

Paper Type: Review Article

A Comprehensive Survey on Machine Learning and Deep Learning Approaches for Network Intrusion Detection Systems: Techniques, Datasets, and Open Challenges

Yahya Ali Alnattaf ^{1,*}  and Hanaa Fathi Mahmoud ¹ 

¹ Department of Computer Science, College of Education for Pure Science, University of Mosul, Nineveh, Iraq;
Emails: Ya6519260@gmail.com; Dr.hanah@uomosul.edu.iq.

Received: 07 Feb 2026

Revised: 12 May 2026

Accepted: 29 Aug 2026

Published: 23 Sep 2026

Abstract

The rapid expansion of networked infrastructure, cloud computing, and IoT environments has significantly increased the attack surface, while traditional signature-based Network Intrusion Detection Systems (NIDS) remain limited in detecting zero-day attacks, polymorphic threats, and malicious activities within encrypted traffic. This survey reviews and synthesizes a decade of research from 2015 to 2025 on Machine Learning (ML) and Deep Learning (DL) approaches for NIDS, providing a unified taxonomy that organizes existing methods according to both their learning paradigms, including supervised, unsupervised, semi-supervised, reinforcement, and self-supervised learning, and their architectural designs, such as classical ML, convolutional, recurrent, autoencoder, generative, attention-based, transformer, graph, and federated models. The review shows that tree-based ensemble methods, particularly Random Forest and XGBoost, continue to provide strong and competitive baselines for tabular network-flow data, highlighting the need for more complex DL models to demonstrate clear practical advantages. Hybrid spatio-temporal architectures, such as CNN-LSTM models, and transformer-based approaches have achieved strong performance on recent datasets; however, results on saturated benchmarks such as NSL-KDD provide limited evidence of meaningful progress. More importantly, cross-dataset evaluations reveal substantial generalization gaps, with F1-score differences of approximately 20–30 percentage points, indicating persistent problems related to dataset bias and label quality. Overall, NIDS research is gradually moving beyond the pursuit of higher accuracy toward practical challenges such as real-time edge inference, adversarial robustness, explainability, privacy-preserving and federated learning, and intrusion detection over encrypted traffic. Based on these findings, this survey highlights standardized feature representations, rigorous temporal and cross-dataset evaluation, and self-supervised pre-training as promising directions for developing more robust and generalizable NIDS models.

Keywords: Intrusion Detection, NIDS, Machine Learning, Deep Learning, Anomaly Detection, Cybersecurity.

1 | Introduction

Modern digital infrastructure—spanning enterprise networks, cloud data centres, 5G/6G mobile systems, and billions of IoT devices—is the target of an ever-growing volume and sophistication of cyberattacks [1, 2]. Reports from major security vendors consistently document increases in ransomware, distributed denial-



Corresponding Author: : Ya6519260@gmail.com



<https://doi.org/10.61356/j.nois.2026.11667>



Licensee **Neutrosophic Optimization and Intelligent systems**. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0>).

of-service (DDoS), advanced persistent threats (APTs), and supply-chain compromises, with average breach costs in 2024 exceeding USD 4.5 million per incident [3]. Within this landscape, Intrusion Detection Systems (IDS) are a fundamental layer of defence, complementing firewalls, endpoint protection, and identity controls.

A NIDS continuously inspects network traffic—either as raw packets or as aggregated flow records—to identify malicious activity such as scans, exploits, lateral movement, command-and-control communication, or data exfiltration [4]. Traditional NIDS rely primarily on signatures: hand-crafted rules (e.g., Snort, Suricata) that match known attack patterns. Signature-based detection is precise and fast for known threats, but is fundamentally unable to detect novel (zero-day) attacks and is brittle in the face of obfuscation, payload encryption, and protocol mutation [2, 5].

Anomaly-based detection seeks to overcome these limitations by modelling normal behaviour and flagging statistically significant deviations. Early statistical and rule-mining methods have largely been superseded by data-driven approaches based on Machine Learning (ML) and, since around 2015, Deep Learning (DL) [6]. Modern ML/DL NIDS exploit the availability of large labelled traffic datasets, GPU acceleration, and high-capacity neural architectures to automatically learn discriminative representations of benign and malicious flows [7, 8].

Despite remarkable reported accuracy on benchmark datasets—often above 99%—deploying ML/DL NIDS in production remains challenging. Class imbalance, dataset shift between training and operational environments, adversarial evasion, the need for sub-millisecond inference at line rate, and limited interpretability all complicate practical adoption [9,10]. Furthermore, the rapid emergence of new model families (transformers [11], graph neural networks (GNNs) [12], federated learning [13]) has produced a fragmented literature that is difficult for practitioners to navigate.

Contributions. This survey aims to provide a clear, up-to-date entry point to the field. Our contributions are:

- A unified taxonomy of ML/DL approaches for NIDS, covering classical ML, deep feed-forward, convolutional, recurrent, autoencoder-based, generative, attention/transformer, graph, and federated models (Section 2).
- A structured review of ML approaches (Section 3) and DL approaches (Section 4), including comparison tables of strengths, weaknesses, and typical application scenarios.
- A consolidated overview of benchmark datasets (Section 5) with feature counts, attack categories, year of release, and known biases.
- A discussion of evaluation metrics (Section 6) suited to NIDS, including the limitations of accuracy in highly imbalanced settings.
- A comparative analysis (Section 7) of more than 25 representative recent works, with a single-page summary table.
- A critical review of open challenges and future directions (Section 8), including adversarial robustness, concept drift, explainability, and federated/edge NIDS.

Several prior surveys have addressed ML/DL-based NIDS, including Ahmad et al. [1], Khraisat et al. [2], and Ferrag et al. [57]. However, these works either predate the widespread adoption of transformer-based and graph-based architectures, focus narrowly on a single learning paradigm (e.g., supervised DL only), or do not jointly analyse dataset bias, evaluation-metric pitfalls, and open operational challenges within a single unified framework. Compared to these efforts, the present survey (i) extends coverage to 2025, incorporating self-supervised, federated, and LLM-assisted detection approaches that have emerged only in the last two years; (ii) introduces a taxonomy that explicitly separates learning paradigm from architectural family, allowing readers to reason about both axes independently; (iii) provides a single comparative table spanning more than 25 studies drawn from both classical ML and DL literature, rather than treating them separately as in [1, 2]; and (iv) places particular emphasis on reproducibility pitfalls (data leakage, inconsistent splits) that are frequently overlooked in earlier surveys. This combination of temporal coverage, unified taxonomy, and critical methodological focus distinguishes the present work from existing reviews.

2 | Background and Taxonomy

2.1 | Intrusion Detection Systems

All experiments were implemented in Python [version, e.g., 3.10] using [libraries used, e.g., NumPy for matrix operations and SciPy/custom scripts for entropy and statistical calculations]. Attack

Following the classical taxonomy of Liao et al. [4] and the more recent treatments by Khraisat et al. [2] and Ahmad et al. [1], IDSs can be classified along three orthogonal axes:

- **Deployment scope:** Host-based IDS (HIDS) monitor system calls, log files, and process behaviour on individual hosts; Network-based IDS (NIDS) inspect packets or flows on network links; Hybrid IDS combine both views.
- **Detection methodology:** Signature-based (misuse) detectors match known attack patterns; anomaly-based detectors flag deviations from a learned model of normality; specification-based detectors compare behaviour to a manually defined model of correct operation.
- **Decision granularity:** Per-packet, per-flow, per-session, or per-host.

This survey focuses on anomaly- and hybrid-based NIDS that leverage ML or DL, typically operating on flow-level features extracted by tools such as CICFlowMeter or Argus [19].

2.2 | Machine Learning Paradigms for NIDS

- **Supervised learning** requires labelled benign/malicious samples and learns a classifier $f: X \rightarrow Y$. It dominates the literature because it produces highly accurate models when labels are available.
- **Unsupervised learning** models $p(x)$ without labels; deviations from the learned distribution are flagged as anomalies. It is attractive for zero-day detection but typically yields higher false-alarm rates.
- **Semi-supervised learning** uses a small labelled set together with abundant unlabelled traffic, which is the realistic operational regime [14].
- **Reinforcement learning (RL)** treats detection or response as a sequential decision problem, with the agent learning policies that maximise security utility [15].
- **Self-supervised learning (SSL)** an emerging paradigm, exploits pretext tasks on unlabelled traffic to learn transferable representations [26].

2.3 | Proposed Taxonomy

At the top level, ML/DL NIDS are split into classical ML and deep learning branches. Classical ML covers tree-based, kernel-based, probabilistic, instance-based, and unsupervised methods. The DL branch covers feed-forward (DNN), convolutional (CNN), recurrent (RNN/LSTM/GRU), autoencoder (AE/VAE), generative (GAN), attention-based (Transformer), graph (GNN), and cross-cutting paradigms such as hybrid, federated, and explainable models.

3 | Machine Learning Approaches

Classical ML methods remain highly competitive for NIDS, particularly when training data is limited, interpretability is required, or compute is constrained at the edge [1, 5].

3.1 | Decision Trees and Random Forests

Decision Trees (DT) recursively partition the feature space using information-theoretic splits. They are interpretable and fast at inference. Random Forests (RF) aggregate many decorrelated trees via bagging and feature subsampling, achieving strong accuracy with little tuning [16]. Several studies report RF as the best classical baseline on NSL-KDD, UNSW-NB15, and CIC-IDS2017 [17-20].

3.2 | Gradient Boosting and XGBoost

Gradient-boosted decision trees (GBDT) and their efficient implementations such as XGBoost [21], LightGBM, and CatBoost dominate tabular benchmarks and frequently outperform deep models on flow-level NIDS features [22, 23]. They handle mixed categorical/numerical features well and provide feature-importance scores useful for security analysts.

3.3 | Support Vector Machines

SVMs find a maximum-margin hyperplane in a (kernel-induced) feature space. With RBF kernels they handle non-linear boundaries effectively. SVMs were among the earliest ML methods applied to KDD Cup 99 and NSL-KDD [24]. Their main drawback is $O(n^2)$ – $O(n^3)$ training cost, which limits scalability to modern multi-million-flow datasets.

3.4 | k-Nearest Neighbours

k-NN is a non-parametric instance-based classifier that labels a query by majority vote of its k nearest training points. It requires no training but $O(n)$ inference cost per query, making it impractical for line-rate detection without acceleration structures or prototype selection [25].

3.5 | Naive Bayes and Bayesian Networks

Naive Bayes (NB) assumes feature independence given the class. Despite its simplicity, NB is fast and provides calibrated probabilities. Bayesian Networks relax the independence assumption and have been used to model multi-stage attacks [26].

3.6 | Unsupervised Methods

- **k-Means:** clusters traffic into k groups; clusters dominated by attacks at training time can be flagged at inference.
- **DBSCAN:** finds density-connected clusters and naturally labels points in low-density regions as anomalies.
- **Isolation Forest (iForest):** isolates anomalies via random partitioning; samples that are isolated with few splits are deemed anomalous [27]. It scales linearly in n and is widely used for unsupervised NIDS baselines.
- **One-Class SVM:** learns the support of the benign distribution and flags points outside it [28].

3.7 | Comparative Discussion

Table 1 summarises classical ML methods for NIDS. Tree ensembles (RF, XGBoost) generally provide the best accuracy/cost trade-off on tabular flow features and are strong baselines that any DL method should beat to be considered worthwhile [9].

Table 1. Summary of classical ML approaches for NIDS.

Method	Strengths	Weaknesses	Typical use
Decision Tree	Interpretable, fast inference	Overfits, unstable	Lightweight baselines, edge
Random Forest	Strong accuracy, robust, parallelisable	Larger memory footprint	Tabular flow classification
XGBoost / LightGBM	State-of-the-art on tabular data, importance scores	Hyperparameter sensitive	Production NIDS, competitions
SVM (RBF)	Strong for small high-dim data	Quadratic training cost	Small/medium labelled sets
k-NN	Simple, no training	Slow inference, sensitive to scaling	Prototyping, low-volume HIDS

Naive Bayes	Very fast, calibrated probabilities	Independence assumption	Streaming, baselines
k-Means	Simple unsupervised baseline	Spherical clusters, k tuning	Coarse traffic clustering
DBSCAN	Finds arbitrary-shape clusters	Sensitive to ϵ , scaling	Density-based anomaly detection
Isolation Forest	Linear scaling, few hyperparams	Weak on dense anomalies	Unsupervised NIDS baseline
One-Class SVM	Principled novelty detection	Hard to tune, slow	Zero-day / novelty detection

Discussion. Table 1 shows a clear separation between methods favouring interpretability and speed (Decision Tree, Naive Bayes) and methods favouring raw accuracy (Random Forest, XGBoost). Tree ensembles consistently top classical-ML leaderboards on NSL-KDD and UNSW-NB15 because they natively handle mixed categorical/numerical flow features without extensive preprocessing. Distance-based methods (k-NN, SVM) scale poorly to the multi-million-flow volumes typical of CIC-IDS2017/2018, which explains their declining use in recent production-oriented studies. Among unsupervised methods, Isolation Forest offers the best scalability/accuracy trade-off for zero-day detection, whereas One-Class SVM remains attractive only when strict novelty guarantees are required and dataset size is modest.

4 | Deep Learning Approaches

DL approaches automatically learn hierarchical representations of network traffic, removing much of the burden of manual feature engineering [6, 7]. We group them by architectural family.

4.1 | Deep Neural Networks (DNN/MLP)

Multi-layer perceptrons with several hidden layers were among the first DL methods applied to NIDS. Vinayakumar et al. [7] performed an extensive evaluation of DNNs across KDD Cup 99, NSL-KDD, UNSW-NB15, WSN-DS, CICIDS2017, and Kyoto, reporting strong and consistent results. DNNs are simple to deploy but, on tabular flow features, often do not outperform tuned tree ensembles [9].

4.2 | Convolutional Neural Networks (CNN)

CNNs were originally designed for images, but have been successfully applied to NIDS in two main ways. (i) Tabular-to-image encodings reshape flow feature vectors into 2-D matrices on which standard CNNs operate [29]. (ii) Raw-payload CNNs treat the first N bytes of a packet (or a session) as a 1-D sequence, learning protocol-aware features end-to-end [30,31]. CNNs excel at capturing local correlations among adjacent bytes or features.

4.3 | Recurrent Neural Networks (RNN/LSTM/GRU)

Network traffic is intrinsically temporal: connections evolve, attacks unfold over multiple packets, and user behaviour exhibits strong temporal correlations. RNNs, and especially their gated variants Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), model these dependencies effectively [32-34]. Bi-directional and stacked LSTMs are commonly used for sequence-level intrusion classification on packet or flow sequences.

4.4 | Autoencoders and Variational Autoencoders

Autoencoders (AE) compress benign traffic into a low-dimensional bottleneck and reconstruct it; high reconstruction error at test time signals anomaly [8]. Sparse, denoising, and stacked variants have all been used in NIDS. Variational Autoencoders (VAE) impose a probabilistic latent prior, enabling likelihood-based scoring [35]. AE/VAE are central to semi-supervised NIDS, where only benign data is available at training time.

4.5 | Generative Adversarial Networks

GANs train a generator and a discriminator in a minimax game [36]. In NIDS, GANs are used to (i) augment rare attack classes, mitigating class imbalance [36]; (ii) perform anomaly detection via discriminator scores or reconstruction (AnoGAN-style); and (iii) generate adversarial examples to evaluate and improve robustness [37, 38].

4.6 | Attention and Transformer Models

Transformer architectures, originally proposed for sequence modelling in NLP [11], have rapidly been adapted to NIDS. Models such as RTIDS [26], FlowTransformer [39], and ET-BERT [40] treat packets, flows, or byte sequences as tokens and apply self-attention to capture long-range dependencies. Self-supervised pre-training on unlabelled traffic followed by fine-tuning on small labelled sets is a particularly promising direction. Recent work has extended this direction toward large language models (LLMs) as feature encoders or classifiers for network traffic [76,78,79], as well as cloud-specific transformer deployments [77].

4.7 | Graph Neural Networks

Networks are naturally graphs: hosts are nodes and flows are edges. GNNs exploit this structure, propagating information along communication patterns [12,41]. E-GraphSAGE [41] learns flow-level embeddings that incorporate topological context and outperforms purely flow-feature-based models on several datasets.

4.8 | Hybrid Architectures

Many state-of-the-art NIDS combine multiple architectures, e.g., CNN+LSTM [7, 42], AE+CNN, or transformer+GNN. Hybrid pipelines first extract spatial or local features and then model temporal or relational structure.

4.9 | Federated and Edge DL

Sharing raw traffic across organisations is rarely feasible due to privacy and regulatory constraints. Federated Learning (FL) [13] allows multiple parties to collaboratively train a NIDS while keeping data local. Several recent works apply FL to IoT NIDS [43-45], addressing both privacy and non-IID traffic distributions.

Table 2. Summary of deep learning architectures for NIDS.

Architecture	Strengths	Weaknesses	Typical use
DNN / MLP	Simple, end-to-end	Often matched by RF/XGB on tabular data	Flow classification baseline
CNN	Local feature extraction, GPU efficient	Needs reshaping of tabular data	Payload, image-encoded flows
LSTM / GRU	Captures temporal dependencies	Slower training, vanishing gradients	Packet/flow sequences
AE / VAE	Semi-supervised, zero-day capable	Threshold tuning, latent collapse	Anomaly detection
GAN	Data augmentation, novelty detection	Mode collapse, unstable training	Imbalance, adversarial training
Transformer	Long-range context, transferable pre-training	High compute and memory	Self-supervised NIDS
GNN	Topological context, multi-hop reasoning	Graph construction overhead	Lateral movement, botnets
Hybrid (CNN+LSTM)	Combines spatial + temporal cues	Higher complexity	General-purpose NIDS
Federated DL	Privacy preserving, multi-tenant	Communication cost, non-IID	Cross-org / IoT NIDS

Discussion. Table 2 highlights that no single DL architecture dominates across all deployment scenarios. CNNs and LSTMs remain the workhorses for spatial and temporal feature extraction respectively, while autoencoder-based models are the default choice whenever only benign traffic is available for training. Transformer and GNN architectures achieve the strongest reported results on modern IoT and cloud datasets but incur substantially higher memory and compute costs, which limits their applicability at the network edge unless combined with distillation or quantisation. Federated DL is the only architectural family primarily motivated by a non-accuracy concern (privacy), which explains why its reported accuracy is often comparable to, rather than better than, centralised counterparts.

5 | Benchmark Datasets

The choice of dataset has a decisive impact on reported performance and on the operational relevance of conclusions. We summarise the most widely used NIDS datasets in Table 3.

5.1 | KDD Cup 99 and NSL-KDD

The KDD Cup 99 dataset [46] was derived from the 1998 DARPA Intrusion Detection Evaluation. It has been heavily criticised for redundant records, inconsistent labelling, and outdated attacks. NSL-KDD [17] removed duplicates and rebalanced the test set, but inherits the same underlying traffic. Despite being obsolete in operational terms, both remain popular as historical baselines.

5.2 | UNSW-NB15

UNSW-NB15 [18] was generated by the IXIA PerfectStorm tool at the Australian Centre for Cyber Security in 2015. It contains 49 features and 9 attack categories (Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms) with approximately 2.5M records, providing more modern attack patterns than KDD.

5.3 | CIC-IDS2017 and CSE-CIC-IDS2018

The Canadian Institute for Cybersecurity (CIC) released CIC-IDS2017 [19] and CSE-CIC-IDS2018, generated in a realistic testbed and labelled with attacks including brute-force, DoS, DDoS, Heartbleed, web attacks, infiltration, and botnet. Flow features are extracted with CICFlowMeter and number around 80. These datasets are now de-facto standards but are known to suffer from label noise and class imbalance [47].

5.4 | IoT-specific Datasets

The proliferation of IoT motivated dedicated datasets:

- **Bot-IoT [48]:** realistic IoT botnet traffic generated at UNSW Canberra; over 72M flows.
- **ToN-IoT [49]:** telemetry, OS, and network data from a heterogeneous IoT/IIoT testbed.
- **Edge-IIoTset [50]:** cyber-attacks on industrial IoT and edge gateways across 14 attack types.
- **N-BaIoT [51]:** traffic from 9 commercial IoT devices infected with Mirai and BASHLITE.
- **IoT-23 [52]:** 20 malicious and 3 benign IoT traffic captures by Stratosphere Lab.

5.5 | Other Datasets

The CTU-13 [53] dataset is a long-standing botnet benchmark, while ISCXIDS2012, MAWILab, and DARPA OpTC are commonly used for specific scenarios. Recent multi-source datasets such as 5G-NIDD [54] target 5G core traffic.

Table 3. Widely used benchmark datasets for NIDS.

Dataset	Year	Features	#Records	Attack categories	Notes / limitations
---------	------	----------	----------	-------------------	---------------------

KDD Cup 99	1999	41	~5M	4 (DoS, Probe, R2L, U2R)	Outdated, redundant records
NSL-KDD	2009	41	~150K	4	Cleaner KDD, still old traffic
UNSW-NB15	2015	49	~2.5M	9	Modern attacks, mild imbalance
CIC-IDS2017	2017	80	~2.8M	7 super-classes	Label noise reported
CSE-CIC-IDS2018	2018	80	~16M	7 super-classes	Large, AWS-based testbed
CTU-13	2014	flow	~80M	13 botnet scenarios	Botnet-focused
Bot-IoT	2019	46	~72M	4 IoT attack classes	Highly imbalanced
ToN-IoT	2021	44	~22M flows	9	Multi-modal (telemetry+net)
Edge-IIoTset	2022	61	~20M	14	IIoT/edge focus
N-BaIoT	2018	115	~7M	Mirai, BASHLITE	9 commercial IoT devices
IoT-23	2020	flow	~325M pkts	20 malicious captures	Stratosphere Lab
5G-NIDD	2022	52	~1.2M	DoS, port scan, etc.	5G core network

Table 3 illustrates a clear temporal and thematic progression: legacy datasets (KDD Cup 99, NSL-KDD) offer only outdated attack patterns, mid-2010s datasets (UNSW-NB15, CIC-IDS2017/2018) provide more realistic traffic but suffer from documented label noise, and IoT-specific datasets (Bot-IoT, ToN-IoT, Edge-IIoTset, N-BaIoT) address the growing importance of constrained devices but frequently exhibit extreme class imbalance. No single dataset in Table 3 is simultaneously large, modern, balanced, and free of labelling issues, which is why cross-dataset evaluation (Section 7) is essential for assessing real-world generalisation rather than relying on in-dataset accuracy alone.

6 | Performance Metrics and Evaluation

Let TP, TN, FP, FN denote true/false positives and negatives. The most common metrics are:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$$

$$\text{F1} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN})$$

The Receiver Operating Characteristic (ROC) curve plots TPR against FPR; the Area Under the Curve (AUC-ROC) is threshold-free. The Precision–Recall AUC is more informative under heavy class imbalance, which is the norm in NIDS where attacks are rare [55].

Beyond accuracy. Accuracy is misleading on imbalanced datasets: a classifier that always predicts "benign" on Bot-IoT can exceed 99% accuracy. Best practice is to report per-class precision/recall/F1, macro/weighted averages, and the confusion matrix [9]. Operational metrics such as alerts/hour, time-to-detect, and analyst workload are increasingly recommended.

Evaluation protocols. Stratified k-fold cross-validation, temporal splits (train on past, test on future) and cross-dataset evaluation should all be considered. Many published works inadvertently leak information by random shuffling of temporally correlated flows [47, 56].

7 | Comparative Analysis of Recent Works

Table 4 summarises 25 representative works published between 2017 and 2025. Several trends are visible:

- Reported accuracies on NSL-KDD and CIC-IDS2017 saturate above 99%, suggesting that further benchmark gains are unlikely to translate into operational improvement [47,56].
- Hybrid CNN+LSTM and transformer architectures dominate top-results on UNSW-NB15 and CIC-IDS2018.
- IoT datasets (Bot-IoT, ToN-IoT, Edge-IIoTset) increasingly drive new methods, especially federated and lightweight DL.
- GNN-based detectors [41,42] are gaining traction for botnet and lateral-movement detection.
- Self-supervised and pre-trained transformer models [40,26] mark an emerging direction.

Table 4. Representative recent ML/DL NIDS studies (2017–2025).

Ref.	Year	Method	Dataset	Acc. (%)	F1 (%)	Key contribution
[17]	2009	DT/SVM/NN	NSL-KDD	82.0	–	Cleaned KDD baseline
[20]	2016	Random Forest	NSL-KDD	99.6	99.6	RF for IDS
[34]	2017	RNN	NSL-KDD	83.3	–	Deep RNN-IDS
[30]	2017	1-D CNN (raw)	ISCX2012	99.4	–	End-to-end byte CNN
[8]	2018	Stacked NDAE	KDD/NSL-KDD	97.9	97.8	Non-symmetric AE
[7]	2019	DNN	6 datasets	93–99	93–99	Comprehensive DL benchmark
[37]	2018	GAN (IDSGAN)	NSL-KDD	–	–	Adversarial NIDS evasion
[43]	2019	CNN+LSTM	CIC-IDS2017	99.4	99.0	Hybrid spatio-temporal
[57]	2020	DL survey	multi	–	–	Survey of DL NIDS
[58]	2019	Survey	–	–	–	Dataset survey
[59]	2020	Stacking ensemble	NSL-KDD/UNSW	99.0	99.0	Stacking C5.0 + OC-SVM
[23]	2021	RF+Bayesian opt.	CIC-IDS2017	99.6	99.6	Multi-stage optimised ML
[60]	2021	BiLSTM	NSL-KDD	94.3	94.0	Bidirectional LSTM
[61]	2021	GAN ensemble	CIC-IDS2017	99.7	99.7	Adversarial autoencoders
[45]	2021	Federated DNN	Bot-IoT	99.5	99.5	Federated IoT NIDS
[62]	2022	DNN + clustering	IoT-23	99.0	99.0	Hybrid IoT NIDS
[41]	2022	E-GraphSAGE (GNN)	ToN-IoT/Bot-IoT	99.0	99.0	Edge-feature GNN
[26]	2022	Transformer (RTIDS)	CIC-IDS2017	99.4	99.4	Self-attention NIDS
[40]	2022	BERT (ET-BERT)	ISCX/USTC	98.9	98.8	Pre-trained traffic LM
[50]	2022	DL+Edge-IIoTset	Edge-IIoTset	94.7	94.6	Federated IIoT NIDS
[63]	2022	Tree ensembles	NF-UQ-NIDS	99.0	99.0	NetFlow-standard datasets
[42]	2022	Self-sup. GNN	ToN-IoT	99.0	99.0	Anomal-E unsupervised
[39]	2024	Transformer	4 datasets	99.4	99.4	Modular transformer NIDS
[64]	2023	Cross-dataset eval	multi	60–90	60–90	Generalisation study
[65]	2023	FL+SSL	Edge-IIoTset	96.0	95.5	Self-sup. federated NIDS

Table 4 confirms two trends noted throughout this survey: reported accuracy on NSL-KDD and early CIC-IDS releases has plateaued above 99%, while newer IoT- and cloud-oriented datasets (Bot-IoT, ToN-IoT, Edge-IIoTset) still show meaningful differentiation between methods, making them more informative testbeds for future work. Transformer- and GNN-based models [26,39,40,41,42] occupy the upper end of reported F1-scores on these newer datasets, but the sparse reporting of false-positive rate and computational cost in most cited works (marked "-" in the table) makes it difficult to assess their true operational readiness.

Caveats. Direct comparison across rows of Table 4 should be made cautiously: differences in pre-processing, splits, feature subsets, and metric definitions can change reported accuracy by several percentage points [47, 56, 64]. Cross-dataset evaluations [64] consistently show large generalisation gaps—models trained on one dataset can drop 20–30 percentage points in F1 on another.

8 | Open Challenges and Future Directions

8.1 | Class Imbalance and Rare Attacks

Real traffic is overwhelmingly benign; some attack classes account for <0.01% of flows. Standard remedies include resampling (SMOTE [66], ADASYN), cost-sensitive learning, focal loss, and GAN-based augmentation [37,61]. Few-shot and meta-learning approaches are promising but under-explored [67].

8.2 | Concept Drift

Attacker behaviour, services, and user patterns drift over time, causing performance decay. Online learning, drift detectors (ADWIN, DDM), and continual-learning techniques (replay buffers, elastic weight consolidation) are needed for sustainable production NIDS [68].

8.3 | Adversarial Robustness

ML/DL NIDS are vulnerable to evasion (adversarial perturbations of malicious flows) and poisoning (manipulation of training data) [9, 37-38]. Adversarial training, certified defences, and detection of evasion attempts remain open problems, particularly under realistic constraints on packet manipulation.

8.4 | Explainability

Security analysts need to understand why a flow is flagged. Post-hoc explainers (SHAP, LIME, integrated gradients) and inherently interpretable models are increasingly integrated into NIDS [69,70]. Explanations also help debug datasets, exposing label noise and shortcut features.

8.5 | Real-Time Inference and Edge NIDS

At 100 Gb/s line rates, a NIDS may have a few microseconds per packet. Lightweight DL (knowledge distillation, quantisation, pruning), in-network ML on programmable switches (P4), and smart-NIC offloading are active research areas [71,72].

8.6 | Federated and Privacy-Preserving NIDS

Cross-organisation collaboration improves coverage but raises privacy concerns. FL [13, 45, 65], secure aggregation, differential privacy, and homomorphic encryption are being combined for collaborative NIDS. Open issues include non-IID traffic, communication overhead, and resilience to malicious clients.

8.7 | Self- and Semi-Supervised Pre-Training

Labels are expensive in security. Self-supervised pre-training on large unlabelled corpora, followed by fine-tuning, has revolutionised NLP and vision and is now reaching NIDS via traffic-language models [40,26,42]. This paradigm could yield foundation models for network security, Emerging 2024–2025 studies [76-79] provide early evidence that contrastive and LLM-based pre-training can transfer across datasets more effectively than supervised-only baselines, though systematic cross-domain evaluation remains limited.

8.8 | Encrypted Traffic and Privacy-Preserving Detection

TLS 1.3 and QUIC encrypt almost all payloads. Detection must rely on metadata (sizes, timings, certificates, JA3/JA4 fingerprints) and learn over encrypted flows [40, 72]. Deep models that respect user privacy while retaining detection power are an open challenge.

8.9 | Standardised Datasets and Reproducibility

Recent work [47, 56, 63-64, 73-80] highlights pervasive issues in NIDS research: data leakage, inappropriate splits, label noise, and missing baselines. Adoption of standardised NetFlow features (NF-UQ-NIDS-v2 [63]), open code, and rigorous protocols is essential.

9 | Conclusion

We have collected a decade of research in ML- and DL-based NIDS. We have presented a unified taxonomy of classical ML and current DL families (CNN, RNN, AE, GAN, transformer, GNN, federated, hybrid) and discussed the most popular benchmark datasets, their limitations, and compared their reported results in 25 published recent papers. Our results suggest that:

- Tree ensembles remain strong, and often hard-to-beat baselines on tabular flow features - any DL contribution should clearly justify its added complexity.
- Hybrid spatio-temporal architectures (CNN+LSTM) and transformer-based models have the highest reported scores on modern datasets, but accuracy figures on saturated benchmarks are no longer a meaningful differentiator.
- Our cross-dataset tests show significant generalisation gaps, and standardised features such as NF-UQ-NIDS-v2 and rigorous evaluation are implemented in these cases.
- The frontier of the field is moving from pure accuracy towards operational concerns: real-time edge inference, adversarial robustness, explainability, federated training and privacy-preserving detection over encrypted traffic. We hope this brief survey will serve as a starting point for researchers and practitioners and support the design of the next generation of trustworthy, robust and deployable NIDS.

Funding

This research has no funding source.

Data Availability

The data and code used in this study are not publicly available due to privacy and confidentiality considerations.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

All authors contributed equally to this study.

References

- [1] 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(1):e4150. <https://doi.org/10.1002/ett.4150>

- [2] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*. 2019;2(1):1-22. <https://doi.org/10.1186/s42400-019-0038-7>
- [3] IBM Security. Cost of a Data Breach Report 2023. IBM Corporation; 2023.
- [4] Liao HJ, Lin CHR, Lin YC, Tung KY. Intrusion detection system: A comprehensive review. *J Netw Comput Appl*. 2013;36(1):16-24. <https://doi.org/10.1016/j.jnca.2012.09.004>
- [5] Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun Surveys Tuts*. 2016;18(2):1153-1176. <https://doi.org/10.1109/COMST.2015.2494502>
- [6] LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436-444. <https://doi.org/10.1038/nature14539>
- [7] Vinayakumar R, Alazab M, Soman KP, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. *IEEE Access*. 2019;7:41525-41550. <https://doi.org/10.1109/ACCESS.2019.2895334>
- [8] Shone N, Ngoc TN, Phai VD, Shi Q. A deep learning approach to network intrusion detection. *IEEE Trans Emerg Topics Comput Intell*. 2018;2(1):41-50. <https://doi.org/10.1109/TETCI.2017.2772792>
- [9] Apruzzese G, Laskov P, Montes de Oca E, Mallouli W, Brdalo Rapa L, Vladimirov A, Vega LF. SoK: The impact of unlabelled data in cyberthreat detection. In: *Proc IEEE Eur Symp Security Privacy (EuroS&P)*; 2022.
- [10] Sommer R, Paxson V. Outside the closed world: On using machine learning for network intrusion detection. In: *Proc IEEE Symp Security Privacy*; 2010. p. 305-316. <https://doi.org/10.1109/SP.2010.25>
- [11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998–6008. <https://doi.org/10.48550/arXiv.1706.03762>
- [12] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.
- [13] McMahan, B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, 54, 1273–1282. <https://doi.org/10.48550/arXiv.1602.05629>
- [14] Aldweesh A, Derhab A, Emam AZ. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowl-Based Syst*. 2020;189:105124. <https://doi.org/10.1016/j.knosys.2019.105124>
- [15] Nguyen TT, Reddi VJ. Deep reinforcement learning for cyber security. *IEEE Trans Neural Netw Learn Syst*. 2021. <https://doi.org/10.1109/TNNLS.2021.3121870>
- [16] Breiman L. Random forests. *Mach Learn*. 2001;45(1):5-32. <https://doi.org/10.1023/A:1010933404324>
- [17] Tavallaei M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: *IEEE Symp Comput Intell Security Defense Appl (CISDA)*; 2009. p. 1-6. <https://doi.org/10.1109/CISDA.2009.5356528>
- [18] Moustafa N, Slay J. UNSW-NB15: A comprehensive data set for network intrusion detection systems. In: *Mil Commun Inf Syst Conf (MilCIS)*; 2015. p. 1-6. <https://doi.org/10.1109/MilCIS.2015.7348942>
- [19] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: *Proc Int Conf Inf Syst Security Privacy (ICISSP)*; 2018. p. 108-116. <https://doi.org/10.5220/0006639801080116>
- [20] Farnaaz N, Jabbar MA. Random forest modeling for network intrusion detection system. *Procedia Comput Sci*. 2016;89:213-217. <https://doi.org/10.1016/j.procs.2016.06.047>
- [21] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. In: *Proc 22nd ACM SIGKDD Int Conf Knowl Discovery Data Mining*; 2016. p. 785-794. <https://doi.org/10.1145/2939672.2939785>
- [22] Dhaliwal SS, Nahid AA, Abbas R. Effective intrusion detection system using XGBoost. *Information*. 2018;9(7):149. <https://doi.org/10.3390/info9070149>
- [23] Injadat M, Moubayed A, Nassif AB, Shami A. Multi-stage optimized machine learning framework for network intrusion detection. *IEEE Trans Netw Service Manage*. 2021;18(2):1803-1816. <https://doi.org/10.1109/TNSM.2020.3014205>
- [24] Horng SJ, et al. A novel intrusion detection system based on hierarchical clustering and support vector machines. *Expert Syst Appl*. 2011;38(1):306-313. <https://doi.org/10.1016/j.eswa.2010.06.066>
- [25] Liao Y, Vemuri VR. Use of K-nearest neighbor classifier for intrusion detection. *Comput Security*. 2002;21(5):439-448. [https://doi.org/10.1016/S0167-4048\(02\)00514-X](https://doi.org/10.1016/S0167-4048(02)00514-X)
- [26] Panda M, Patra MR. Network intrusion detection using Naive Bayes. *Int J Comput Sci Netw Security*. 2007;7(12):258-263.
- [27] Liu FT, Ting KM, Zhou ZH. Isolation forest. In: *Proc 8th IEEE Int Conf Data Mining*; 2008. p. 413-422. <https://doi.org/10.1109/ICDM.2008.17>
- [28] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. *Neural Comput*. 2001;13(7):1443-1471. <https://doi.org/10.1162/089976601750264965>
- [29] Wang W, Sheng Y, Wang J, Zeng X, Ye X, Huang Y, Zhu M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access*. 2018;6:1792-1806. <https://doi.org/10.1109/ACCESS.2017.2780250>
- [30] Wang W, Zhu M, Zeng X, Ye X, Sheng Y. Malware traffic classification using convolutional neural network for representation learning. In: *Int Conf Inf Netw (ICOIN)*; 2017. p. 712-717. <https://doi.org/10.1109/ICOIN.2017.7899588>
- [31] Kim J, Kim J, Kim H, Shim M, Choi E. CNN-based network intrusion detection against denial-of-service attacks. *Electronics*. 2020;9(6):916. <https://doi.org/10.3390/electronics9060916>
- [32] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9(8):1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

- [33] Kim J, Kim J, Thu HLT, Kim H. Long short term memory recurrent neural network classifier for intrusion detection. In: Int Conf Platform Technol Service (PlatCon); 2016. p. 1-5. <https://doi.org/10.1109/PlatCon.2016.7456805>
- [34] Yin C, Zhu Y, Fei J, He X. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access. 2017;5:21954-21961. <https://doi.org/10.1109/ACCESS.2017.2762418>
- [35] An J, Cho S. Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE. 2015;2:1-18.
- [36] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, & Bengio Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2672–2680.
- [37] Lin Z, Shi Y, Xue Z. IDSGAN: Generative adversarial networks for attack generation against intrusion detection. arXiv preprint arXiv:1809.02077; 2018.
- [38] Usama M, Asim M, Latif S, Qadir J, Al-Fuqaha A. Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems. In: 15th Int Wireless Commun Mobile Comput Conf (IWCMC); 2019. p. 78-83. <https://doi.org/10.1109/IWCMC.2019.8766353>
- [39] Manocchio LD, Layeghy S, Lo WW, Kulatilleke GK, Sarhan M, Portmann M. FlowTransformer: A transformer framework for flow-based network intrusion detection systems. Expert Syst Appl. 2024;241:122564. <https://doi.org/10.1016/j.eswa.2023.122564>
- [40] Lin X, Xiong G, Gou G, Li Z, Shi J, Yu J. ET-BERT: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. In: Proc ACM Web Conf (WWW); 2022. p. 633-642. <https://doi.org/10.1145/3485447.3512217>
- [41] Lo WW, Layeghy S, Sarhan M, Gallagher M, Portmann M. E-GraphSAGE: A graph neural network based intrusion detection system for IoT. In: NOMS 2022 IEEE/IFIP Netw Oper Manage Symp; 2022. p. 1-9. <https://doi.org/10.1109/NOMS54207.2022.9789878>
- [42] Caville E, Lo WW, Layeghy S, Portmann M. Anomal-E: A self-supervised network intrusion detection system based on graph neural networks. Knowl-Based Syst. 2022;258:110030. <https://doi.org/10.1016/j.knosys.2022.110030>
- [43] Khan FA, Gumaei A, Derhab A, Hussain A. A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access. 2019;7:30373-30385. <https://doi.org/10.1109/ACCESS.2019.2899721>
- [44] Nguyen TD, Marchal S, Miettinen M, Fereidooni H, Asokan N, Sadeghi AR. DfIoT: A federated self-learning anomaly detection system for IoT. In: 39th IEEE Int Conf Distrib Comput Syst (ICDCS); 2019. <https://doi.org/10.1109/ICDCS.2019.00080>
- [45] Popoola SI, Ande R, Adebisi B, Gui G, Hammoudeh M, Jogunola O. Federated deep learning for zero-day botnet attack detection in IoT edge devices. IEEE Internet Things J. 2021;9(5):3930-3944. <https://doi.org/10.1109/JIOT.2021.3100755>
- [46] University of California, Irvine. KDD Cup 1999 Data. 1999. Available: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
- [47] Engelen G, Rimmer V, Joosen W. Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In: IEEE Security Privacy Workshops (SPW); 2021. p. 7-12. <https://doi.org/10.1109/SPW53761.2021.00009>
- [48] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Future Gener Comput Syst. 2019;100:779-796. <https://doi.org/10.1016/j.future.2019.05.041>
- [49] Moustafa N. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustainable Cities Soc. 2021;72:102994. <https://doi.org/10.1016/j.scs.2021.102994>
- [50] Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access. 2022;10:40281-40306. <https://doi.org/10.1109/ACCESS.2022.3165809>
- [51] Meidan Y, Bohadana M, Mathov Y, Mirsky Y, Shabtai A, Breitenbacher D, Elovici Y. N-BaIoT: Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. 2018;17(3):12-22. <https://doi.org/10.1109/MPRV.2018.03367731>
- [52] Garcia S, Parmisano A, Erquiaga MJ. IoT-23: A labeled dataset with malicious and benign IoT network traffic. Stratosphere Lab., Tech. Rep.; 2020. <https://doi.org/10.5281/zenodo.4743746>
- [53] García S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. Comput Security. 2014;45:100-123. <https://doi.org/10.1016/j.cose.2014.05.011>
- [54] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, & Bengio Y. (2014). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2672–2680.
- [55] Davis J, Goadrich M. The relationship between precision-recall and ROC curves. In: Proc 23rd Int Conf Mach Learn (ICML); 2006. p. 233-240. <https://doi.org/10.1145/1143844.1143874>
- [56] Arp D, Quiring E, Pendlebury F, Warnecke A, Pierazzi F, Sterling C, Wressnegger C, & Rieck K. (2022). Dos and don'ts of machine learning in computer security. In *Proceedings of the 31st USENIX Security Symposium* (pp. 3971–3988). USENIX Association.
- [57] Ferrag MA, Maglaras L, Moschogiannis S, Janicke H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J Inf Security Appl. 2020;50:102419. <https://doi.org/10.1016/j.jisa.2019.102419>
- [58] Ring M, Wunderlich S, Scheuring D, Landes D, Hotho A. A survey of network-based intrusion detection data sets. Comput Security. 2019;86:147-167. <https://doi.org/10.1016/j.cose.2019.06.005>

- [59] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J, Alazab A. Hybrid intrusion detection system based on the stacking ensemble of C5 decision tree classifier and one class support vector machine. *Electronics*. 2020;9(1):173. <https://doi.org/10.3390/electronics9010173>
- [60] Imrana Y, Xiang Y, Ali L, Abdul-Rauf Z. A bidirectional LSTM deep learning approach for intrusion detection. *Expert Syst Appl*. 2021;185:115524. <https://doi.org/10.1016/j.eswa.2021.115524>
- [61] Andresini G, Appice A, De Rose L, Malerba D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. *Future Gener Comput Syst*. 2021;123:108-127. <https://doi.org/10.1016/j.future.2021.04.017>
- [62] de Souza CA, Westphall CB, Machado RB, Loffi L, Westphall CM, Geronimo GA. Intrusion detection and prevention in fog based IoT environments: A systematic literature review. *Comput Netw*. 2022;214:109154. <https://doi.org/10.1016/j.comnet.2022.109154>
- [63] Sarhan M, Layeghy S, Portmann M. Towards a standard feature set for network intrusion detection system datasets. *Mobile Netw Appl*. 2022;27:357-370. <https://doi.org/10.1007/s11036-021-01843-0>
- [64] Layeghy S, Portmann M. Explainable cross-domain evaluation of ML-based network intrusion detection systems. *Comput Electr Eng*. 2023;108:108692. <https://doi.org/10.1016/j.compeleceng.2023.108692>
- [65] Aouedi O, Piamrat K, Muller G, Singh KD. Federated semisupervised learning for attack detection in industrial Internet of Things. *IEEE Trans Ind Informat*. 2023;19(1):286-295. <https://doi.org/10.1109/TII.2022.3170733>
- [66] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. *J Artif Intell Res*. 2002;16:321-357. <https://doi.org/10.1613/jair.953>
- [67] Wang Y, Yao Q, Kwok JT, Ni LM. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput Surveys*. 2020;53(3):63:1-63:34. <https://doi.org/10.1145/3386252>
- [68] Andresini G, Pendlebury F, Pierazzi F, Loglisci C, Appice A, Cavallaro L. INSOMNIA: Towards concept-drift robustness in network intrusion detection. In: *Proc 14th ACM Workshop Artif Intell Security (AISec)*; 2021. p. 111-122. <https://doi.org/10.1145/3474369.3486863>
- [69] Wang M, Zheng K, Yang Y, Wang X. An explainable machine learning framework for intrusion detection systems. *IEEE Access*. 2020;8:73127-73141. <https://doi.org/10.1109/ACCESS.2020.2988359>
- [70] Barredo Arrieta A, et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf Fusion*. 2020;58:82-115. <https://doi.org/10.1016/j.inffus.2019.12.012>
- [71] Xiong Z, Zilberman N. Do switches dream of machine learning? Toward in-network classification. In: *Proc 18th ACM Workshop Hot Topics Netw (HotNets)*; 2019. <https://doi.org/10.1145/3365609.3365864>
- [72] Anderson B, Paul S, McGrew D. Deciphering malware's use of TLS (without decryption). *J Comput Virol Hacking Tech*. 2018;14:195-211. <https://doi.org/10.1007/s11416-017-0306-6>
- [73] Khan MA. HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system. *Processes*. 2021;9(5):834. <https://doi.org/10.3390/pr9050834>
- [74] Rahman SA, Tout H, Talhi C, Mourad A. Internet of things intrusion detection: Centralized, on-device, or federated learning? *IEEE Network*. 2020;34(6):310-317. <https://doi.org/10.1109/MNET.001.1900506>
- [75] Wu Z, Zhang H, Wang P, Sun Z. RTIDS: A robust transformer-based approach for intrusion detection system. *IEEE Access*. 2022;10:64375-64387. <https://doi.org/10.1109/ACCESS.2022.3182333>
- [76] Kheddar H. Transformers and large language models for efficient intrusion detection systems: A comprehensive survey. *Inf Fusion*. 2025;124:103347. <https://doi.org/10.1016/j.inffus.2025.103347>
- [77] Long Z, Yan H, Shen G, Zhang X, He H, Cheng L. A transformer-based network intrusion detection approach for cloud security. *J Cloud Comput*. 2024;13(5). <https://doi.org/10.1186/s13677-023-00574-9>
- [78] Zhang, H., Sediq, A. B., Afana, A., & Erol-Kantarci, M. (2024). Large language models in wireless application design: In-context learning-enhanced automatic network intrusion detection. In *Proceedings of the IEEE Global Communications Conference (GLOBECOM 2024)* (pp. 2479–2484). IEEE.
- [79] Koukoulis, I., Syrigos, I., & Korakis, T. (2025). Self-supervised transformer-based contrastive learning for intrusion detection systems. *arXiv preprint arXiv:2505.08816*.
- [80] Ghadermazi, J., Hore, S., Shah, A., & Bastian, N. D. (2025). GTAE-IDS: Graph transformer-based autoencoder framework for real-time network intrusion detection. *IEEE Transactions on Information Forensics and Security*, 20, 4026–4041. <https://doi.org/10.1109/TIFS.2025.3557741>

Disclaimer/Publisher's Note: The perspectives, opinions, and data shared in all publications are the sole responsibility of the individual authors and contributors, and do not necessarily reflect the views of Sciences Force or the editorial team. Sciences Force and the editorial team disclaim any liability for potential harm to individuals or property resulting from the ideas, methods, instructions, or products referenced in the content.