviews and absent selection transparency in 15 of 22. The findings provide a bounded architectural framework rather than a universal performance ranking. Limitations include open-index coverage, 76 unretrieved reports, single-reviewer assessment, and the use of primary studies only as separate illustrative validation sources. The review was not prospectively registered; a revision-stage protocol was frozen before the new search, and its amendment is reported. A worked five-axis example, a decision-level contrast between sequential and hierarchical integration, appraisal-weighted confidence statements, and proposed evaluation criteria for each open challenge make the framework directly usable for design and assessment.



Academic Editor: Paolo Bellavista

Received: 20 September 2026

Revised: 14 October 2026

Accepted: 15 October 2026

Published: 17 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: hybrid intrusion detection; network intrusion detection; AI architecture; deployment evaluation; evidence synthesis; explainable AI; federated learning; reinforcement learning

1. Introduction

Intrusion detection is no longer primarily a choice between static signatures and statistical anomaly detectors [1,2]. Contemporary IDS research increasingly uses artificial

intelligence to learn traffic and host behaviour, fuse heterogeneous evidence, adapt decision thresholds, and support analyst triage [3–5]. This shift is driven by encrypted traffic, polymorphic attacks, cloud-native infrastructure, IoT heterogeneity, and the operational cost of false alerts [6–8]. Yet the term “AI-based IDS” is often applied too loosely: a paper may use an advanced classifier while retaining outdated data, unrealistic train–test splits, or no mechanism for drift, explanation, and deployment assurance [9–11].

The core technical problem is a coverage–precision–adaptation trade-off. Signature components remain efficient and interpretable for known attacks, whereas learned detectors can identify previously unseen patterns but are sensitive to class imbalance, distribution shift, adversarial manipulation, and benign behavioural change [9,12–14]. AI therefore does not remove the traditional IDS problem; it redistributes it across data quality, model selection, fusion logic, deployment architecture, and human oversight [11,15].

Recent work spans ensemble learning [16], convolutional and recurrent networks [17,18], autoencoders [19], graph neural networks [20], deep reinforcement learning [21,22], transformer encoders [23,24], federated optimisation [25,26], generative data augmentation [27,28], and explainability layers [29–31]. Hybrid designs published in 2026 illustrate how broadly the term “hybrid” is now used, including: metaheuristic hybridisation of feature selection before an SVM detector [32]; multi-scale feature interaction combined with attention-based bidirectional recurrent modelling [33]; and fusion of graph, sequential, and tabular representations supported by a generative augmentation loop [34]. Reviewing these approaches only by their algorithms produces an incomplete picture because operational performance also depends on where evidence is observed, how components are combined, where the model is deployed, and how uncertainty and drift are governed.

Existing reviews provide important but partial views. Broad surveys organise the field by detection technique and dataset [1,2,4,5]; hybrid-system reviews catalogue combinations of signature and anomaly components [35–37]; algorithm-centred reviews compare ML and DL classifiers [3,38–41]; and focused surveys examine reinforcement learning [21,22], adversarial learning [13,42], class imbalance [12], federated and collaborative learning [25,43,44], explainability [30,45], datasets [46,47], encrypted traffic [6], and SDN or programmable-data-plane deployment [7,48]. Methodological critiques further show that benchmark-centred evaluation can overstate operational value when datasets contain labelling artefacts, when temporal ordering is ignored, or when closed-world assumptions are made [9–11,49]. While these strands are valuable, they each answer a different question, use a different organising logic, and report evidence of different methodological quality. The present study therefore uses a systematic review-of-reviews design to compare these secondary syntheses and derive an operational taxonomy. Primary empirical papers, standards, and methodological guidance are retained only as separately labelled contextual or validation sources and are not counted in the PRISMA corpus.

1.1. Identified Research Gaps and Problem Statement

Reflection on this literature reveals three interlinked gaps. First, a fragmentation gap: none of the reviews compared in Table 1 jointly relates what a detector observes, what it learns, how its components are fused, where it is deployed, and how it is governed; consequently, two systems with the same classifier but different observation points, fusion logic, or deployment topologies are frequently treated as equivalent. Second, an integration gap: the terms sequential, parallel, and hierarchical hybridisation are widely used [35–37], but their decision logic, error propagation, and operational trade-offs are rarely distinguished explicitly, so architectural choices tend to be justified by benchmark accuracy rather than by deployment constraints. Third, an evidence-quality gap: secondary reviews differ markedly in coverage and rigour, yet their conclusions are commonly reused without

regard to how reproducibly they were searched or how transparently their studies were selected [50–52]. Together with the documented gap between reported and operational performance [9,11,49], these gaps mean that practitioners and researchers lack a validated, evidence-weighted basis for designing, comparing, and evaluating hybrid AI-NIDSs.

Table 1. Selected scope comparators from the included and contextual AI-based IDS review literature.

Survey (Corroborating Sources)	Period	Systematic Method	ML/DL	RL/Transformers	Federated/XAI	Fusion Taxonomy	Quantitative Evidence
Khraisat et al. [1]; see also [2,53–55]	To 2019	Narrative survey	Partial	No	No	General IDS types	No
Maseno et al. [35]; see also [36,37]	To 2022	Systematic review	Yes	Limited	Limited	Hybrid components	Descriptive
Jacob & Habibullah [3]; see also [5,39–41]	To 2024	Systematic analysis	Yes	Limited	No	Algorithm- centred	No
Louati et al. [21]; see also [22]	To 2024	Focused survey	RL focus	RL only	No	RL methods	No
Sowmya & Anita [38]; see also [4,54]	To 2021	Narrative review	Yes	No	No	Algorithm- centred	No
Levy & Khoshgoftaar [46]; see also [47,56,57]	To 2020	Dataset-focused survey	Yes	No	No	Dataset/model taxonomy	Descriptive
Papadogiannaki & Ioannidis [6]; see also [58]	To 2021	Encrypted-traffic survey	Partial	No	No	Traffic-analysis taxonomy	Descriptive
Janabi et al. [7]; see also [48]	To 2024	SDN-focused survey	Yes	Limited	Limited	SDN IDS taxonomy	Descriptive
Khenwar & Nawal [59]; see also [49,60]	To 2024	Challenges assessment	Partial	No	No	IDS limitation categories	No
This review	2020–2026	Systematic review of reviews; PRISMA 2020	Yes	Yes	Yes	Five-axis system taxonomy	Qualitative source-specific synthesis

Table 1 presents selected scope comparators rather than the complete PRISMA corpus. Each comparator is accompanied by corroborating reviews or studies with a similar focus, so that every scope statement rests on more than one contribution. The complete set of 22 included reviews, their axis contributions, and their domain-level appraisal results are reported in Supplementary Table S1. The comparison supports a bounded novelty claim: prior reviews commonly specialise in algorithms, datasets, adversarial learning, or deployment paradigms, whereas the present framework aligns observation source, AI capability, fusion logic, deployment topology, and trust/operations within one coordinate system.

Three patterns emerge from Table 1. First, coverage is uneven across the five design questions: most comparators address ML/DL algorithms, whereas RL or transformer models, federated or XAI mechanisms, and fusion logic are covered only partially or within single-topic surveys [6,7,21,25]. Second, the organising logic is predominantly algorithm-centred or dataset-centred [3,4,38,46]; only the hybrid-system reviews [35–37] treat the combination of components as the unit of analysis, and none of the listed comparators relates fusion logic to deployment topology and operational assurance. Third, quantitative evidence is, at most, descriptive: no comparator pools performance across studies, which is consistent with the heterogeneity of datasets, splits, and metrics documented by dataset-validity and evaluation-methodology studies [10,11,47,49,56,57]. In addition, most comparators cover the literature only up to 2019–2024, so recent transformer, federated, neurosymbolic, graph-based, and programmable-data-plane work is under-

represented [20,24,48,61,62]. These observations define the gap targeted here, which is the absence of a quality-aware synthesis that aligns their partial perspectives within one operational frame and not the absence of reviews themselves.

The global research problem addressed by this review is therefore the following: how can the fragmented and methodologically heterogeneous secondary evidence on AI-based network intrusion detection be consolidated, with explicit regard to its quality, into an operational architecture framework that allows hybrid AI-NIDSs to be consistently described, compared, and evaluated for trustworthy deployment? Dataset obsolescence, opaque fusion logic, encrypted-traffic blindness, privacy leakage in federated training, and limited explainability are treated as manifestations of this single problem rather than as separate objectives. The study addresses the problem by (i) appraising and synthesising the secondary-review evidence base; (ii) deriving, illustrating, and stress-testing a multi-axial architectural taxonomy; and (iii) translating the remaining gaps into an evaluable research agenda.

Scope boundary. The review is architecture-focused and network-centred. It includes host data only when host events are fused with network observations, and it excludes standalone malware classification, generic vulnerability scanning, purely cryptographic schemes, and classifier studies that do not inform integration, deployment, or evaluation decisions. Algorithm families are discussed only where they clarify an architectural or operational trade-off.

Within this boundary, the global research problem is decomposed into four research questions.

- RQ1: Which limitations of conventional and single-model network IDSs motivate hybrid AI designs?
- RQ2: How should hybrid AI-NIDS architectures be classified across observation, learning, fusion, deployment, and assurance dimensions?
- RQ3: What evaluation and reporting weaknesses recur across prior reviews?
- RQ4: Which evidence gaps most constrain trustworthy operational deployment?

The contribution is a systematic synthesis of reviews and an architectural taxonomy; it is not a census of all classifier papers or a pooled estimate of model performance. RQ1 is answered in Sections 3 and 4; RQ2 in Section 5; RQ3 in Sections 2.6 and 2.8, and 7; and RQ4 in Sections 6 and 8.

1.2. Novelty and Contributions

Existing reviews separately examine machine learning algorithms, reinforcement learning designs, federated learning, datasets, or explainability. This review contributes a PRISMA-audited synthesis of secondary AI-NIDS reviews and connects their findings only where they affect the architecture and operational evaluation of hybrid network-centred IDSs. The contribution is integrative rather than a claim that each individual axis is unprecedented.

- We develop a five-axis taxonomy that classifies hybrid AI-NIDSs across observation layers, AI intelligence, fusion patterns, deployment topologies, and trust and operational assurance.
- We synthesise three dominant hybrid deployment architectures (sequential, parallel, and hierarchical), explicitly distinguishing their decision logic and mapping how rule-based heuristics and learning-based models interact to optimise the computational coverage–precision trade-off.
- We retain source-specific performance findings only when they can be traced to the cited study, and we explain why heterogeneous datasets, metric definitions, and validation protocols do not support statistical pooling or a universal performance ranking.

- We identify and formulate six critical barriers to production-grade deployment, ranging from adversarial robustness and non-IID federated convergence to encrypted-traffic blind spots, and we present specific, evidence-backed mitigations to guide future research, together with proposed evaluation criteria that make each mitigation testable.
- We demonstrate the taxonomy through a complete worked example and link the review-level appraisal outcomes to the confidence attached to each architectural recommendation and research gap.

2. Review Methodology

This study was redesigned as a systematic review of secondary AI-NIDS reviews. Reporting follows PRISMA 2020 and PRISMA-S [63,64]. The review uses qualitative synthesis because the included reviews differ in scope, study units, search methods, datasets, and reported outcomes; no statistical pooling or universal performance ranking was attempted.

2.1. Information Sources, Search Date, and Search Strategy

OpenAlex and arXiv were searched on 22 September 2026 for reports published from 1 January 2020 to 22 September 2026. OpenAlex was selected as a reproducible multidisciplinary scholarly index and arXiv as a preprint source relevant to computer science. Subscription-only databases were not simulated. The use of open indexes, rather than direct IEEE Xplore, Scopus, Web of Science, or ACM Digital Library exports, is reported as a coverage limitation.

The OpenAlex Works API used the filter `title_and_abstract.search`: “network intrusion detection” with the publication-date bounds above. The arXiv API used: (ti: “network intrusion detection” OR abs: “network intrusion detection” OR (all: NIDS AND all: intrusion)) AND (all: “machine learning” OR all: “deep learning” OR all: “reinforcement learning” OR all: transformer OR all: “federated learning” OR all: “explainable AI” OR all: XAI), restricted by a `submittedDate` range from 202001010000 to 202609222359. Complete endpoint parameters, saved exports, and query dates are reported in Supplementary Methods S1.

The initial sensitivity search was deliberately broad. Prespecified deterministic rules then required a direct network-IDS expression and an AI/learning expression. These rules were applied before title/abstract screening and are reported as automation instead of being silently treated as database filters. Preprints were retained only when no archival version of the same report was identified.

2.2. Eligibility Criteria and Evidence Roles

The PRISMA corpus unit was a secondary review: a systematic review, mapping study, survey, meta-analysis, or explicit taxonomy substantially focused on AI-enabled, network-centred intrusion detection. Eligible reports had to contribute evidence to at least one architectural axis—observation, AI intelligence, fusion, deployment, or trust/operations—or to cross-study evaluation practice.

Eligible publication types were English-language journal articles, peer-reviewed conference papers, review articles, and preprints without an identified archival version. The review period was 2020–2026. No geographic restriction was imposed.

- Reports were excluded when they were primary classifier evaluations, host-only malware studies without network evidence, generic vulnerability-scanning or cryptographic studies, non-network anomaly detection, editorials or tutorials without synthesis, outside the date range, or insufficiently complete for appraisal. One primary full-text exclusion reason was assigned to each report.
- Primary empirical studies are not part of the PRISMA corpus. They are used separately and explicitly as illustrative validation sources for datasets, mechanisms, or recent

architectures. Standards, regulations, and review-method guidance are contextual sources only. None of these sources contributes to the review-selection counts.

- The included reviews were grouped for synthesis by the five architectural questions and by review scope. Numerical model outcomes reported in underlying primary studies were not re-pooled because the reviews used incompatible datasets, split protocols, attack definitions, and metrics.

Selection of secondary reviews. Because secondary reviews differ widely in scientific coverage and methodological quality, selection was governed by four decision rules applied in sequence to every record, and neither coverage nor quality was used as a reason for exclusion at the screening stage. (i) Secondary status: The report had to synthesise evidence from multiple prior studies (systematic review, mapping study, survey, meta-analysis, or explicit taxonomy) rather than report a new classifier experiment. (ii) Topical focus: AI-enabled detection of network intrusions had to be a substantial focus of the report, as stated in its title, abstract, or objectives and reflected in its body; reviews of general cybersecurity AI, malware, or host-only detection were not eligible. (iii) Architectural or evaluative relevance: The review had to contribute extractable statements to at least one of the five axes or to cross-study evaluation practice. (iv) Appraisability: The full text had to be retrievable and sufficiently complete for the five-domain appraisal described in Section 2.4. Reviews meeting these rules were included irrespective of venue, citation count, or methodological quality.

Differences in coverage and quality were handled after inclusion rather than by exclusion. For coverage, the extraction form recorded each review's declared period, topical scope, and the axes it contributed to (Supplementary Table S1), so a narrow review contributes only to the axes it covers. For quality, each review was appraised on five domains (D1–D5, Section 2.4), and the appraisal determined how strongly its statements could support each type of synthesis claim (Section 2.8). This design avoids two opposite biases: excluding narrow or methodologically weak reviews would discard relevant architectural evidence and inflate apparent consensus, whereas treating all reviews as equally reliable would overstate weakly supported claims.

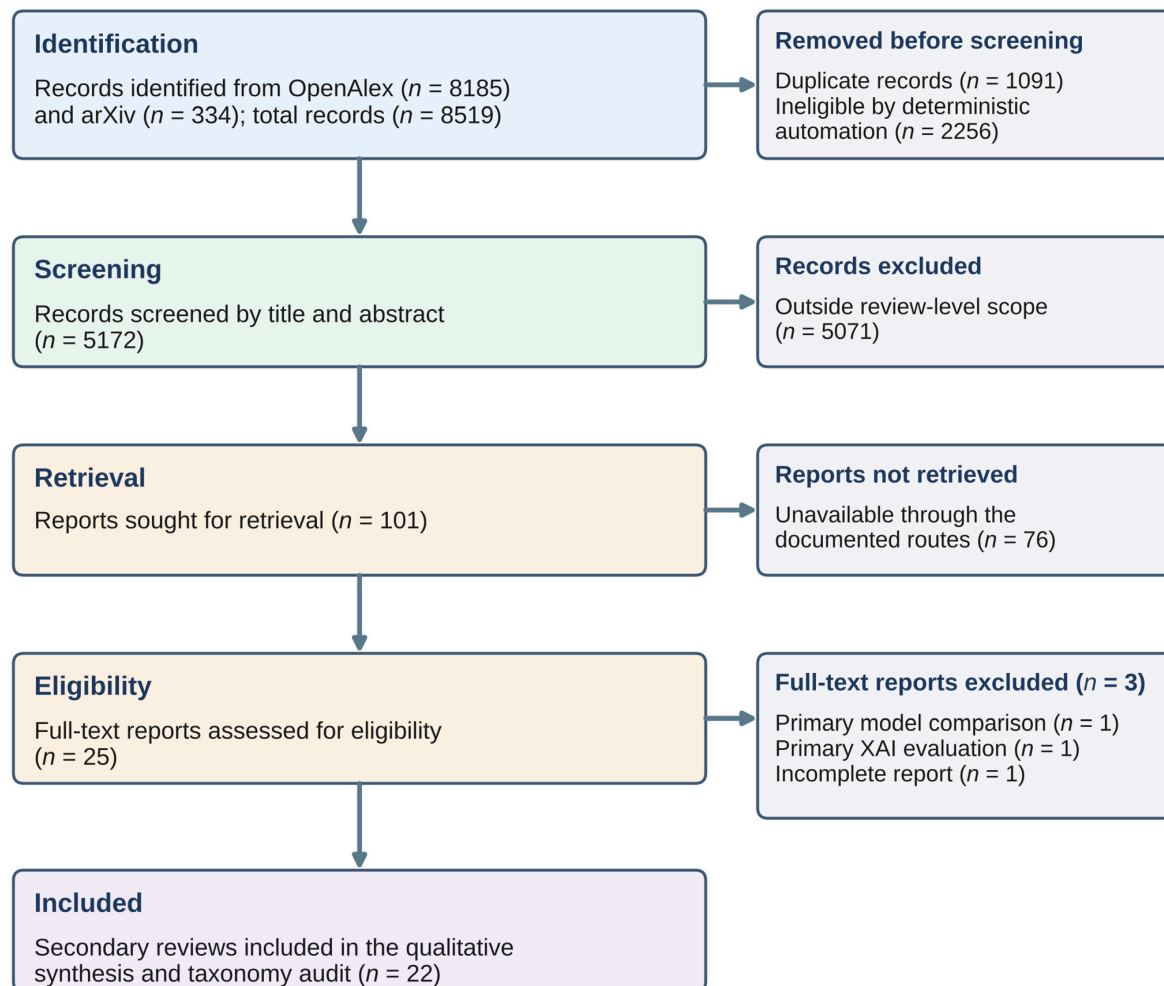
2.3. Study Selection, Automation, and PRISMA 2020 Compliance

The review is reported in accordance with the PRISMA 2020 checklist, with search reporting informed by PRISMA-S [63,64]. It was not prospectively registered. A revision-stage protocol was frozen before the new searches; this local protocol is not presented as a retrospective registration. One eligibility clarification made after the sensitivity pilot but before full-text decisions is reported in Section 2.7 and Supplementary Methods S2.

Records were deduplicated first by exact DOI and then by normalised title plus compatible author, venue, year, or preprint-version information. Deterministic Python (version 3.11) rules removed records lacking a direct network-IDS or AI/learning expression. Remaining titles and abstracts were screened for secondary-review status and architectural or evaluative relevance. Full reports were then sought and assessed against Section 2.2. The screening ledger preserves every record, duplicate link, stage, decision, and reason.

A single reviewer conducted the revision-stage record screening, full-text eligibility assessment, extraction, and review-level appraisal. No independent duplicate screening, disagreement arbitration, or inter-rater statistic was performed. The deterministic automation was rule-based rather than a learned classifier; its exact expressions and all automatically excluded records are retained for audit. This single-reviewer design is treated as a methodological limitation.

Figure 1 reports the complete numerical study-selection flow. Counts were accepted only after arithmetic reconciliation with the record-level ledger and full-text retrieval log.



Search window: 1 January 2020 to 22 September 2026; searches executed on 22 September 2026.

Reconciled counts: $8519 - 1091 - 2256 = 5172$; $5172 - 5071 = 101$; $101 - 76 = 25$; $25 - 3 = 22$.

Figure 1. PRISMA 2020 flow diagram for the systematic review of secondary AI-NIDS reviews. The search covered 1 January 2020–22 September 2026 and was executed on 22 September 2026. Deterministic pre-screening rules and record-level decisions are preserved in the supplementary audit files [63,64].

2.4. Data Collection and Review-Level Risk-of-Bias Appraisal

For each included review, one reviewer extracted bibliographic data, declared scope, publication type, search and selection information, evidence role, synthesis method, and contribution to the five axes. Missing or unclear information was recorded as not reported rather than inferred. A complete included-review matrix is provided in Supplementary Table S1.

Retrieval status was recorded separately from eligibility. A report was counted as assessed only when the saved full text could be opened and contained enough content for appraisal. Reports that could not be obtained through the documented routes were classified as not retrieved, not as methodologically ineligible.

Risk of bias and methodological limitations were assessed with five prespecified review-level domains informed by systematic review guidance and ROBIS [50–52]: (D1) explicit eligibility and scope; (D2) reproducible identification and search informa-

tion; (D3) transparent selection and data collection; (D4) traceable characteristics and findings for cited studies; and (D5) synthesis appropriate to heterogeneous evidence and bounded claims.

Each domain was recorded as met (Y), partially met (P), or not met (N) using the decision rules in Supplementary Methods S3. Domain judgements were not summed into an unexplained total score or converted into labels, such as critical, high, or medium. Instead, limitations constrained the strength and generality of the synthesis, while operational implications were interpreted using general information security risk principles [65,66].

2.5. Qualitative Synthesis and Taxonomy Derivation

A hybrid deductive–inductive thematic procedure was used. The review questions supplied sensitising domains, while additional codes were retained when included reviews explicitly described observation sources, AI mechanisms, component interactions, deployment constraints, or assurance practices [67].

The unit of coding was a source-linked architectural or evaluative statement. Codes derived from the 22 included reviews formed the systematic evidence base [12,13,36,37,39–44,49,53–56,60,62,68–72]. Separately labelled primary studies could test whether a recent architecture was placeable within the axes, but they could not create a PRISMA count or substitute for an included review.

Synonyms were consolidated only when they represented the same mechanism; codes were split when they represented different design decisions. For example, packet headers and encrypted-flow statistics were grouped as observable network evidence, whereas sequential escalation and parallel score fusion remained distinct coupling patterns.

A candidate axis was retained when it answered a distinct design question, appeared across more than one included review, affected architecture or evaluation, and did not duplicate another axis. This yielded five axes: what is observed, what is learned, how components are fused, where computation occurs, and how trust and operations are governed (Table 2).

Table 2. Axis-level audit for the five-axis taxonomy derived from the included reviews.

Taxonomic Axis	Distinct Design Question and Retention Rule	Consolidated Codes	Independent Source Families
Observation layer	What evidence can the detector observe? Retain when the source changes input visibility or modality.	Packet/payload; flow and TLS metadata; host events; cross-layer evidence	[37,39–41,53,54,56,60,69–72]
AI intelligence	What representation or decision mechanism is learned? Retain when the learner changes inference or adaptation.	Classical ML; deep/transformer; RL; graph and generative learning	[12,13,36,37,39–44,49,53–56,60,62,68,70–72]
Fusion pattern	How are components or evidence combined? Retain when coupling changes processing order or decision aggregation.	Sequential escalation; parallel/ensemble; hierarchical/tiered; network-host fusion	[36,37,60,69]
Deployment	Where do training, inference, and coordination occur? Retain when topology changes data, parameter, or traffic movement.	Centralised; edge/fog; SDN/cloud; federated; data-plane acceleration	[37,43,44,49,60,62,72]
Trust and operations	How are explanation, privacy, robustness, drift, latency, and analyst control governed? Retain when assurance risk changes.	XAI and validation; privacy/leakage; adversarial robustness; drift; human and latency controls	[12,13,36,37,39,42–44,49,53,55,56,60,62,70]

Negative-case and boundary checking allowed one review to contribute to multiple axes and prevented implementation details from being promoted to independent axes. Frequency was not interpreted as importance, and the taxonomy is not a performance ranking or causal model.

Synthesis followed structured narrative principles; tables were used to compare scope, evidence roles, and axis contributions. No effect measure was prespecified because no compatible intervention effect was estimated. Statistical assessment of reporting bias, meta-analysis, subgroup analysis, and sensitivity analysis were not performed. Formal GRADE certainty ratings were not applied; confidence was bounded through the domain-level appraisal and source-specific wording [73].

2.6. Study Selection and Characteristics of Included Reviews

The searches identified 8519 records: 8185 from OpenAlex and 334 from arXiv. Before screening, 1091 duplicate records and 2256 records marked ineligible by deterministic automation were removed. Of 5172 records screened by their titles and abstracts, 5071 were excluded and 101 reports were sought. Seventy-six reports were not retrieved, 25 full texts were assessed, three were excluded with one documented reason each, and 22 reviews were included in the qualitative synthesis (Figure 1).

The three full-text exclusions were one primary model-comparison study, one primary explainability evaluation, and one incomplete report. The 22 included reviews were published from 2020 to 2026 and covered general ML/DL, adversarial learning, imbalanced data, AutoML, federated and quantum learning, hybrid models, dataset validity, and the gap between benchmark and operational evaluation [12,13,36,37,39–44,49,53–56,60,62,68–72]. Supplementary Table S1 provides the complete characteristics and source-to-axis matrix.

The appraisal exposed substantial review-level limitations. Explicit eligibility criteria were fully reported in four reviews and only partially reported in 18. Search reproducibility was met in four, partial in six, and absent in 12. Selection and data-collection transparency was met in four, partial in three, and absent in 15. Evidence traceability was met in 14 and partial in eight; synthesis appropriateness was met in eight and partial in 14. These findings support cautious thematic synthesis but not claims of exhaustive or bias-free coverage.

No statistical reporting-bias test or certainty-of-effect framework was applicable because the review did not estimate a common effect. Access-related bias remains material: 76 of 101 sought reports were not retrieved. The synthesis therefore distinguishes findings repeatedly reported across accessible reviews from claims that rest on a single or methodologically weak source.

2.7. Protocol Amendment and Availability of Review Materials

After the broad sensitivity export and before eligibility decisions, the systematic unit was clarified from mixed primary and secondary evidence to secondary AI-NIDS reviews, surveys, mappings, meta-analyses, and explicit taxonomies. The broad search returned thousands of single-classifier primary studies that did not match the manuscript's meta-survey purpose. The amendment narrowed the evidence role without changing the search export, date limits, research questions, or taxonomy axes; it is documented to satisfy PRISMA Item 24c.

The exact search parameters, screening ledger, retrieval log, full-text exclusion reasons, included-review characteristics, appraisal matrix, and completed PRISMA checklist accompany this revision. Raw API responses are retained with the revision audit package.

Table 3 presents specialised taxonomies used as contextual stress tests; it is not a substitute for the complete included-review matrix in Supplementary Table S1. The bounded nov-

elty claim is that the specialised literature can be aligned through five operational questions, not that no prior taxonomy exists or that the present framework is universally superior.

Table 3. Comparison with recent AI-NIDS taxonomies and surveys.

Source	Primary Scope and Organising Logic	Boundary of that Scope	Relation to the Five-Axis Framework
Wardana and Sukarno [25]	Collaborative IDS using federated learning; participants, collaboration, and FL design.	Specialised to collaborative/federated IDSs.	Deepens deployment and trust categories; does not replace the cross-paradigm axes.
Bizzarri et al. [61]	Neurosymbolic NIDS; integration of neural and symbolic reasoning.	Specialised to neurosymbolic techniques.	Tests AI-intelligence, fusion, and trust placement.
Ismaeli Taba et al. [20]	GNN-NIDS; graph construction, model families, and evaluation.	Specialised to graph-based representations and learning.	Tests observation and AI-intelligence placement.
Liu et al. [62]	AutoML-NIDS; automated pipelines and parallel/distributed paradigms.	Specialised to automation of model and pipeline design.	Tests AI-intelligence and deployment placement.
Fida and Ahmed [48]	Programmable-data-plane NIDS; in-network execution and line-rate constraints.	Specialised to data-plane deployment.	Tests deployment and trust/operations placement.
Present review	Cross-paradigm operational architecture organised by five distinct design questions.	Systematic review of 22 secondary reviews; no universal performance ranking.	Aligns specialised taxonomies across observation, intelligence, fusion, deployment, and trust/operations.

2.8. Implications of Review-Level Appraisal for the Synthesis

The appraisal results in Section 2.6 are not just descriptive; they constrain how the architectural recommendations and research gaps in Sections 5–8 should be read. Weak search reproducibility (not met in 12 of 22 reviews) means that the coverage of an individual review cannot be verified, so the absence of a topic within that review cannot be interpreted as its absence from the literature. Absent selection transparency (15 of 22) means that the studies a review discusses may reflect convenience or preference rather than systematic selection, so the prevalence or apparent success of a technique within that review cannot be generalised. In contrast, evidence traceability was met in 14 reviews and at least partially met in all 22, which supports the use of the corpus as a source of traceable descriptive statements about mechanisms, design options, and reported limitations.

Accordingly, the confidence attached to each type of synthesis claim was graded qualitatively according to its sensitivity to these weaknesses (Table 4). Claims about the existence of design options and mechanisms, such as the five axes and the three integration patterns, depend mainly on traceable description and are therefore comparatively robust; in addition, an axis was retained only when it recurred across more than one included review (Section 2.5) and could be populated by separately labelled primary studies. Claims about relative prevalence, maturity, or superiority are highly sensitive to unreproducible searches and opaque selection and were therefore not made. Research gaps are the most sensitive claim type, because a gap reported by a review with an unreproducible search may reflect that review’s search rather than the literature; gaps are therefore phrased

as under-reported in the accessible review literature and were retained only when they recurred across reviews and were consistent with primary-study evidence and independent methodological critiques [9–11,49].

Table 4. Influence of review-level appraisal outcomes on confidence in synthesis claims.

Claim Type	Examples in this Review	Sensitivity to Weak Search (D2) and Selection (D3) Reporting	Safeguard Applied	Resulting Confidence
Existence of design dimensions and mechanisms	Five axes (Table 2); three integration patterns (Section 5.2)	Low—depends on traceable description (D4 met in 14, partial in 8)	Retention only when recurring in >1 review; boundary checking; placement of primary studies [20,24–26,31–34,48,61,62,74,75]	Moderate to high for structure; not a claim of exhaustiveness
Conditional architectural recommendations	Sequential for throughput, parallel for coverage, and hierarchical for selective escalation (Sections 7.1 and 8)	Moderate—rests on mechanism reasoning and source-specific primary results	Phrased conditionally; no ranking; validation in the target environment required	Moderate, conditional
Prevalence, maturity, or performance ranking	Most-used or most-accurate model family	High—unreproducible searches (12/22) and opaque selection (15/22) bias counts	Not claimed; frequency not read as importance; no pooling	Not supported
Research gaps	Encrypted traffic, non-IID federation, explanation reconciliation, and temporal drift (Section 6)	High—an apparent gap may reflect a review’s search	Retained only when recurring across reviews and corroborated by primary or methodological evidence; framed as under-reported	Low to moderate; to be confirmed by updated searches
Weaknesses of evaluation practice	Random splits, dataset age, missing latency and drift metrics (Section 7.4)	Low to moderate—reported consistently across reviews	Triangulation with independent methodological studies [9–11,49]	Moderate to high

These grades are qualitative and are not GRADE certainty ratings. They make explicit why the review offers conditional design guidance and an evaluable research agenda rather than definitive rankings or proof that a specific gap exists; Sections 6 and 8 refer back to them when stating challenges and conclusions.

3. Benchmark Datasets for IDS Research

Dataset choice is one of the most consequential methodological decisions in IDS research. The datasets discussed below illustrate differences in age, modality, class balance, and realism that shape what a benchmark can and cannot establish. Table 5 presents a consolidated comparison.

Table 5. Comparative analysis of IDS benchmark datasets.

Dataset	Reference	Type	Attack Categories	Key Strengths	Key Limitations	Hybrid IDS Suitability
NSL-KDD	Tavallae et al. [76]	Network	4 (DoS, Probe, R2L, U2R)	Extensive baselines; publicly available	1999-era traffic; obsolete vectors	Moderate, benchmarking only
CICIDS 2017	Neto et al. [77]; Sajid [17]	Network	7 modern categories	Realistic mix; DL-compatible; $n = 284$ K	Minority class imbalance	High, broad coverage
CSE-CIC-IDS 2018	Leevy [46]; Chen [78]	Network	12 categories incl. botnet	Rich taxonomy; large scale; 16 M records	High preprocessing demand; imbalance	Very high, scalable evaluation
UNSW-NB15	Yin et al. [79]; Hermosilla [29]	Hybrid	9 categories incl. shellcode	Cross-layer; balanced; modern	Some types underrepresented	Very high, cross-layer
AIT IDS	Landauer [80]	Hybrid	APT; multi-stage attacks	Enterprise realism; temporal context	Limited adoption; few baselines	Very high, APT evaluation

3.1. Network-Based Datasets

NSL-KDD was introduced by Tavallae et al. [76] as a refined version of KDD Cup 1999 in which duplicate records were reduced. Its 41 features describe TCP connections and host-related indicators across DoS, Probe, Remote-to-Local, and User-to-Root attacks. Its extensive use supports methodological comparison, but the simulated 1998-era traffic pre-dates current cloud, IoT, encrypted-traffic, and ransomware behaviours. High performance on NSL-KDD should therefore be interpreted as competence on a historical benchmark, not evidence of modern deployment robustness.

Sajid et al. [17] combined feature selection with CNN–LSTM classification, illustrating a pipeline in which preprocessing and temporal modelling operate together. The reported benchmark values remain specific to the authors’ dataset construction and validation protocol.

CSE-CIC-IDS 2018 extends CICIDS 2017 with scenarios that better reflect what enterprise networks actually face, such as cryptojacking, botnet command traffic, social-engineering-based infiltration, and SSH/FTP brute force. The data were collected over five days in a 50-machine simulation environment and contain roughly 16 million labelled records across 12 attack categories, making it the largest and most diverse publicly available network-IDS benchmark. Leevy and Khoshgoftaar [46] examined multiple classifiers on this dataset from a throughput perspective and found that the sheer volume creates real preprocessing and streaming challenges. This observation itself is practically useful: any hybrid IDS evaluated on CSE-CIC-IDS 2018 must address scalability and not just accuracy.

3.2. Host-Based Datasets

Host-based data, including system-call traces, process events, and Windows logs, can reveal activity that is not visible in network flows. The survey by Bridges et al. [81] catalogues the wide variety of host data sources and detection methods; because these sources depend on the operating system and logging configuration, environmental specificity limits transfer to a different estate.

Host-event studies also require careful domain validation because operating-system versions, logging policies, and organisational baselines can change the observed distribution. A model evaluated in one environment should therefore not be assumed to transfer unchanged to another.

3.3. Hybrid Datasets

UNSW-NB15 combines network-flow attributes with host-derived features across nine attack categories. It is widely used for evaluating feature selection, explainability, and hybrid classifiers, but results remain dependent on preprocessing and validation design [29,79].

The AIT IDS dataset from the Austrian Institute of Technology is arguably the most realistic publicly available benchmark we reviewed. It combines packet captures with host logs collected during weeks of enterprise simulation under multi-stage APT scenarios, capturing the temporal and causal relationships between detection events that single-stage datasets simply cannot represent. Landauer et al. [80] describe how such log datasets can be generated and maintained in a testbed with labelled multi-step attack scenarios, enabling detectors to be evaluated on temporally ordered evidence collected from several hosts, which is precisely the evaluation that the dataset was designed to enable. The main limitation is limited adoption: fewer comparative baselines exist than for UNSW-NB15 or CICIDS 2017, making it more difficult to contextualise new results. That is likely to change as multi-stage APT detection becomes a higher research priority.

Table 5 demonstrates that no single benchmark is sufficient for deployment-level evaluation. Older network-only datasets support comparison with established baselines, whereas newer or hybrid datasets provide broader attack coverage and richer context but introduce greater preprocessing cost and fewer comparable baselines. Dataset selection must therefore follow the intended observation layer and threat model.

4. Structural Limitations of Existing IDS Tools

To answer RQ1 meaningfully, we need to go beyond listing what these tools cannot do and instead examine why they cannot do it, as well as whether the limitation is fundamental to the architecture or merely a feature of the current implementation. Table 6 summarises the findings; the subsections explain the evidence behind them.

Table 6. Structural limitations of deployed IDS tools and evidence-based mitigations.

Tool	Architecture	Primary Limitation	Evidence	Hybrid AI Mitigation
Snort	Aho–Corasick rule-matching pipeline	Zero-day blind spot; maintenance overhead; detection latency	Xu [82]; Rathee [83]; Ugtakbayar [84]	ML/DL augmentation for anomaly detection alongside signatures
Suricata	Multi-thread + rules + Lua	Packet drop >10 Gbps; alert fatigue; FPR not improved by speed	Hu [85]; Ghazi [86]; Saheed [87]	ML augmentation; evaluate FPR under common load
Zeek (Bro)	Event-driven protocol logging	Reduced payload visibility under TLS, DoH, and QUIC	Papadogiannaki & Ioannidis [6]; Janabi [7]	JA3/JA4 fingerprinting; AI flow metadata analysis [6]
Cisco IPS	Proprietary sig. + prevention stack	Signature-update delay	Khraisat et al. [1]; Chinnasamy et al. [88]	AI behavioural baselines alongside static sigs; adaptive ML [88]

4.1. Snort

Snort processes packets through a pipeline that culminates in Aho–Corasick multi-pattern matching against a community-maintained rule database. The algorithm is $O(n)$ in packet payload length, which is why Snort can classify packets at near line rate on commodity hardware. Speed is not the problem. The problem is that this design is fundamentally

reactive: it can only recognise threats whose patterns are already in the database. Xu [82] showed through systematic benchmarking that this creates an unavoidable zero-day blind spot, and that ML integration is the practical path to reducing false positives from benign traffic that superficially resembles known attack patterns.

Rathee et al. [83] explored how deep learning models can augment Snort by learning behavioural representations that generalise beyond explicit rule patterns, catching variant attacks that evade exact matching. Ugtakhybayar et al. [84] went a step further and proposed a hybrid model that pairs Snort-style signature detection with ML-based anomaly scoring, demonstrating improved zero-day coverage in experiments on NSL-KDD. A secondary concern is maintenance. The Snort community rule set demands continuous monitoring, testing, and distribution of updates, a task that many smaller organisations cannot sustain [89]. Outdated rule sets are not merely less effective; they can also create a false sense of security. This statement requires qualification. An outdated rule set still detects the attacks it encodes, so in terms of detection coverage it is not literally worse than having no IDS. The risk is organisational: because a signature engine is silent whenever no rule matches, the absence of alerts is easily misread by operators and management as evidence that no intrusion is occurring, whereas it may only reflect missing signatures for current threats. When this misplaced assurance leads to reduced investment in complementary controls, such as log review, threat hunting, or endpoint monitoring, an unmaintained IDS can leave an organisation less protected than one that knowingly operates without such a system and compensates through other controls. The concern therefore relates to the governance of detection coverage rather than to the detection capability of the engine itself, and it motivates hybrid designs that pair signature matching with learned anomaly components and with explicit monitoring of rule-set currency.

4.2. Suricata

Suricata was built by the Open Information Security Foundation as a modern successor to Snort, adding native multi-threading, GPU-accelerated pattern matching, and a Lua scripting interface for custom detection logic. These improvements mean Suricata can handle high-bandwidth links that would drop packets on single-threaded Snort deployments. Ghazi et al. [86] confirmed the throughput advantage under load, though they also found that anomaly detection capability remains bounded by the underlying rule engine regardless of how fast the matching runs.

At high packet rates, packet loss and alert volume can reduce practical coverage. Learned components may complement Suricata's rule engine, but operational comparisons require a common traffic load, hardware platform, threshold policy, and false-positive definition [85,86].

4.3. Zeek (Formerly Bro)

Zeek takes a fundamentally different approach from Snort and Suricata. Rather than matching packets against rules, it dissects protocols and emits structured semantic logs: connection summaries, DNS queries, HTTP transactions, TLS handshake metadata, file transfers, and custom application-layer events. This richer representation supports detection of multi-step attack patterns that simple rule engines cannot express, and Zeek scripts written in its purpose-built policy language can implement sophisticated behavioural detection logic.

The growing structural problem is encryption. TLS 1.3 encrypts the handshake entirely after key negotiation, eliminating certificate subject names and many negotiated parameter fields that were observable in TLS 1.2. DNS-over-HTTPS routes resolution queries through encrypted channels, hiding domain names from network observers. QUIC embeds all pay-

load and most headers inside mandatory-encryption UDP datagrams. Papadogiannaki and Ioannidis [6] surveyed countermeasures and identified TLS handshake fingerprinting, such as JA3, which hashes the observable ClientHello parameters before encryption completes, combined with flow-level statistical analysis, as the most practical current mitigation for payload-blind detection; the later JA4 method extends this fingerprinting approach. Janabi et al. [7] demonstrated that resource-aware deployment architectures can help Zeek maintain real-time coverage in SDN environments where resource constraints would otherwise force it offline under load.

4.4. Cisco IPS

Cisco IPS, now integrated with Cisco Firepower, represents the commercial end of the spectrum: a tightly coupled hardware–software stack with centralised management, dedicated threat intelligence feeds, and deep interoperability with the broader Cisco security ecosystem. Its technical architecture differs from the open-source tools above primarily in the quality of its signature curation and the speed of its update distribution pipeline.

Commercial IPS platforms benefit from curated intelligence and integrated management, yet signature-dependent detection still inherits the delay between threat discovery, rule creation, validation, and distribution. Behavioural models may complement this cycle, but benchmark accuracy alone does not establish production effectiveness [7,90].

Table 6 indicates that the principal limitations arise from different architectural causes: signature dependence constrains Snort and commercial IPS platforms, throughput and alert handling constrain Suricata, and encryption reduces Zeek’s observable payload. Consequently, the AI component should be selected to address the specific limitation rather than added as a generic replacement for the existing tool.

5. Hybrid AI-Driven IDS Architectures

Answering RQ2 requires more than cataloguing techniques. What practitioners need requires an understanding of how those techniques are architecturally related to one another, which combinations work, why they work, and how the design choice affects operational performance. We begin with the taxonomy, as this frames everything that follows.

5.1. Multi-Axial Taxonomy of AI-Based IDSs

The integration-pattern taxonomy is useful but not sufficient by itself. Figure 2 extends it into five distinct design questions. The observation layer identifies what evidence the system can see; AI intelligence identifies how representations and decisions are learned; fusion identifies how rules and learned components interact; deployment captures where training and inference occur; and trust and operations captures explanation, drift, adversarial robustness, privacy, latency, and human validation. The complete axis-level audit is reported in Table 2, the source-level disposition is provided in Supplementary Table S1, and recent specialised taxonomies are compared in Section 2.7 and Table 3.

Figure 2 should be read from left to right as five independent design decisions. It separates the evidence observed with the IDS from the learning paradigm, fusion logic, deployment location, and assurance controls. Systems using the same classifier are therefore not treated as equivalent when their operational architectures differ.

5.2. Hybrid IDS Integration Patterns

The reviewed literature reveals three recurring structural patterns, sequential, parallel, and hierarchical integration, each addressing a different operational trade-off. Although sequential and hierarchical integration both process traffic in stages, they differ in what triggers further processing, what the stages contribute, and where the final decision author-

ity lies. These differences are made explicit below, illustrated with a single example flow, and summarised in Table 7.

1 Observation layer	Network flow	Packet / payload	Host / endpoint	Cross-layer
2 AI intelligence	Classical ML	Deep / graph learning	RL / adaptive	Transformer / LLM
3 Fusion pattern	Sequential	Parallel	Hierarchical	Adaptive orchestration
4 Deployment	Centralised	Edge / fog	Federated	Cloud-native / SDN
5 Trust and operations	XAI / confidence	Drift monitoring	Adversarial robustness	Human validation

Figure 2. Multi-axial taxonomy for analysing hybrid AI-based network intrusion detection systems. The axes follow the explicit retention rules in Section 2.5 and Table 2; Supplementary Table S1 gives the complete source disposition. Recent federated, neurosymbolic, GNN, AutoML, programmable-data-plane, transformer, XAI, and multi-scale studies were used as revision-stage coverage tests [20,24–26,31,48,61,62,74,75].

Table 7. Decision-process comparison of sequential, parallel, and hierarchical hybrid integration.

Property	Sequential	Parallel	Hierarchical
Trigger for further processing	Deterministic binary rule outcome (no match)	None; each detector processes all flows	Learned confidence inside an uncertainty band ($\tau_{low} < p < \tau_{high}$)
Nature of the stages	Heterogeneous paradigms (rules, then ML/DL)	Heterogeneous, independent detectors	Usually the same task with increasing capacity, feature depth, or label granularity
Traffic reaching the costly model	Unmatched residual traffic	All traffic	Only uncertain or flagged cases
Final decision authority	First stage that reaches a conclusion	Fusion function (voting, stacking, Bayesian)	Last tier reached; higher tiers may refine or overturn
Typical output	Known-attack label or anomaly score	Fused alert with combined confidence	Alert with confidence, tier, and refined class
Dominant error mode	Rule false positives are final; novel attacks depend on one model	Correlated errors and miscalibrated fusion; full computational cost	Confident tier-1 false negatives are never escalated; threshold mis-tuning
Main tuning parameter	Rule-set content and allow-lists	Fusion weights and decision threshold	Escalation thresholds and tier capacity
Illustrative source	Alqahtani [91]	Prabhu and Sunitha [34]	Chen et al. [78]

In sequential integration, traffic passes through a fixed-order pipeline of heterogeneous detectors, typically a rule-based engine first. Flows matching known signatures are classified immediately and efficiently and leave the pipeline. Only the residual flows that do not match the rule engine proceed to ML or DL analysis. Progression is therefore triggered by a deterministic, binary outcome of the preceding stage (match or no match); each stage applies a different detection paradigm to a different subset of traffic; and the final verdict for a flow is issued by the first stage that reaches a conclusion. The learned stage does not refine a decision made upstream; it covers what the upstream stage cannot recognise. When the first stage also contains allow-list rules for well-characterised benign services, the expensive learned inference applies only to the remaining unexplained traffic; without such rules, most benign traffic still reaches the learned stage, and the efficiency gain is limited to the share of traffic matched by signatures. Alqahtani [91] describes a closely related design for SDN environments: an incremental adaptive random-forest module first detects known attacks, only the instances it labels benign are passed to an adaptive one-class SVM that searches for unknown attacks, and drift detection triggers incremental updates of both modules.

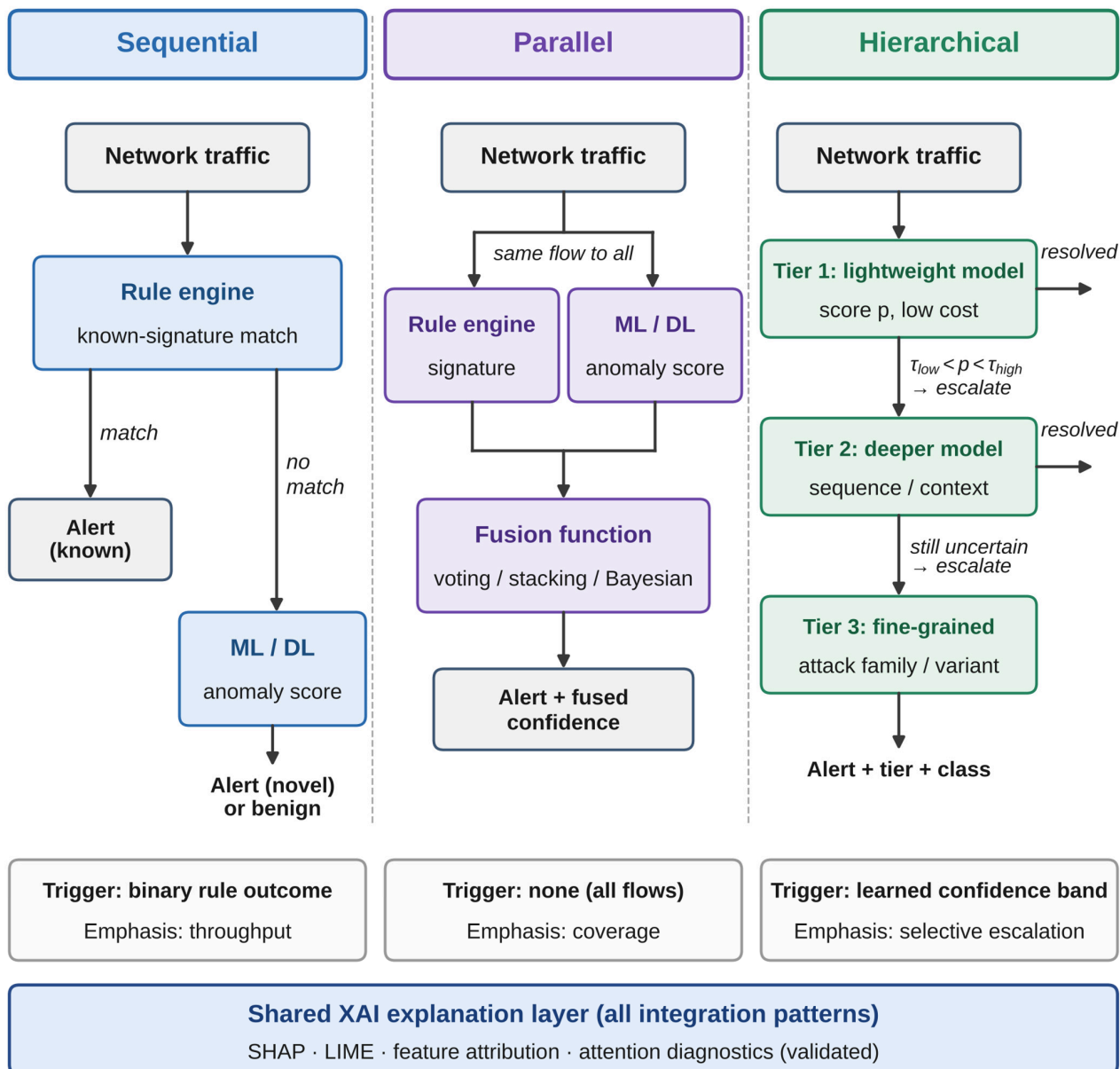
In parallel integration, the rule engine and one or more learned detectors analyse the same traffic simultaneously and independently, and their outputs are combined by a fusion function such as Boolean logic, weighted voting, stacking, or Bayesian combination of calibrated scores. No component filters the input of another, so coverage is maximised at the cost of running every detector on every flow, and the fusion rule becomes the critical design parameter because it trades false negatives against false positives. Prabhu and Sunitha [34] illustrate a representation-level variant in which graph, sequential, and tabular encoders are applied to the same traffic and their embeddings are fused before a gradient-boosted classifier.

Hierarchical integration applies detection in tiers of increasing computational complexity and decreasing traffic volume. A lightweight first tier handles high-volume traffic and escalates ambiguous cases to deeper models. Unlike sequential integration, progression is triggered by the uncertainty of a learned decision rather than by the failure of a rule to match: each tier issues a score or class probability p , and only cases whose confidence falls inside an uncertainty band ($\tau_{\text{low}} < p < \tau_{\text{high}}$) are escalated. The tiers usually share the same detection objective and often the same evidence, but differ in model capacity, feature depth, label granularity (benign versus attack, then attack family, then variant), or location (edge, fog, or core). A higher tier can therefore refine or overturn the provisional decision of a lower tier, and the escalation rate is an explicit tuning parameter that couples detection quality to latency and cost. Chen et al. [78] exemplify granularity-based hierarchy: a binary first stage separates benign from malicious samples, and a multi-class second stage labels only the detected attacks. This architecture can reduce unnecessary inference, but comparable throughput evidence across common hardware and traffic loads remains limited [59].

The difference is easiest to see by following one suspicious flow through both designs—for example, a slow SSH brute-force variant whose packet pattern does not exactly match an existing signature. In a sequential design, the rule engine finds no match, so the flow is forwarded unconditionally to the anomaly model, whose score alone determines the verdict; had the flow matched a brute-force signature, it would have been labelled at the first stage and never seen by the model. In a hierarchical design, a tier-1 model (for example, a compact tree ensemble on flow statistics) assigns the flow an attack probability of 0.55; because this lies inside the illustrative uncertainty band (0.2, 0.9), the flow is escalated to a tier-2 sequence model that examines the timing of repeated authentication attempts and raises the probability to 0.97; this now-confident malicious verdict is passed to a tier-3

classifier that assigns the attack family. A tier-1 probability of 0.05 or 0.98 would instead have been resolved at tier 1 without escalation. Sequential integration therefore partitions traffic by what is already known, whereas hierarchical integration partitions it by how confident the system is (Table 7).

Figure 3 consolidates the three integration patterns described above. The sequential path filters traffic before learned inference, the parallel path combines independent outputs, and the hierarchical path escalates uncertain cases through progressively more capable stages. The arrows are labelled with the condition that triggers further processing, which is the property that most clearly separates the sequential from the hierarchical pattern.



Conceptual synthesis; thresholds are illustrative and no performance ranking is implied.

Figure 3. Conceptual comparison of sequential, parallel, and hierarchical hybrid IDS integration patterns. Arrow labels show the processing trigger: a binary rule outcome in the sequential pattern, simultaneous processing in the parallel pattern, and a learned confidence band ($\tau_{low} < p < \tau_{high}$) in the hierarchical pattern. Illustrative design considerations are informed by adaptive and hybrid IDS examples [34,78,91], hybrid ML/DL evidence [17], and layered IDS limitations [59]; they do not establish universal superiority or derive from a supported quantitative corpus.

The figure makes the principal trade-off explicit: sequential integration favours efficiency, parallel integration favours coverage, and hierarchical integration balances both through selective escalation. This classification concerns component interaction and should not be interpreted as a universal performance ranking.

5.3. Machine Learning Approaches

Random Forest, SVM, and k-NN remain common supervised classifiers in hybrid IDS pipelines, including stacking designs that combine conventional and one-class learners [3,16]. Their reported performance depends strongly on feature engineering, class balance, dataset age, and validation design. Hybridisation can also occur inside the learning pipeline rather than between detectors: Alshinwan et al. [32] combine Prairie Dog Optimisation and Ant Colony Optimisation for feature selection before an SVM classifier evaluated on NSL-KDD. In the five-axis framework, such work belongs to the AI-intelligence axis rather than the fusion axis, because the hybrid element changes how the representation is learned, not how detection outputs are combined.

Chen et al. [78] proposed a two-stage NIDS–HIDS architecture that combines host-event and network-flow features in a collaborative classifier (a binary first stage followed by multi-class labelling of detected attacks). Their reported relative macro-F1 improvements are retained only as source-specific results and are not converted into absolute or pooled performance estimates. Rai et al. [92] likewise discuss optimised tree-based pipelines whose results depend on the selected benchmarks and tuning protocol.

GAN-based augmentation can address minority-class scarcity, but gains must be interpreted under the source study's sampling and validation protocol. Rao and Babu [27] and Allagi et al. [28] illustrate generative augmentation approaches; neither provides a universal estimate of IDS improvement.

5.4. Deep Learning Approaches

CNN–LSTM hybrids are frequently used because convolutional layers can learn local relationships among packet or flow features, while LSTM units model temporal dependencies across sequences. This complementary design is plausible for attacks that manifest through both structural and temporal patterns, but its value must be established separately for each dataset and validation protocol.

Sajid et al. [17] evaluated a CNN–LSTM architecture with feature selection and class balancing across multiple datasets. Its results are treated as source-specific because dataset construction and validation protocols differ across benchmarks.

Jakubcova et al. [33] process numerical and categorical flow features through separate normalisation and embedding paths, then apply a multi-scale feature-interaction block, channel-wise attention, and a bidirectional GRU with attention pooling, with evaluation on CIC-IDS2017 and CSE-CIC-IDS2018. The design shows that recent DL hybrids combine representation learning and short-range temporal modelling within one network; as with other benchmark studies, its reported results remain specific to the datasets and validation protocol used.

Hussein and Répás [19] combined autoencoder-based representation learning, Gaussian mixture modelling, and supervised classifiers, reporting F1 scores above 0.94 and AUC values above 0.97 under their stated protocol. Susilo, Muis, and Sari [18] used a multi-stage autoencoder–LSTM–CNN design for IoT intrusion detection. These results are source-specific and are not pooled.

Dias et al. [93] proposed an interpretable hybrid IDS that combines expert-authored rules with knowledge generated by a decision-tree component. The cited source does not evaluate a graph-neural-network architecture.

5.5. Reinforcement Learning Approaches

Reinforcement learning brings a capability that static classifiers cannot offer: autonomous real-time adjustment of detection policy based on observed outcomes [94,95]. An RL agent embedded in an IDS receives rewards for true positives and penalties for false positives and false negatives, gradually calibrating its detection threshold to the specific traffic distribution it encounters [95]. MAFSIDS integrates deep reinforcement learning with multi-agent feature selection, illustrating how policy learning can be coupled to feature selection [96].

However, claiming that RL-based IDSs can continuously adapt in live production environments without any human intervention or labelled retraining data oversimplifies operational realities [94,97]. In practice, autonomous policy adaptation introduces severe structural challenges as follows:

- **Reward Function Engineering:** Constructing effective reward signals requires balancing opposing security objectives [22,97]. Over-rewarding true positive detections often leads to excessive false alarms, while heavily penalising false positives risks suppressing alerts for critical, low-volume zero-day attacks [22].
- **Exploration Risks in Production Networks:** Standard RL agents rely on exploration policies (e.g., ϵ -greedy) to discover optimal detection strategies [98]. Executing exploratory or sub-optimal actions on live production networks can cause catastrophic operational disruption, such as dropping legitimate mission-critical traffic or temporarily exposing endpoints to unchecked lateral movement [97,98].
- **Non-Stationary Traffic Dynamics:** Network traffic distributions are inherently non-stationary due to shifting user behaviour, emerging network protocols, and active adversarial evasion [99]. Non-stationary state spaces can cause RL policy convergence failure, leading to policy instability, rapid performance decay, or severe policy drift [97,99].
- **Necessity of Human-in-the-Loop Governance:** Autonomous policy updates without safety constraints create severe security vulnerabilities [97,100]. Real-world RL deployment necessitates human-in-the-loop governance, including constrained policy spaces, shadow execution modes, safety bounds, and periodic auditing by SOC analysts, to validate policy shifts before committing updates to active traffic enforcement [100,101].

5.6. Transformer and LLM-Augmented Approaches

Transformer self-attention can model long-range dependencies across packet or flow sequences in parallel, offering an alternative to recurrent processing. FlowTransformer provided a framework for comparing transformer variants in flow-based NIDS [24], while Xi et al. combined multi-scale feature extraction with transformer modelling [75]. In the five-axis framework, these studies affect both the observation representation and the AI-intelligence mechanism. They do not, by themselves, establish operational superiority: latency, memory, temporal generalisation, and zero-day claims still require evaluation on realistic and temporally separated traffic.

5.7. Federated and Privacy-Preserving Approaches

In healthcare, financial services, and multi-tenant cloud environments, the most valuable intrusion data is often the least shareable. Raw network traffic may contain patient records, financial transactions, or proprietary business communications that cannot legally or contractually leave the organisation. Federated learning (FL) addresses this tension by coordinating model training through the exchange of gradient updates or model parameters between participating nodes, ensuring raw traffic data remains localised on the collection node [102,103].

However, claiming that keeping raw data local inherently guarantees data privacy overstates FL's security boundary and creates analytical inconsistency across literature. Local data retention alone is insufficient to guarantee privacy in collaborative IDS environments. Without privacy-preserving augmentations, standard federated architectures remain vulnerable to three critical attack vectors [103,104]:

- **Gradient Inversion and Reconstruction Attacks:** Malicious aggregation servers or eavesdroppers can reconstruct sensitive raw network flow features, packet payloads, or topological IP metadata directly from shared model gradients or parameter updates during communication rounds [105,106].
- **Membership Inference Attacks:** Adversaries analysing global model updates across communication epochs can infer whether specific target IP addresses, proprietary traffic sequences, or specialised network traces were present in a participant's local training set [107,108].
- **Model- and Data-Poisoning Attacks:** Distributed execution exposes the pipeline to rogue or compromised client nodes injecting poisoned local gradient updates to deliberately degrade global detection performance or implant covert evasion backdoors [109,110].

Huang et al. [111] combined hybrid deep learning with federated learning to preserve data locality across distributed sensors. Recent work on lightweight federated NIDS [26] and semi-supervised federated IoT intrusion detection [74] has further illustrated that deployment efficiency and label scarcity are separate design problems. However, broader privacy claims must remain distinct from formal privacy guarantees unless explicit defence mechanisms and parameters are evaluated. Provable guarantees require additional mechanisms such as differential privacy with a declared privacy budget [112,113], secure aggregation [114], or appropriately specified cryptographic protocols [115].

5.8. Explainable AI in Hybrid IDSs

Black-box detection decisions create two distinct problems in practice. Security analysts cannot validate them without understanding them, which means a highly accurate DL system may be trusted less than a less accurate but interpretable one, a real paradox in production deployment. And organisations subject to GDPR Article 22, HIPAA security requirements, NIS2 accountability provisions, or DORA operational resilience obligations may face legal barriers to deploying automated security decisions without explainability provisions. Both problems are motivating a growing body of work on XAI for IDS.

Hermosilla et al. [29] compared SHAP and LIME for intrusion-detection analysis, highlighting differences in explanation behaviour and forensic usefulness. Alabbadi and Bajaber [116] likewise applied explainability methods to IoT data streams. E-XAI extends the recent evidence base by evaluating black-box explanation frameworks specifically for NIDS [31]. These studies support the need to evaluate explanation fidelity and analyst utility; they do not justify treating any single explanation method as universally reliable.

Hybrid IDSs create an additional challenge beyond the general XAI problem. When a detection results from the combined output of a rule engine, an ML classifier, and a DL anomaly scorer, an explanation that addresses only one of those components is incomplete. Reconciling contributions from three heterogeneous sources into a unified, analyst-readable output is a problem the field has only begun to address. Transformer attention visualisation offers one promising path: attention weights over packet sequence features provide a natural indication of which specific traffic events most influenced the decision, without the post hoc approximation that SHAP and LIME require.

While transformer attention mechanisms provide intuitive spatial and temporal saliency maps over packet sequences, raw attention weights must not be naively equated

with causal feature importance [117,118]. Empirical XAI literature demonstrates that attention weights can correlate weakly with actual model decision outputs, as intermediate representations and non-linear projections redistribute feature influence downstream. Consequently, relying solely on raw attention matrices risks generating misleading security explanations. To establish faithful interpretability, attention-based IDSs should pair intrinsic attention visualisations with post hoc attribution or gradient-based techniques, such as Integrated Gradients [119], Attention Rollout, or Layer-wise Relevance Propagation (LRP), to verify causal feature attribution before presenting outputs to SOC analysts.

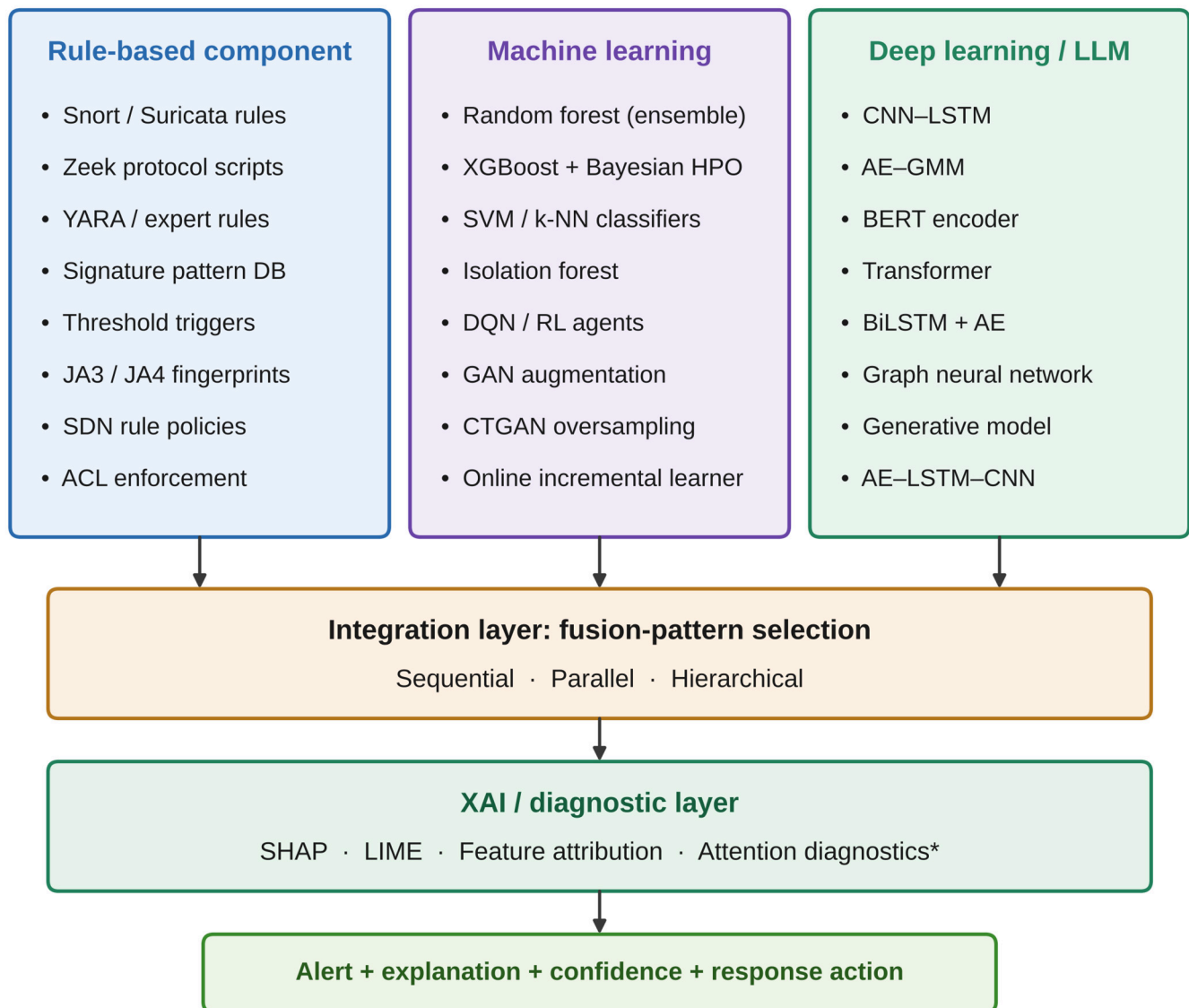
Table 8 compares representative architectures by integration pattern, core technique, source-specific result, and stated limitation. It is intended to expose design diversity and evidence boundaries, not to rank models trained under incompatible protocols.

Table 8. Comparative summary of hybrid AI-IDS architectures (2020–2026).

Reference	Architecture	Key Technique	Key Result	Limitation
Sajid et al. [17]	Hybrid DL + rules	CNN–LSTM; XGBoost feat. sel.	Dataset-specific CNN–LSTM evaluation	Dataset- and protocol-specific evaluation
Chen et al. [78]	Hierarchical two-stage ML (NIDS + HIDS)	Host-feature flattening; binary then multi-class classifier	+8.1% macro F1 vs. baseline on CICIDS 2018	Relative result; not pooled
Hussein & Répás [19]	Three-tier hybrid	AE–GMM + XGBoost + LR	F1 > 0.94; AUC > 0.97 under the reported protocol	No adversarial testing
Susilo et al. [18]	DL Multi-stage IoT	AE → LSTM → CNN	Multi-stage AE–LSTM–CNN evaluation	IoT datasets only
Yang & Peng [23]	BERT-based	Bidirectional encoder	BERT-based traffic classification	No adversarial test
Hermosilla et al. [29]	XAI-enhanced	XGBoost + SHAP + LIME	Comparative SHAP/LIME analysis	Interpretability– accuracy trade-off
Rao & Babu [27]	GAN-augmented	IGAN + LeNet-5/LSTM	GAN-based minority-class augmentation	Synthetic realism limits
MAFSIDS [96]	RL-based hybrid	DQN + multi-agent feat. selection	DRL-based multi-agent feature selection	FPR not reported in source
Alshinwan et al. [32]	Hybrid feature selection	PDO–ACO feature selection + SVM	About 98% accuracy on NSL-KDD (source-reported)	Legacy dataset; single classifier
Jakubcova et al. [33]	Hybrid DL (multi-scale + temporal)	Embeddings + multi-scale interaction + channel attention + BiGRU	CIC-IDS2017 and CSE-CIC-IDS2018 evaluation	Offline benchmark evaluation
Prabhu and Sunitha [34]	Parallel representation fusion	HGT + LSTM + TabTransformer embeddings; generative augmentation; XGBoost	+3.7% F1 and 38% lower FPR vs. baselines; <5% loss under FGSM/PGD (source-reported)	Synthetic augmentation; dataset-specific results

Table 8 highlights the diversity of hybrid AI-IDS designs and the limited comparability of their reported outcomes. The studies differ in datasets, tasks, model combinations, and validation protocols; therefore, the table should be read as an architectural comparison rather than a performance ranking.

Figure 4 brings the reviewed AI paradigms into a common architecture. Machine learning, deep learning, reinforcement learning, transformers, federated learning, and generative models provide different intelligence functions, while the integration and explainability layers connect these functions to operational IDS decisions.



*Attention weights require independent validation and are not causal explanations.

Figure 4. Architectural overview of rule-based, machine learning, and deep learning/LLM components connected through integration and explanation layers. Representative source families include CNN-LSTM [17], AE-GMM [19], AE-LSTM-CNN [18], BERT-based detection [23], and SHAP/LIME-based IDS analysis [29,116].

As illustrated in Figure 4, these paradigms are complementary rather than mutually exclusive. Their practical value depends on how they share evidence, where training and inference occur, and whether explanations, robustness monitoring, and human validation are incorporated into the deployment pipeline.

5.9. Worked Example: Applying the Five-Axis Taxonomy to Table 8's Architecture

To show how the five axes jointly describe a system, Table 9 places the two-stage NIDS-HIDS architecture of Chen et al. [78] (Table 8) on every axis. This system was selected because its published description contains enough architectural detail to populate all five

axes and because it combines more than one evidence source. The placement uses only information reported in the source; where an attribute is not reported, the cell states this explicitly instead of inferring it.

Table 9. Five-axis characterisation of the hybrid NIDS–HIDS architecture of Chen et al. [78].

Axis	Design Question	Placement of Chen et al. [78]	What the Placement Reveals
Observation layer	What evidence can the detector observe?	Cross-layer: network-traffic features from the NIDS view combined with host-resource evidence (logs, files, and folders); two-dimensional host features are flattened into one-dimensional vectors; evaluated on CSE-CIC-IDS2018 and NDSec-1	Requires both a network sensor and host agents; host evidence can compensate for limited network visibility (e.g., encrypted traffic), but coverage depends on host logging policy
AI intelligence	What representation or decision mechanism is learned?	Supervised classical ML on tabular features, organised as a two-stage collaborative classifier; XGBoost serves as the conventional baseline	Interpretable feature space; no online adaptation or reinforcement learning; zero-day capability limited to supervised generalisation
Fusion pattern	How are components or evidence combined?	Early (feature-level) fusion of host and network evidence, followed by a hierarchical, granularity-based decision: stage 1 separates benign from attack; stage 2 assigns the attack class	Hierarchical rather than sequential: progression depends on a learned verdict, not on a rule mismatch; stage-1 false negatives are never re-examined
Deployment	Where do training, inference, and coordination occur?	No operational topology is specified; evaluation is offline on benchmark datasets, implying centralised collection of host and network features	Line-rate latency, host-agent overhead, and edge or federated variants remain to be evaluated
Trust and operations	How are explanation, robustness, drift, and analyst control governed?	Not reported in the published description: no explanation layer, adversarial test, temporal-drift evaluation, or analyst-workflow assessment	Reported macro-F1 gains (+8.1% on CSE-CIC-IDS2018; +3.7% on NDSec-1) are relative benchmark results, not evidence of operational trustworthiness

Read together, the five placements characterise the system far more completely than its classifier label. Two systems that both use gradient-boosted trees would be positioned differently if one observed only encrypted-flow statistics at an edge gateway and the other fused host and network evidence centrally. The example also shows the diagnostic value of the taxonomy: unreported cells on the deployment and trust/operations axes identify exactly the evidence that must be generated before the reported benchmark gains can support an operational recommendation. The same procedure can be applied to every architecture in Table 8—for instance, Prabhu and Sunitha [34] would be placed on the AI-intelligence axis as graph, sequential, and tabular deep learning; on the fusion axis as parallel representation-level fusion; and on the trust/operations axis as partially addressed through SHAP explanations and FGSM/PGD robustness testing.

6. Open Challenges and Research Roadmap

The critical synthesis identifies six recurring deployment challenges. They are presented without ordinal priority labels because the literature does not provide a common scoring scale or comparable operational evidence from which a defensible ranking could be calculated. Consistent with Section 2.8, they are stated as challenges that are under-reported or unresolved in the accessible review literature and corroborated by primary or methodological evidence, rather than as proven absences; Table 10 adds proposed evaluation criteria for each.

Table 10. Research roadmap of six open challenges with evidence, mitigations, and proposed evaluation criteria.

Challenge	Evidence	Current Practice	Emerging Mitigations	Proposed Evaluation Criteria
Adversarial robustness	WGAN-generated adversarial traffic	Adversarial training; ensemble diversity	Certified robustness; feature squeezing; denoising AE	Detection and attack-success rates under realistic, constraint-respecting perturbations with a stated attacker budget; clean-versus-robust performance gap; cost of the defence [120]
Encrypted traffic	TLS 1.3, DoH, and QUIC reduce payload visibility [6]	JA3/JA4 fingerprinting; flow statistics	Behavioural profiling; CT log monitoring	Per-attack-family recall and FPR with each indicator ablated; feature-availability rate per protocol (TLS 1.2/1.3, QUIC, DoH) and sensor location; temporally separated test sets [58,121]
Interpretability	Black-box decisions impede analyst validation [29]	SHAP/LIME analysis of ML classifiers	Attention visualisation; rule extraction from DL	Explanation fidelity (deletion/insertion tests), stability, and robustness [122]; analyst triage time, decision accuracy, and trust calibration in user studies [123]
Data imbalance	APT/zero-day severely underrepresented [27]	IGAN [27]; CTGAN [28]	Diffusion synthesis; transfer learning	Per-class recall, macro-F1, and precision–recall AUC for rare attacks; train-on-synthetic, test-on-real realism checks
Computational scalability	DL latency and high-rate traffic processing	Quantisation and staged inference	SmartNIC; knowledge distillation	Sustained throughput (flows/s or Gbit/s), p95/p99 latency, and packet-drop rate at line rate; memory and energy per inference on declared hardware
Federated convergence	Non-IID degrades FedAvg quality	FedAvg + local tuning	FedProx; per-FedAvg; differential privacy	Rounds and transmitted bytes to reach a target macro-F1; worst-client versus mean-client performance under non-IID splits; privacy budget (ϵ , δ); robustness to poisoned clients [124]

6.1. Adversarial Robustness

Chauhan et al. [125] demonstrated that WGANs can generate polymorphic adversarial traffic, reinforcing the need for adaptive-threat evaluation. Adversarial training, diverse ensembles, and certified robustness are promising responses, but their cost and transferability must be measured explicitly.

6.2. Encrypted-Traffic Analysis

TLS 1.3, DNS-over-HTTPS, and QUIC reduce payload visibility [126–128]. Flow statistics, TLS fingerprints, behavioural profiles, certificate context, and endpoint evidence can provide complementary signals, but each must be evaluated by attack family and

under temporal shift [6,58]. Their availability also differs by protocol version and by sensor placement, as summarised in Section 8.

6.3. Model Interpretability, Explainability, and Regulatory Compliance

In regulated enterprise environments, model interpretability and explainability are no longer optional architectural features; they are critical operational and legal requirements [45,129]. However, the regulatory drivers for explainable AI (XAI) in IDS deployment fall into two distinct legal categories:

- **Direct Legal Mandates:** Frameworks such as Article 22 of the EU General Data Protection Regulation (GDPR) establish a direct legal right to meaningful information regarding the logic involved in automated decision-making [129]. When an automated IDS triggers active mitigation (e.g., blocking traffic, terminating sessions, or isolating user endpoints) without human intervention, black-box classification creates direct legal liability and non-compliance risk [130].
- **Indirect Operational Requirements:** Directives and standards, such as NIS2 (Network and Information Security Directive) [131], DORA (Digital Operational Resilience Act) [132], and HIPAA (Health Insurance Portability and Accountability Act), do not explicitly mandate specific XAI algorithms. Instead, they mandate comprehensive security risk documentation, auditability, incident root-cause analysis, and evidence of operational resilience [45,131,132]. In these environments, black-box detection creates severe compliance exposure because security operations centre (SOC) analysts cannot produce transparent, auditable forensic trails during regulatory audits or post-incident reviews [45,131].

Beyond formal regulatory compliance, analysts require interpretable outputs to triage high-volume alerts effectively and minimise alert fatigue [8,45]. Hermosilla et al. [29] evaluated post hoc explanation frameworks in network security, demonstrating that SHAP generates globally more consistent and forensically useful explanations than LIME, with feature attributions aligning significantly better with domain expert knowledge.

Despite these advances, a fundamental challenge remains in hybrid IDS architectures: explanation reconciliation [30]. Merging disparate attribution outputs from a rule engine (deterministic matches), an ML classifier (statistical feature weights), and a DL anomaly scorer (latent representations) into a unified, analyst-readable output remains largely unsolved [30]. Intrinsic model visualisation via transformer attention mechanisms offers a promising path forward by highlighting specific packet sequence features and temporal windows [133]. However, raw attention weights do not always correlate directly with causal feature importance [117]. To prevent misinterpretation during incident response, intrinsic attention visualisations must be combined with post hoc attribution methods, such as Integrated Gradients [119] or Attention-SHAP, to ensure that highlighted packet features and temporal windows accurately reflect true model decision dependencies.

6.4. Data Imbalance

As discussed in Section 5.3, generative augmentation approaches [27,28] can mitigate minority-class scarcity, but their gains remain specific to each study's sampling and validation protocol. Future studies should therefore report per-class recall and precision–recall AUC for rare attack classes and verify the realism of synthetic samples (Table 10).

6.5. Computational Scalability

Reduced-precision inference can lower memory and arithmetic costs, but IDS studies should report accuracy, latency, throughput, energy use, hardware, and quantisation settings together.

6.6. Federated Learning Convergence Under Non-IID Data

In federated IDS deployments spanning multiple organisations, each node observes traffic shaped by its specific applications, users, and attack exposure. This non-IID distributional heterogeneity causes standard FedAvg to produce a global model that performs poorly for many participants. Heterogeneity-aware algorithms, such as FedProx [124], and personalised variants, such as Per-FedAvg, allow nodes to maintain locally adapted models while still benefiting from the global gradient, addressing heterogeneity without abandoning collaboration entirely. Differential privacy mechanisms add calibrated noise to gradient updates before transmission, providing provable guarantees against membership inference at a modest accuracy cost [112,113]. Gradient compression reduces communication overhead, which matters when nodes are connected over bandwidth-constrained links.

Table 10 links each deployment challenge to current practices and candidate mitigations and proposes criteria by which each mitigation can be evaluated. The table is a non-ranked research map; it does not imply that one challenge is universally more important than another.

Across all six challenges, we further propose that results be reported under temporally separated train–test splits, with performance tracked as a function of time since training (for example, the time-aware evaluation and area-under-time summary proposed in TESSER-ACT [121]) and with drift detection and retraining costs recorded [14]. These criteria are proposed evaluation standards for future studies rather than experiments performed in this review; their purpose is to make each mitigation in Table 10 testable and comparable across studies.

7. Discussion

7.1. Architecture Selection and Enterprise Infrastructure Integration

Sequential integration can reduce computational load by filtering traffic before expensive inference; parallel integration prioritises coverage by combining independent evidence; and hierarchical integration allocates complex cases to progressively stronger models. The appropriate pattern depends on traffic rate, attack prevalence, analyst capacity, hardware budget, and acceptable latency. The reviewed literature does not support a universal throughput threshold at which one architecture becomes optimal.

Technology readiness differs across paradigms. Classical ensembles and CNN/RNN pipelines have the largest benchmark evidence base. Transformer, reinforcement learning, federated, and LLM-assisted approaches are promising but have less consistent evidence under concept drift, non-IID data, adversarial conditions, and real network throughput. Maturity should therefore be judged by reproducibility and operational validation, not by model novelty.

Operational value cannot be inferred from benchmark accuracy alone. A defensible deployment assessment should report alerts per analyst-hour, time to detection, inference throughput, resource consumption, calibration error, drift retraining frequency, and the cost of false positives and false negatives. Organisation-specific ROI figures require primary operational data and are outside the evidence available in this review.

In production enterprise environments, an intrusion detection system (IDS) rarely functions as a standalone security solution. IDS deployments are heavily shaped by legacy IT infrastructure, requiring modern AI-based systems to operate as integrated components within broader security ecosystems alongside perimeter firewalls; Security Information and Event Management (SIEM) platforms; and Security Orchestration, Automation, and Response (SOAR) frameworks [134].

7.1.1. Enterprise Infrastructure Integration and Telemetry Aggregation

Legacy perimeter firewalls execute high-throughput, rule-based filtering at enterprise ingress points but lack adaptive anomaly detection. AI-based IDSs complement these static barriers by analysing internal lateral movement and complex traffic patterns downstream. However, placing AI models behind encrypted boundaries (e.g., TLS 1.3 or QUIC) restricts payload inspection, requiring features to be extracted from flow-level statistics and TLS metadata (e.g., JA3/JA4 fingerprints) [135].

Telemetry feeds rely on physical Network Taps or Switched Port Analyzer (SPAN) ports. On high-speed links exceeding 10 Gbps, SPAN port oversubscription can cause packet dropping, which corrupts flow-sequence integrity and impairs temporal deep learning models (e.g., CNN-LSTM or transformer architectures).

AI-based IDSs act as telemetry providers for centralised SIEM platforms, which correlate network anomalies with host logs. When integrated with SOAR engines for automated threat response, AI models must output structured alert formats (e.g., STIX/TAXII or JSON) coupled with calibrated confidence scores to prevent false-positive alerts from triggering disruptive automated containment actions.

7.1.2. Operational Latency Overheads and SOC Analyst Workflows

Deep learning architectures introduce inference delays compared to $O(n)$ payload pattern matching in traditional signature engines. On high-throughput links, complex model execution can create queuing bottlenecks and buffer overflows unless managed through sequential filtering, hierarchical escalation, or hardware offloading (e.g., SmartNICs) [136].

Uncalibrated anomaly detection engines often produce high false-positive rates (FPR), causing severe alert fatigue among security operations centre (SOC) analysts. To integrate effectively into human-in-the-loop workflows, AI alerts must include Explainable AI (XAI) layers, such as SHAP/LIME feature attributions or visual attention maps, to provide analysts with interpretable evidence for rapid incident triage and validation [123].

7.2. What the Evidence Means in Practice

The reviewed evidence does not support a single pooled performance gap between signature, anomaly, and hybrid IDS categories. Benchmark values are protocol-dependent and should be interpreted study by study.

Hybrid systems can combine known-attack precision with learned anomaly detection, but the magnitude of any benefit depends on the fusion rule, dataset, threshold selection, and validation protocol.

For security architects, the taxonomy provides conditional rather than universal guidance. Sequential integration is appropriate when throughput is the principal constraint; parallel integration prioritises coverage at higher computational cost; and hierarchical integration selectively escalates ambiguous traffic. The decision-level distinctions in Table 7 suggest a practical selection rule: where most malicious traffic is already well characterised by signatures, sequential integration exploits that knowledge; where uncertainty rather than novelty dominates, hierarchical escalation allocates model capacity more effectively. The choice must be validated under the target traffic rate, hardware, attack prevalence, latency budget, and analyst capacity because the reviewed evidence does not establish a universally superior architecture.

7.3. Data Integrity and Corrections

Source verification identified multiple bibliographic and attribution errors in the draft corpus. Quantitative pooling was therefore removed, and only traceable, source-specific claims were retained. In this revision, further attributions were corrected (the AIT log-

dataset source [80], the host-data survey [81], the CTGAN augmentation study [28], and the description of [91]), and preprint citations were replaced by their peer-reviewed versions wherever such versions exist [56,60,105–110,113,117,119].

Sajid et al. [17] report dataset-specific results for their proposed pipeline. These values should remain attached to the corresponding task and validation protocol rather than being pooled with heterogeneous studies.

7.4. Limitations

To ensure analytical clarity, limitations are separated into limitations of the systematic review process and limitations of the underlying AI-NIDS literature.

7.4.1. Methodological Limitations of This Review

- Information-source coverage: The reproducible revision search used OpenAlex and arXiv rather than direct exports from subscription databases. Although OpenAlex indexes records from many publishers, the review may miss records that are absent or poorly indexed. Non-English reports were excluded.
- Retrieval and reviewer limitations: Seventy-six of 101 sought reports were not retrieved through the documented routes, creating access-related selection risk. One reviewer performed screening, extraction, and appraisal; no independent duplicate screening, arbitration, or inter-rater statistic was available. Deterministic text rules improved reproducibility but may have produced false exclusions where titles and abstracts used unusual terminology.
- Protocol and synthesis boundaries: The review was not prospectively registered. The revision-stage protocol and eligibility amendment are reported, but retrospective rebuilding cannot reproduce the chronology of the earlier narrative review. The PRISMA corpus comprises secondary reviews only; primary studies, standards, and regulations are separate illustrative or contextual sources. No statistical pooling, reporting-bias test, or certainty-of-effect grading was performed.
- Source type: three included reviews [13,42,43] are available only as preprints. They were retained because the prespecified arXiv source and eligibility rules admitted preprints without an archival version, as excluding them after the search would have introduced selection bias. Their topics (adversarial learning and federated or quantum learning) are also covered by peer-reviewed sources in the corpus and in the contextual literature [12,25,44,120,125], so no axis of the taxonomy depends on preprint evidence alone (Table 2).

7.4.2. Limitations of Included Reviews and the Primary Literature

- Benchmark Dataset Deficiencies: A significant portion of the primary literature continues to rely on legacy or simulated datasets (e.g., NSL-KDD) that exhibit severe class imbalance, redundant records, and obsolete threat vectors. These benchmarks fail to capture contemporary cloud-native behaviour, encrypted payloads (TLS 1.3, DoH, QUIC), or multi-stage APT scenarios.
- Non-Standardised Split Protocols: Primary studies frequently employ random train-test splits or uncalibrated cross-validation without temporal separation, introducing data leakage and optimistic bias that artificially inflate baseline accuracy.
- Lack of Operational Latency and Drift Evaluation: The existing literature heavily prioritises offline classification metrics (e.g., accuracy, F1-score) while omitting crucial deployment parameters, including line-rate inference latency, memory consumption, throughput under traffic spikes, model drift adaptation, and resilience under active adversarial manipulation.

7.5. Comparison with Related Reviews

The PRISMA corpus confirms that prior reviews commonly concentrate on algorithms, datasets, adversarial learning, class imbalance, or individual deployment paradigms [12,13,36,37,39–44,49,53–56,60,62,68–72]. Contextual surveys by Khraisat et al. [1], Maseno et al. [35], Jacob and Sultana Habibullah [3], and Sowmya and Anita [38] remain useful for broader historical positioning, as do earlier broad surveys of ML and DL for intrusion detection [2,4,5] and methodological critiques of ML-based security evaluation [9,11,15]. Recent specialised taxonomies address collaborative federated IDS [25], neurosymbolic NIDS [61], GNN-NIDS [20], AutoML-NIDS [62], and programmable-data-plane NIDS [48]. The present taxonomy aligns these specialised concerns; it does not supersede them.

The novelty claim is architectural and testable: a system should be describable across observation, intelligence, fusion, deployment, and trust/operations without redefining an axis. The systematic review establishes the recurring design questions, while separately labelled transformer, XAI, federated, and multi-scale primary studies [24,26,31–34,74,75] stress-test category placement. This distinction prevents primary examples from being misrepresented as part of the 22-review PRISMA corpus.

8. Conclusions

This systematic review of 22 secondary AI-NIDS reviews argues that intrusion detection should be evaluated as an operational architecture rather than as an isolated classifier. The five-axis taxonomy links observation layers, learning paradigms, fusion patterns, deployment topologies, and trust mechanisms through explicit retention rules and a source-to-axis audit. Its practical contribution is a consistent set of questions for comparing heterogeneous systems; it does not establish that one architecture is universally superior.

Sequential integration can reduce average inference demand by routing known-signature matches through a rule engine before learned analysis, but the magnitude of any latency benefit must be measured under the target traffic mix and hardware. Parallel integration can broaden evidence coverage by combining rule and anomaly outputs, at the cost of additional computation and calibration. Hierarchical integration can reserve more complex models for ambiguous cases, but its benefit depends on escalation thresholds, error propagation, and tier-specific latency. Sequential and hierarchical integration should therefore not be conflated: the former advances traffic when a deterministic rule fails to match, whereas the latter escalates cases whose learned confidence is insufficient. These are conditional design implications, not universal speed or detection guarantees.

For encrypted traffic, the evidence supports a set of complementary indicators rather than a single remedy. Flow statistics (packet sizes, inter-arrival times, burst structure, and byte volumes) remain observable for TLS, QUIC, and DoH and support behavioural detection of, for example, beaconing, exfiltration volume, or scanning [6,58]. Certificate context (issuer, validity period, self-signed status, or subject names) characterises the server and can expose malicious infrastructure, but it is visible on the wire only for TLS 1.2 and earlier; under TLS 1.3 the certificate is encrypted, so this context must come from a TLS-terminating proxy, endpoint telemetry, or external sources such as Certificate Transparency logs [126]. JA3 and JA4 fingerprints summarise client hello parameters to characterise the client implementation and can link traffic to known tools or malware families, but they identify software rather than intent, are shared by benign applications using the same libraries, can be imitated, and are degraded when clients randomise extension order, a limitation that JA4 partly mitigates by sorting these fields. For QUIC, the handshake is carried in Initial packets that a passive sensor must first decode [127], and encrypted client hello mechanisms further reduce visibility. DoH queries are indistinguishable from other

HTTPS traffic to a perimeter sensor and can be analysed only through flow behaviour, resolver-endpoint policy, or endpoint logs [128]. The availability of each indicator therefore depends on the protocol and its version and on the observation point (perimeter tap, internal SPAN port, TLS-terminating proxy, or endpoint). These indicators should be treated as supporting evidence to be fused with other observation layers and validated per attack family and under temporal shift, not as a universal substitute for payload inspection.

In resource-constrained and edge topologies, such as in bandwidth-sensitive, IoT, or SDN environments where monolithic deep learning creates severe latency bottlenecks, deployment must adopt quantized tree ensembles or autoencoders integrated into sequential or federated frameworks, utilising algorithms such as FedProx to maintain convergence under non-IID data distributions. For high-assurance SOC integration, in production enterprise environments subject to regulatory auditing and analyst alert fatigue, black-box classifiers create unacceptable operational risk. Hybrid IDSs must pair parallel or hierarchical fusion engines with shared XAI explanation layers (e.g., SHAP, LIME, or attention attribution) to supply SOC analysts and SOAR platforms with interpretable evidence and calibrated confidence scores.

Benchmark accuracy on historical datasets is insufficient for production deployment. Trustworthy AI-NIDS deployment requires transparent validation protocols, temporal-shift evaluation, operational metrics, and alignment with enterprise architecture. These conclusions are supported as a bounded qualitative synthesis; their confidence is graded in Table 4, Table 10 provides criteria through which they can be tested, and they should be updated when broader database access, duplicate independent screening, and additional retrievable reviews become available.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/computers15100726/s1>, Supplementary Methods S1–S3 and Tables S1–S2 provide the full search strategies, eligibility amendment, appraisal rules, included-review matrix, and full-text exclusion log. The completed PRISMA 2020 checklist and record-level screening ledger are also provided as Supplementary Materials.

Author Contributions: Conceptualization, A.H. and M.H.H.; methodology, A.H.; software, S.A.W.; validation, A.H. and M.H.H.; formal analysis, A.H. and S.A.W.; investigation, S.A.W. and A.H.; data curation, S.A.W. and A.H.; writing—original draft preparation, A.H.; writing—review and editing, M.H.H.; visualization, A.H.; supervision, A.H.; project administration, A.H.; funding acquisition, M.A.S.A.-H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Arab Open University Oman through the Research Initiative Grant Scheme—Grant No. AOU_OM/2025/FCS5.

Data Availability Statement: The original contributions presented in this study are included in the article and its Supplementary Materials, which contain the search strategies, screening ledger, retrieval log, full-text exclusion reasons, appraisal matrix, and completed PRISMA 2020 checklist. The raw API responses underlying the search are available from the corresponding authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
2. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1153–1176. [CrossRef]
3. Jacob, S.L.; Habibullah, P.S. A systematic analysis and review on intrusion detection systems using machine learning and deep learning algorithms. *J. Comput. Cogn. Eng.* **2025**, *4*, 108–120. [CrossRef]

4. Liu, H.; Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* **2019**, *9*, 4396. [\[CrossRef\]](#)
5. Ahmad, Z.; Shahid Khan, A.; Wai Shiang, C.; Abdullah, J.; Ahmad, F. Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [\[CrossRef\]](#)
6. Papadogiannaki, E.; Ioannidis, S. A survey on encrypted network traffic analysis: Applications, techniques, and countermeasures. *ACM Comput. Surv.* **2021**, *54*, 1–35. [\[CrossRef\]](#)
7. Janabi, A.H.; Kanakis, T.; Johnson, M. Survey: Intrusion detection system in software-defined networking. *IEEE Access* **2024**, *12*, 164097–164120. [\[CrossRef\]](#)
8. Tariq, S.; Chhetri, M.B.; Nepal, S.; Paris, C. Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities. *ACM Comput. Surv.* **2025**, *57*, 224. [\[CrossRef\]](#)
9. Sommer, R.; Paxson, V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In Proceedings of the 2010 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 16–19 May 2010; pp. 305–316. [\[CrossRef\]](#)
10. Engelen, G.; Rimmer, V.; Joosen, W. Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 27 May 2021; pp. 7–12. [\[CrossRef\]](#)
11. 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.
12. Al-Qarni, E.A.; Al-Asmari, G.A. Addressing Imbalanced Data in Network Intrusion Detection: A Review and Survey. *Int. J. Adv. Comput. Sci. Appl.* **2024**, *15*, 136. [\[CrossRef\]](#)
13. Alatwi, H.A.; Morisset, C. Adversarial Machine Learning in the Network Intrusion Detection Domain: A Systematic Review. *arXiv* **2021**, arXiv:2112.03315. [\[CrossRef\]](#)
14. Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A. A Survey on Concept Drift Adaptation. *ACM Comput. Surv.* **2014**, *46*, 1–37. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Apruzzese, G.; Laskov, P.; Montes de Oca, E.; Mallouli, W.; Brdalo Rapa, L.; Grammatopoulos, A.V.; Di Franco, F. The Role of Machine Learning in Cybersecurity. *Digit. Threats Res. Pract.* **2023**, *4*, 1–38. [\[CrossRef\]](#)
16. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree and one-class SVM. *Electronics* **2020**, *9*, 173. [\[CrossRef\]](#)
17. Sajid, M.; Malik, K.R.; Almogren, A.; Malik, T.S.; Khan, A.H.; Tanveer, J.; Rehman, A.U. Enhancing intrusion detection: A hybrid machine and deep learning approach. *J. Cloud Comput.* **2024**, *13*, 123. [\[CrossRef\]](#)
18. Susilo, B.; Muis, A.; Sari, R.F. Intelligent intrusion detection system against various attacks based on a hybrid deep learning algorithm. *Sensors* **2025**, *25*, 580. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Hussein, S.A.; Répás, S.R. A Hybrid Intrusion Detection Framework Using Deep Autoencoder and Machine Learning Models. *AI* **2026**, *7*, 39. [\[CrossRef\]](#)
20. Ismaeli Taba, M.R.; Zare, H.; Rakhshani, S. Graph neural networks for network intrusion detection systems: A comprehensive survey, taxonomy, and empirical evaluation. *Artif. Intell. Rev.* **2026**, *in press*. [\[CrossRef\]](#)
21. Louati, F.; Ktata, F.B.; Amous, I. Enhancing intrusion detection systems with reinforcement learning: A comprehensive survey. *SN Comput. Sci.* **2024**, *5*, 665. [\[CrossRef\]](#)
22. Yang, W.; Acuto, A.; Zhou, Y.; Wojtczak, D. A Survey for Deep Reinforcement Learning Based Network Intrusion Detection. *Appl. AI Lett.* **2026**, *7*, e70026. [\[CrossRef\]](#)
23. Yang, Y.; Peng, X. BERT-based network for intrusion detection system. *EURASIP J. Inf. Secur.* **2025**, *2025*, 11. [\[CrossRef\]](#)
24. Manocchio, L.D.; Layeghy, S.; Lo, W.W.; Kulatilleke, G.K.; Sarhan, M.; Portmann, M. FlowTransformer: A transformer framework for flow-based network intrusion detection systems. *Expert Syst. Appl.* **2024**, *241*, 122564. [\[CrossRef\]](#)
25. Wardana, A.A.; Sukarno, P. Taxonomy and survey of collaborative intrusion detection system using federated learning. *ACM Comput. Surv.* **2024**, *57*, 1–36. [\[CrossRef\]](#)
26. Bouayad, A.; Alami, H.; Janati Idrissi, M.; Berrada, I. Lightweight federated learning for efficient network intrusion detection. *IEEE Access* **2024**, *12*, 172027–172045. [\[CrossRef\]](#)
27. Rao, Y.N.; Babu, K.S. An imbalanced generative adversarial network-based approach for network intrusion detection in an imbalanced dataset. *Sensors* **2023**, *23*, 550. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Allagi, T.; Pawan, T.; Leong, W.Y. Enhanced Intrusion Detection Using Conditional Tabular-Generative-Adversarial-Network-Augmented Data and a Convolutional Neural Network: A Robust Approach to Addressing Imbalanced Cybersecurity Datasets. *Mathematics* **2025**, *13*, 1923. [\[CrossRef\]](#)
29. Hermosilla, P.; Berríos, S.; Allende-Cid, H. Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models. *Appl. Sci.* **2025**, *15*, 7329. [\[CrossRef\]](#)
30. Neupane, S.; Ables, J.; Anderson, W.; Mittal, S.; Rahimi, S.; Banicescu, I.; Seale, M. Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities. *IEEE Access* **2022**, *10*, 112392–112415. [\[CrossRef\]](#)

31. Arreche, O.; Guntur, T.R.; Roberts, J.W.; Abdallah, M. E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection. *IEEE Access* **2024**, *12*, 23954–23988. [\[CrossRef\]](#)
32. Alshinwan, M.; Batyha, R.M.; Alayed, W.; Alqahtany, S.S.; Abuowaida, S.; Mashagba, H.A.; Abd Aziz, A.B.; Al-Bawri, S.S. Enhancing Intrusion Detection Systems Using Hybrid AI-Based Approaches. *Comput. Mater. Contin.* **2026**, *87*, 2. [\[CrossRef\]](#)
33. Jakubcova, E.; Jakubec, M.; Pocta, P. A Hybrid Deep Learning Architecture for Intrusion Detection Deploying Multi-Scale Feature Interaction and Temporal Modeling. *AI* **2026**, *7*, 87. [\[CrossRef\]](#)
34. Prabhu, B.P.A.; Sunitha, N.R. A Hybrid Artificial Intelligence Model for Cyberattack Detection Using a Generative AI Embedded Approach. *Discov. Artif. Intell.* **2026**, *6*, 123. [\[CrossRef\]](#)
35. Maseno, E.M.; Wang, Z.; Xing, H. A Systematic Review on Hybrid Intrusion Detection System. *Secur. Commun. Netw.* **2022**, *2022*, 9663052. [\[CrossRef\]](#)
36. Rabiou, S.M.; Aminu, B.K.; Zubairu, D.A. AI-Driven Network Intrusion Detection Systems: A Systematic Review of Hybrid Models, Zero-Day Attack Mitigation, and Emerging Challenges in Cybersecurity. *Int. J. Comput. Appl.* **2025**, *187*, 27–33. [\[CrossRef\]](#)
37. Gheisari, M.; Pingmei, Z.; Riskhan, B.; Sibiya, M.; Muhammad, F.K.; Gheblezadeh, S.K. A Comprehensive Systematic Review of AI-Based Network Intrusion Detection Systems: Techniques, Datasets, Challenges, and Future Research Directions. *J. Comput. Data Technol.* **2026**, *2*, 140–150. [\[CrossRef\]](#)
38. Sowmya, T.; Mary Anita, E.A. A Comprehensive Review of AI Based Intrusion Detection System. *Meas. Sens.* **2023**, *28*, 100827. [\[CrossRef\]](#)
39. Al-Safaar, D.W.; Al-Yaseen, W.L. A Survey of Network Intrusion Detection Systems Based on Deep Learning Approaches. *Sci. Tech. J. Inf. Technol. Mech. Opt.* **2023**, *23*, 352–363. [\[CrossRef\]](#)
40. Alsulami, B. A Review on Machine-Learning-Based Approaches to Network Intrusion Detection Systems. *Int. J. Curr. Sci. Res. Rev.* **2022**, *5*, 2159–2177. [\[CrossRef\]](#)
41. Sharma, M. A Review of Machine Learning Techniques for Network Intrusion Detection Systems. *Int. J. Cyber Threat Intell. Secur. Netw.* **2026**, *3*, 14–22. [\[CrossRef\]](#)
42. Debicha, I.; Cochez, B.; Kenaza, T.; Debatty, T.; Dricot, J.-M.; Mees, W. Review on the Feasibility of Adversarial Evasion Attacks and Defenses for Network Intrusion Detection Systems. *arXiv* **2023**, arXiv:2303.07003. [\[CrossRef\]](#)
43. Chaudhary, D.; Rajasegarar, S.; Pokhrel, S.R. Towards Adapting Federated and Quantum Machine Learning for Network Intrusion Detection: A Survey. *arXiv* **2025**, arXiv:2509.21389. [\[CrossRef\]](#)
44. Mansoor, D.; Noori, A.W.; Qaderi, M.P. Continual Federated Learning for Network Intrusion Detection: An Open-Index Systematic Map. *J. Comput. Sci. Technol. Stud.* **2026**, *8*, 364–386. [\[CrossRef\]](#)
45. Capuano, N.; Fenza, G.; Loia, V.; Stanzione, C. Explainable Artificial Intelligence in CyberSecurity: A Survey. *IEEE Access* **2022**, *10*, 93575–93600. [\[CrossRef\]](#)
46. Leevy, J.L.; Khoshgoftaar, T.M. A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data. *J. Big Data* **2020**, *7*, 104. [\[CrossRef\]](#)
47. Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-Based Intrusion Detection Data Sets. *Comput. Secur.* **2019**, *86*, 147–167. [\[CrossRef\]](#)
48. Fida, M.-R.; Ahmed, A.H. Leveraging programmable data plane for network intrusion detection: A survey. *Comput. Netw.* **2026**, *281*, 112202. [\[CrossRef\]](#)
49. Lee, N.; Park, J.-H.; Cooper, E.J. Machine Learning for Network Intrusion Detection: An Analytical Review of Methods, Benchmark Datasets, and the Gap between Reported and Operational Performance. *Int. J. Inf. Digit. Secur.* **2026**, *4*, 23–31. [\[CrossRef\]](#)
50. Kitchenham, B.; Charters, S. *Guidelines for Performing Systematic Literature Reviews in Software Engineering*; Technical Report EBSE-2007-01; School of Computer Science and Mathematics, Keele University: Keele, UK; Department of Computer Science, University of Durham: Durham, UK, 2007. Available online: <https://ebse.webspace.durham.ac.uk/ebse-bibliography/guidelines-for-performing-systematic-literature-reviews-in-software-engineering/> (accessed on 14 October 2026).
51. 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.; John Wiley & Sons: Chichester, UK, 2019. Available online: <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119536604> (accessed on 14 October 2026).
52. Whiting, P.; Savović, J.; Higgins, J.P.T.; Caldwell, D.M.; Reeves, B.C.; Shea, B.; Davies, P.; Kleijnen, J.; Churchill, R. ROBIS: A New Tool to Assess Risk of Bias in Systematic Reviews. *J. Clin. Epidemiol.* **2016**, *69*, 225–234. [\[CrossRef\]](#) [\[PubMed\]](#)
53. Moamin, S.A.H.; Abdulhameed, M.K.; Al-Amri, R.M.; Radhi, A.D.; Naser, R.; Pheng, L.G. Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey of Techniques, Datasets, Challenges, and Future Directions. *Babylon. J. Artif. Intell.* **2025**, *2025*, 77–98. [\[CrossRef\]](#)
54. Dhakad, R.; Katore, S. Machine Learning Techniques for Network Intrusion Detection System (NIDS): A Survey. *Int. J. Emerg. Trends Eng. Res.* **2021**, *9*, 1477–1483. [\[CrossRef\]](#)
55. Ahmed, W.; Abdullah, Q.N. Network Intrusion Detection Systems: Machine-Learning-Based Attack and Remedy Strategies—A Review. *Al-Salam J. Eng. Technol.* **2025**, *4*, 11–29. [\[CrossRef\]](#)

56. Maseer, Z.K.; Kadhim, Q.K.; Al-Bander, B.; Yusof, R.; Saif, A. Meta-Analysis and Systematic Review for Anomaly Network Intrusion Detection Systems: Detection Methods, Dataset, Validation Methodology, and Challenges. *IET Netw.* **2024**, *13*, 339–376. [\[CrossRef\]](#)
57. Sarhan, M.; Layeghy, S.; Portmann, M. Towards a Standard Feature Set for Network Intrusion Detection System Datasets. *Mob. Netw. Appl.* **2022**, *27*, 357–370. [\[CrossRef\]](#)
58. Anderson, B.; Paul, S.; McGrew, D. Deciphering Malware’s Use of TLS (Without Decryption). *J. Comput. Virol. Hacking Tech.* **2018**, *14*, 195–211. [\[CrossRef\]](#)
59. Khenwar, M.; Nawal, M. Challenges and limitations of IDS: A comprehensive assessment and future perspectives. *Ski. Res. J.* **2024**, *14*, 35–39. [\[CrossRef\]](#)
60. Chou, D.; Jiang, M. A Survey on Data-Driven Network Intrusion Detection. *ACM Comput. Surv.* **2021**, *54*, 182. [\[CrossRef\]](#) [\[PubMed\]](#)
61. Bizzarri, A.; Yu, C.-E.; Jalaian, B.; Riguzzi, F.; Bastian, N.D. Neurosymbolic AI for network intrusion detection systems: A survey. *J. Inf. Secur. Appl.* **2025**, *94*, 104205. [\[CrossRef\]](#)
62. Liu, H.; Wang, X.; He, F.; Zheng, Z. Automated network defense: A systematic survey and analysis of AutoML paradigms for network intrusion detection. *Appl. Sci.* **2025**, *15*, 10389. [\[CrossRef\]](#)
63. 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*, 71. [\[CrossRef\]](#) [\[PubMed\]](#)
64. Rethlefsen, M.L.; Kirtley, S.; Waffenschmidt, S.; Ayala, A.P.; Moher, D.; Page, M.J.; Koffel, J.B.; PRISMA-S Group. PRISMA-S: An extension to the PRISMA Statement for Reporting Literature Searches in Systematic Reviews. *Syst. Rev.* **2021**, *10*, 39. [\[CrossRef\]](#) [\[PubMed\]](#)
65. National Institute of Standards and Technology (NIST). *Guide for Conducting Risk Assessments*; NIST Special Publication 800-30, Revision 1; U.S. Department of Commerce: Gaithersburg, MD, USA, 2012. [\[CrossRef\]](#)
66. ISO/IEC 27005:2022; Information Security, Cybersecurity and Privacy Protection—Guidance on Managing Information Security Risks. ISO: Geneva, Switzerland, 2022. Available online: <https://www.iso.org/standard/80585.html> (accessed on 14 October 2026).
67. Braun, V.; Clarke, V. One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qual. Res. Psychol.* **2021**, *18*, 328–352. [\[CrossRef\]](#)
68. Sherin, R.C.J.; Parkavi, K. Investigations on Bio-Inspired Algorithms for Network Intrusion Detection—A Review. *Int. J. Comput. Netw. Appl.* **2022**, *9*, 399–423. [\[CrossRef\]](#)
69. Venkatesh, N.; Kashikar, P. A Literary Review of Pattern-Matching Techniques in Network Intrusion Detection. *Int. J. Curr. Sci. Res. Rev.* **2023**, *6*, 6491–6496. [\[CrossRef\]](#)
70. Pachauri, P.; Mahour, S.; Singh, A.; Sharma, R. Comprehensive Review: Network Intrusion Detection System Using Machine Learning on the NSL-KDD Dataset. *Int. J. Res. Trends Innov.* **2026**, *11*, a331–a339. [\[CrossRef\]](#)
71. Aleng, E.A.; Sarjiyus, O.; Yusufu, G. Deep-Learning-Based Network Intrusion Detection for the Internet of Things Using Bidirectional Long Short-Term Memory: A Review. *Iconic Res. Eng. J.* **2026**, *10*, 3707–3721. [\[CrossRef\]](#)
72. Khadtare, A.; Kakani, P.; Jambhulkar, T.; Jagtap, P.; Dhengre, S. The Proposed Intrusion Detection System Using Support Vector Machines in IoT-Enabled Wireless Sensor Networks: A Survey. *Int. J. Res. Appl. Sci. Eng. Technol.* **2023**, *11*, 850–858. [\[CrossRef\]](#)
73. Campbell, M.; McKenzie, J.E.; Sowden, A.; Katikireddi, S.V.; Brennan, S.E.; Ellis, S.; Hartmann-Boyce, J.; Ryan, R.; Shepperd, S.; Thomas, J.; et al. Synthesis without meta-analysis (SWiM) in systematic reviews: Reporting guideline. *BMJ* **2020**, *368*, l6890. [\[CrossRef\]](#) [\[PubMed\]](#)
74. Nguyen, V.T.; Beuran, R. FedMSE: Semi-supervised federated learning approach for IoT network intrusion detection. *Comput. Secur.* **2025**, *151*, 104337. [\[CrossRef\]](#)
75. Xi, C.; Wang, H.; Wang, X. A novel multi-scale network intrusion detection model with transformer. *Sci. Rep.* **2024**, *14*, 23239. [\[CrossRef\]](#) [\[PubMed\]](#)
76. Tavallae, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6. [\[CrossRef\]](#)
77. Neto, E.C.P.; Iqbal, S.; Buffett, S.; Sultana, M.; Taylor, A. Deep Learning for Intrusion Detection in Emerging Technologies: A Comprehensive Survey and New Perspectives. *Artif. Intell. Rev.* **2025**, *58*, 340. [\[CrossRef\]](#)
78. Chen, Z.; Simsek, M.; Kantarci, B.; Bagheri, M.; Djukic, P. Machine learning-enabled hybrid intrusion detection system with host data transformation and an advanced two-stage classification. *Comput. Netw.* **2024**, *250*, 110576. [\[CrossRef\]](#)
79. Yin, Y.; Jang-Jaccard, J.; Xu, W.; Singh, A.; Zhu, J.; Sabrina, F.; Kwak, J. IGRF-RFE: A Hybrid Feature Selection Method for MLP-Based Network Intrusion Detection on UNSW-NB15 Dataset. *J. Big Data* **2023**, *10*, 15. [\[CrossRef\]](#)

80. Landauer, M.; Skopik, F.; Frank, M.; Hotwagner, W.; Wurzenberger, M.; Rauber, A. Maintainable Log Datasets for Evaluation of Intrusion Detection Systems. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 3466–3482. [\[CrossRef\]](#)
81. Bridges, R.A.; Glass-Vanderlan, T.R.; Iannacone, M.D.; Vincent, M.S.; Chen, Q. A Survey of Intrusion Detection Systems Leveraging Host Data. *ACM Comput. Surv.* **2019**, *52*, 1–35. [\[CrossRef\]](#)
82. Xu, W. Advancements in machine learning for network anomaly detection: A comprehensive investigation. In Proceedings of the 1st International Conference on Engineering Management, Information Technology and Intelligence, Shanghai, China, 14 June 2024; pp. 585–589. [\[CrossRef\]](#)
83. Rathee, A.; Malik, P.; Parida, M.K. Network intrusion detection system using deep learning techniques. In Proceedings of the 2023 International Conference on Communication, Circuits, and Systems, IC3S, Bhubaneswar, India, 26–28 May 2023; pp. 1–6. [\[CrossRef\]](#)
84. Ugtakhybayar, N.; Usukhbayar, B.; Baigaltugs, S. A hybrid model for anomaly-based intrusion detection system. In *Advances in Intelligent Information Hiding and Multimedia Signal Processing*; Smart Innovation, Systems and Technologies; Springer: Singapore, 2020; Volume 173, pp. 419–431. [\[CrossRef\]](#)
85. Hu, Q.; Yu, S.-Y.; Asghar, M.R. Analysing performance issues of open-source intrusion detection systems in high-speed networks. *J. Inf. Secur. Appl.* **2020**, *51*, 102426. [\[CrossRef\]](#)
86. Ghazi, D.S.; Saleh, A.M.; Salih, O.M. Snort versus Suricata in intrusion detection. *Iraqi J. Inf. Commun. Technol.* **2024**, *7*, 73–88. [\[CrossRef\]](#)
87. Saheed, Y.K.; Abiodun, A.I.; Misra, S.; Holone, M.K.A.; Colomo-Palacios, R. A machine learning-based intrusion detection for detecting internet of things network attacks. *Alex. Eng. J.* **2022**, *61*, 9395–9409. [\[CrossRef\]](#)
88. Chinnasamy, P.; Yarramsetti, S.; Ayyasamy, R.K.; Rajesh, E.; Vijayasaro, V.; Pandey, D.; Pandey, B.K.; Lelish, M.E. AI-driven intrusion detection and prevention systems to safeguard 6G networks from cyber threats. *Sci. Rep.* **2025**, *15*, 37901. [\[CrossRef\]](#) [\[PubMed\]](#)
89. Scarfone, K.; Mell, P. *Guide to Intrusion Detection and Prevention Systems (IDPS)*; NIST Special Publication 800-94; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2007. [\[CrossRef\]](#)
90. Roshan, S.; Zali, Y.; Rahmani, A.M.; Sakly, A. Adaptive and online network intrusion detection system using clustering and extreme learning machines. *J. Frankl. Inst.* **2020**, *357*, 1752–1779. [\[CrossRef\]](#)
91. Alqahtani, A.H. An Incremental Hybrid Adaptive Network-Based IDS in Software Defined Networks to Detect Stealth Attacks. *arXiv* **2024**, arXiv:2404.01109. [\[CrossRef\]](#)
92. Rai, H.M.; Pal, A.; Ergashoglu, R.A.; Ugli, B.A.K.; Shokirovich, Y.S. Advanced AI-powered intrusion detection systems in cybersecurity protocols for network protection. *Procedia Comput. Sci.* **2025**, *259*, 140–149. [\[CrossRef\]](#)
93. Dias, T.; Oliveira, N.; Sousa, N.; Praça, I.; Sousa, O. A hybrid approach for an interpretable and explainable intrusion detection system. In *Intelligent Systems Design and Applications*; Springer: Cham, Switzerland, 2022; Volume 295, pp. 1035–1045. [\[CrossRef\]](#)
94. Nguyen, T.T.; Reddi, V.J. Deep Reinforcement Learning for Cyber Security. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 3779–3795. [\[CrossRef\]](#) [\[PubMed\]](#)
95. Caminero, G.; Lopez-Martin, M.; Carro, B. Adversarial environment reinforcement learning algorithm for intrusion detection. *Comput. Netw.* **2019**, *159*, 96–109. [\[CrossRef\]](#)
96. Ren, K.; Zeng, Y.; Zhong, Y.; Sheng, B.; Zhang, Y. MAFSIDS: A Reinforcement Learning-Based Intrusion Detection Model for Multi-Agent Feature Selection Networks. *J. Big Data* **2023**, *10*, 137. [\[CrossRef\]](#)
97. Dulac-Arnold, G.; Levine, N.; Mankowitz, D.J.; Li, J.; Paduraru, C.; Gowal, S.; Hester, T. Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Mach. Learn.* **2021**, *110*, 2419–2468. [\[CrossRef\]](#)
98. García, J.; Fernández, F. A Comprehensive Survey on Safe Reinforcement Learning. *J. Mach. Learn. Res.* **2015**, *16*, 1437–1480. Available online: <https://jmlr.org/papers/v16/garcia15a.html> (accessed on 14 October 2026).
99. Khetarpal, K.; Riemer, M.; Rish, I.; Precup, D. Towards Continual Reinforcement Learning: A Review and Perspectives. *J. Artif. Intell. Res.* **2022**, *75*, 1401–1476. [\[CrossRef\]](#)
100. Saunders, W.; Sastry, G.; Stuhlmüller, A.; Evans, O. Trial Without Error: Towards Safe Reinforcement Learning via Human Intervention. In Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems (AAMAS), São Paulo, Brazil, 8–12 May 2017; pp. 2067–2069. Available online: <https://dl.acm.org/doi/10.5555/3237383.3238074> (accessed on 14 October 2026).
101. Khan, S. Human in the loop reinforcement learning framework for adaptive phishing detection. *Discov. Comput.* **2025**, *28*, 311. [\[CrossRef\]](#)
102. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282. Available online: <https://proceedings.mlr.press/v54/mcmahan17a.html> (accessed on 14 October 2026).

103. Mothukuri, V.; Parizi, R.M.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [\[CrossRef\]](#)
104. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [\[CrossRef\]](#)
105. Zhu, L.; Liu, Z.; Han, S. Deep Leakage from Gradients. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; pp. 14774–14784.
106. Geiping, J.; Bauermeister, H.; Dröge, H.; Moeller, M. Inverting Gradients—How Easy Is It to Break Privacy in Federated Learning? In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020; pp. 16937–16947.
107. Shokri, R.; Stronati, M.; Song, C.; Shmatikov, V. Membership Inference Attacks Against Machine Learning Models. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 3–18. [\[CrossRef\]](#)
108. Melis, L.; Song, C.; De Cristofaro, E.; Shmatikov, V. Exploiting Unintended Feature Leakage in Collaborative Learning. In Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 19–23 May 2019; pp. 691–706. [\[CrossRef\]](#)
109. Bhagoji, A.N.; Chakraborty, S.; Mittal, P.; Calo, S. Analyzing Federated Learning Through an Adversarial Lens. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; Volume PMLR 97, pp. 634–643. Available online: <https://proceedings.mlr.press/v97/bhagoji19a.html> (accessed on 14 October 2026).
110. Tolpegin, V.; Truex, S.; Gursoy, M.E.; Liu, L. Data Poisoning Attacks Against Federated Learning Systems. In *Computer Security—ESORICS 2020*; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12308, pp. 480–501. [\[CrossRef\]](#)
111. Huang, J.; Chen, Z.; Liu, S.-Z.; Zhang, H.; Long, H.-X. Improved intrusion detection based on hybrid deep learning models and federated learning. *Sensors* **2024**, *24*, 4002. [\[CrossRef\]](#) [\[PubMed\]](#)
112. Dwork, C. Differential Privacy. In Proceedings of the 33rd International Colloquium on Automata, Languages, and Programming (ICALP), Venice, Italy, 10–14 July 2006; pp. 1–12. [\[CrossRef\]](#)
113. Wei, K.; Li, J.; Ding, M.; Ma, C.; Yang, H.H.; Farokhi, F.; Jin, S.; Quek, T.Q.S.; Poor, H.V. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3454–3469. [\[CrossRef\]](#)
114. Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H.B.; Patel, S.; Ramage, D.; Segal, A.; Seth, K. Practical Secure Aggregation for Privacy-Preserving Federated Learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS), Dallas, TX, USA, 30 October–3 November October 2017; pp. 1175–1191. [\[CrossRef\]](#)
115. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* **2018**, *51*, 1–35. [\[CrossRef\]](#)
116. Alabbadi, A.; Bajaber, F. An intrusion detection system over IoT data streams using eXplainable Artificial Intelligence (XAI). *Sensors* **2025**, *25*, 847. [\[CrossRef\]](#) [\[PubMed\]](#)
117. Jain, S.; Wallace, B.C. Attention is not Explanation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Minneapolis, MN, USA, 2–7 June 2019; pp. 3543–3556. [\[CrossRef\]](#)
118. Wiegrefe, S.; Pinter, Y. Attention is Not NOT Explanation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), Hong Kong, China, 3–7 November 2019; pp. 11–20. [\[CrossRef\]](#)
119. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 6–11 August 2017; Volume PMLR 70, pp. 3319–3328. Available online: <https://proceedings.mlr.press/v70/sundararajan17a.html> (accessed on 14 October 2026).
120. Apruzzese, G.; Andreolini, M.; Ferretti, L.; Marchetti, M.; Colajanni, M. Modeling Realistic Adversarial Attacks against Network Intrusion Detection Systems. *Digit. Threats Res. Pract.* **2022**, *3*, 1–19. [\[CrossRef\]](#)
121. 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.
122. Warnecke, A.; Arp, D.; Wressnegger, C.; Rieck, K. Evaluating Explanation Methods for Deep Learning in Security. In Proceedings of the 2020 IEEE European Symposium on Security and Privacy (EuroS&P), Genoa, Italy, 7–11 September 2020; pp. 158–174. [\[CrossRef\]](#)
123. Nadeem, A.; Vos, D.; Cao, C.; Pajola, L.; Dieck, S.; Baumgartner, R.; Verwer, S. SoK: Explainable Machine Learning for Computer Security Applications. In Proceedings of the 2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P), Delft, The Netherlands, 3–7 July 2023; pp. 221–240. [\[CrossRef\]](#)
124. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated Optimization in Heterogeneous Networks. In Proceedings of the Machine Learning and Systems (MLSys 2020), Austin, TX, USA, 2–4 March 2020; Volume 2, pp. 429–450.
125. Chauhan, R.; Sabeel, U.; Izaddoost, A.; Heydari, S.S. Polymorphic Adversarial Cyberattacks Using WGAN. *J. Cybersecur. Priv.* **2021**, *1*, 767–792. [\[CrossRef\]](#)

126. RFC 8446; The Transport Layer Security (TLS) Protocol Version 1.3. Internet Engineering Task Force: Fremont, CA, USA, 2018. [[CrossRef](#)]
127. RFC 9000; QUIC: A UDP-Based Multiplexed and Secure Transport. Internet Engineering Task Force: Fremont, CA, USA, 2021. [[CrossRef](#)]
128. RFC 8484; DNS Queries Over HTTPS (DoH). Internet Engineering Task Force: Fremont, CA, USA, 2018. [[CrossRef](#)]
129. Hacker, P.; Passoth, J.H. Varieties of AI Explanations Under the Law. From the GDPR to the AIA, and Beyond. In *XxAI—Beyond Explainable AI. XxAI 2020; Lecture Notes in Computer Science*; Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, K.R., Samek, W., Eds.; Springer: Cham, Switzerland, 2022; Volume 13200. [[CrossRef](#)]
130. Olanrewaju, B.A.; Onianwa, B. Regulating AI in Cybersecurity: Balancing Challenges, Opportunities, and Global Frameworks. *Asian J. Res. Comput. Sci.* **2025**, *18*, 1–17. [[CrossRef](#)]
131. European Union. Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on Measures for a High Common Level of Cybersecurity Across the Union (NIS2 Directive); European Union: Luxembourg, 2022; Volume L 333, pp. 80–152. Available online: <http://data.europa.eu/eli/dir/2022/2555/oj> (accessed on 14 October 2026).
132. European Union. Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on Digital Operational Resilience for the Financial Sector (DORA); European Union: Luxembourg, 2022; Volume L 333, pp. 1–79. Available online: <http://data.europa.eu/eli/reg/2022/2554/oj> (accessed on 14 October 2026).
133. Pandey, A.; Agrawal, A.K.; Shukla, A.; Keswani, G.; Bokoro, P.N.; Dubey, P. From Signature to Attention: Transformer-Powered Intrusion Detection Systems for Cybersecurity. *Future Internet* **2026**, *18*, 398. [[CrossRef](#)]
134. Gambo, M.D.; Khan, A.H.; Almulhem, A.; Almadani, B. An Efficient Framework for Automated Cyber Threat Intelligence Sharing. *Electronics* **2025**, *14*, 4045. [[CrossRef](#)]
135. Nouman, M.; Ujjan, R.; Hassanu, M. TLS-Aware Anomaly Detection for Encrypted IoT Traffic Using a β -Variational Autoencoder with ANOVA–Mutual Information Feature Selection. *Future Internet* **2026**, *18*, 310. [[CrossRef](#)]
136. Heer, M.J.; Ramhorst, B.; Alonso, G. Machine Learning-based Deep Packet Inspection at Line Rate for RDMA on FPGAs. In *EuroMLSys '25: Proceedings of the 5th Workshop on Machine Learning and Systems*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 148–155. [[CrossRef](#)]

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.