Frontiers in Computing and Intelligent Systems
ISSN: 2832-6024 | Vol. 18, No. 2, 2026

Machine Learning-Based IoT Intrusion Detection: A Survey and Performance Comparison

Ruixi He *

Department of Electronic and Information Engineering, Zhuhai College of Science and Technology, Zhuhai, Guangdong, 519000, China
*** Corresponding author Email:** FerrayTris@163.com

Abstract: As IoT devices multiply, the cyber attack surface has widened just as fast, and with it the need for intrusion detection systems (IDS) that are efficient, dependable, and protective of user privacy. This paper organizes IoT IDS methods into a taxonomy built around training architecture and reviews them under two main paradigms, misuse detection and anomaly detection, with the focus on centralized traditional machine learning and deep learning. Experiments on several public datasets point to a clear pattern: in centralized settings, tree-based ensemble models and the hybrid CNN-LSTM architecture give the strongest overall results, whereas Isolation Forest holds up better once the data is compressed at the edge. For federated deployment, tree-based ensembles stand out as the preferred local classifiers.

Keywords: Machine Learning; Deep Learning; Internet of Things; Intrusion Detection System.

1. Introduction

Recent market surveys point to a steady rise in the number of Internet of Things (IoT) devices. Cellular IoT connections worldwide are projected to hit 4.7 billion by the end of 2025, up 13% from 4.1 billion in 2024 [1]. The pace over the past five years has been quick, with a 27% jump recorded in 2022 alone [1] That expansion has a downside, though: the more devices come online, the wider the attack surface grows. Figures for the first half of 2025 show global Distributed Denial of Service (DDoS) attacks up 83% against the same period in 2024 [2], and smart home devices are expected to face roughly 29 attacks a day in 2025, about ten times the 2024 figure [3]. As attacks grow more sophisticated and harder to spot, an effective Intrusion Detection System (IDS) has become essential to keeping IoT devices secure.

Rule-based and signature-based IDSs keep the false alarm rate low, but they struggle with complex or previously unseen attacks. This is where machine learning based IDSs come in, since they learn from new situations and adapt, which is exactly what the older methods could not do against unfamiliar attacks. Under supervised learning, misuse detection picks up attack signatures on its own and cuts the cost of maintaining rules by hand. Under unsupervised learning, anomaly detection gives early warning of zero-day attacks, though it tends to raise more false alarms in the process. Federated learning (FL) adds another useful angle: devices exchange updated model parameters instead of their raw data, so they can pool what they have learned without exposing it, sharpening the shared model and helping surface new attack patterns. Against this backdrop, the present paper examines the strengths, weaknesses, and practical viability of the various detection techniques and machine learning (ML) models used for IoT intrusion detection.

2. Classification Framework and Methodology

2.1. Classification System

The present research on IoT intrusion detection is organized into a taxonomy built around the training architecture, shown in Figure 1. Every algorithm in this framework falls into one of two IDS categories. Supervised learning models, covering both traditional machine learning and deep learning (DL), sit under misuse detection: they rely on labeled data to recognize known attacks. Unsupervised learning models fall under anomaly detection instead, flagging threats by watching for departures from the normal traffic distribution. Federated learning in the distributed branch is a training architecture rather than a detection paradigm, whose local models can adopt either of the two paradigms according to scenario requirements. Besides, the diagram does not cover all relevant algorithms exhaustively, but focuses on representative and widely used algorithms in current machine learning-based intrusion detection research.

The framework organizes ML-based IoT intrusion detection into a five-tier hierarchy. Starting from this overarching starting tier, it systematically evaluates the trade-offs and detection performance of various ML paradigms. To this end, the second tier splits methods by training architecture into centralized and distributed machine learning. This setup prevents dimensional confusion between FL paradigms and specific model architectures, while establishing centralized models as a necessary performance benchmark for distributed FL configurations.

Within the centralized branch, methods are further classified by model complexity, learning paradigm, and specific algorithm. First, distinguish between traditional machine learning and deep learning. Both categories are then subdivided into supervised and unsupervised learning. Semi-supervised learning is noted in the framework but omitted from the classification diagram; given that this topic is beyond the core scope of this review, which focuses on the two most widely adopted paradigms in current IoT IDS research, it will not be discussed in detail here. In traditional methods, supervised approaches include K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF), and XGBoost, whereas K-Means and Isolation Forest (IF) represent unsupervised learning. On the deep learning side, supervised methods include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM), while unsupervised

methods are represented by Variational Autoencoders (VAEs) and Adversarial Networks (GANs).

In the distributed branch, the focus is on FL. The framework divides FL into aggregation optimization algorithms, with FedAvg serving as the baseline method, and local model architectures, such as FedCNN and FedLSTM.

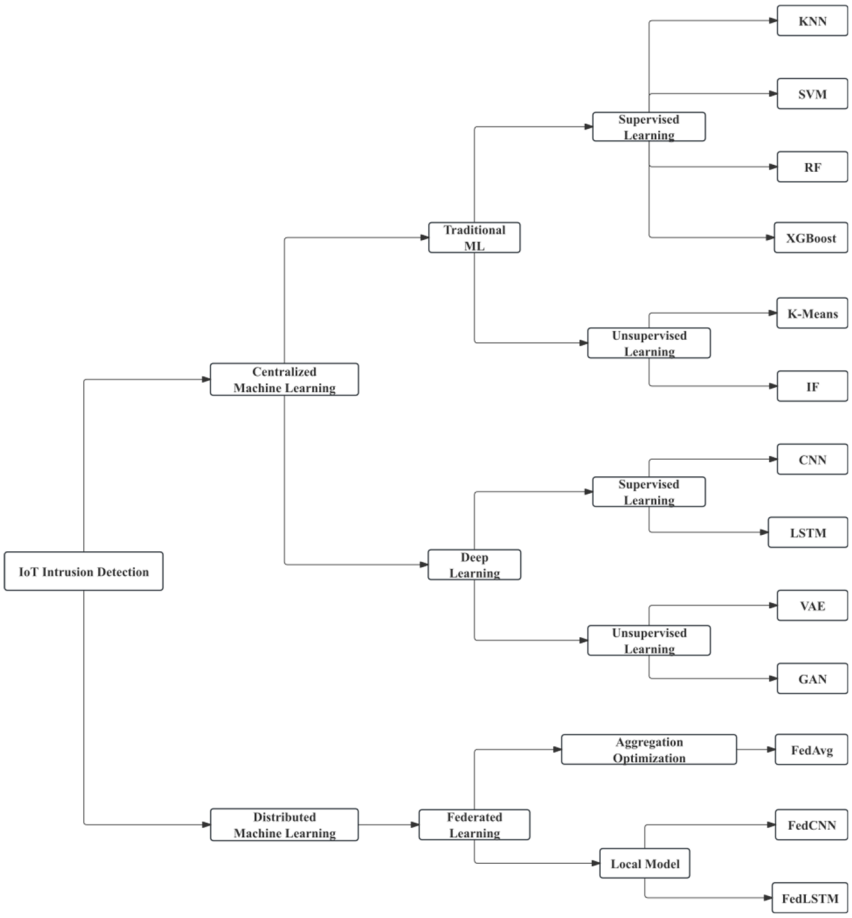


Figure 1. Taxonomy of IoT intrusion detection algorithms (with training architecture as the axis)

2.2. Literature Collection Strategy and Datasets

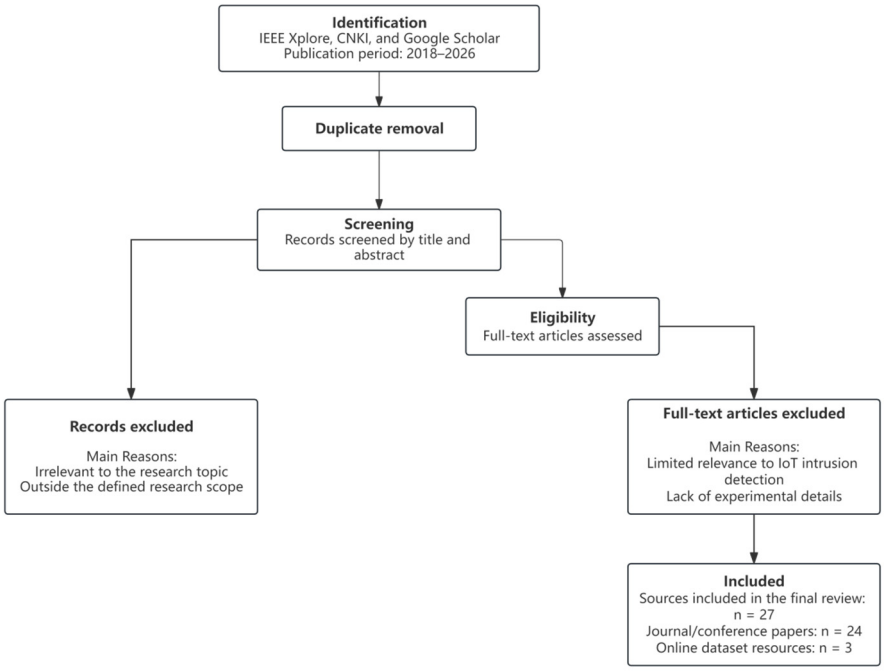


Figure 2. Literature Search and Screening Process

The literature search covered IEEE Xplore, CNKI, and Google Scholar, targeting work published between 2018 and 2026. The queries combined four keywords: 'Internet of Things', 'intrusion detection', 'datasets', and 'machine learning'. Records were then filtered against the inclusion and exclusion criteria, with particular attention to studies on IoT intrusion detection that apply or evaluate machine learning techniques. This left 27 sources for the final review, made up of 24 journal or conference papers and 3 online dataset sources, and Figure 2 traces the full search and selection process. A few foundational and seminal works were kept regardless of the publication window, so that essential background would not be left out.

Public datasets are still the backbone of benchmark testing and experimental replication in IoT security research. Table 1 lists the ones most often used as core benchmarks in current empirical work, among them NSL-KDD [4], UNSW-NB15 [5], CICIDS2017 [6], Bot-IoT [7], TON_IoT [8], IoTID20 [9], and Edge-IIoTset [10]. Between them they span a wide range of traffic patterns and attack scenarios, and it is this breadth of telemetry that makes them suitable for training both centralized and distributed machine learning models.

Table 1. Public benchmark datasets for IoT intrusion detection

Datasets	Year	Attack Type	Core Scene	Public	Reference
NSL-KDD	2009	DoS, Probe, R2L, U2R	Traditional Network Benchmark	Yes	[4]
UNSW-NB15	2015	DoS, Exploit, Fuzzers	Modern Comprehensive Network	Yes	[5]
CICIDS2017	2018	DDoS, Botnet, Web Attack	Enterprise Network	Yes	[6]
Bot-IoT	2019	DDoS, DoS, Scanning	IoT Botnet Forensics	Yes	[7]
TON_IoT	2020	DDoS, Ransomware, Injection	heterogeneous IoT and hybrid industrial scenarios	Yes	[8]
IoTID20	2020	DoS, MitM, Scan	Smart Home IoT	Yes	[9]
Edge-IIoTset	2022	DDoS, Injection, Malware	Edge Industrial IoT	Yes	[10]

2.3. Performance Evaluation Indicators

To keep the comparisons consistent, a single set of metrics was extracted from every study included in the review. Classification performance was judged mainly through the conventional statistical measures — accuracy, precision, recall, F1-score, and the false alarm rate (FAR) — though the weight given to each varied from one paradigm to another. The computational cost of some models was also taken into account, gauged through its training and inference times.

3. Algorithmic Detection Paradigms for IoT Intrusion

Based on the classification framework established in Section 2, this section systematically reviews two categories of methods that form the core detection capability of IoT intrusion detection systems: centralized machine learning methods (including traditional machine learning and deep learning) and distributed machine learning methods.

3.1. Traditional Machine Learning

3.1.1. Supervised Learning Paradigm

The classical supervised machine learning methods in this section mainly follow the misuse detection paradigm. This paradigm is still a basic approach for IoT intrusion detection, because its computational cost is relatively low and its inference process is deterministic [11]. Early studies relied on distance- and boundary-based models such as KNN and SVM to separate benign from malicious traffic using extracted flow features. On common benchmark datasets like NSL-KDD and Bot-IoT, these methods reached good performance when the traffic distribution was not very diverse [12][13]. However, their scalability drops a lot when they are used on modern traffic datasets that are high-dimensional and large-scale, such as CICIDS2017. For KNN, the model needs a large number of distance computations during inference, which causes latency problems in real-time edge streaming scenarios. For SVM, it suffers from excessive training costs and, more importantly, exhibits marked performance degradation on minority attack classes under severe class imbalance [14]. This makes SVM hard to scale for large intrusion detection datasets.

Tree-based ensemble methods — RF and XGBoost in particular — were adopted largely to overcome the scalability limits of these early supervised classifiers, and they have since become a common choice for IoT intrusion detection. RF pairs bootstrap aggregation with random feature selection, which leaves it fairly resistant to noisy or redundant traffic features and allows it to perform well on NSL-KDD [12] [15], CICIDS2017[14], and IoTID20[16]. XGBoost takes a different route: regularized gradient boosting improves its generalization, and it too remains competitive on NSL-KDD [12] and IoTID20 [16].

Tree-based ensembles bring more than detection accuracy on structured tabular traffic: they also produce interpretable feature-importance scores. These scores make attack behaviour easier to understand and feed directly into practical IoT security analysis. Because detection quality, computational efficiency, and interpretability tend to hold together well in these models, they are a natural fit for deployment on resource-constrained IoT edge gateways.

3.1.2. Unsupervised Learning Paradigm

To mitigate the constraints imposed by label scarcity, some unsupervised frameworks such as K-Means and IF have been added to IoT IDS, and they belong to the anomaly detection paradigm. Instead of matching known historical signatures, they share a common idea: they measure anomalies only through the intrinsic distribution deviations and the statistical density changes in unlabeled feature spaces. Therefore, their main application scenarios are the network cold-start phase and real-time edge-telemetry filtering, where prior attack knowledge is often unavailable [17]. K-Means separates traffic clusters by spatial proximity [18], while IF measures the structural sparsity of outliers using randomized tree depths

[19]. By computing these boundary-free data distributions, these methods together allow resource-constrained IoT gateways to catch new and zero-day anomalies with modest configuration effort. Compared with supervised misuse detection, these methods sacrifice partial precision for zero-day attack discovery capability and lower labeling dependency.

3.2. Deep Learning

Unlike traditional machine learning methods that depend on handcrafted traffic features, the deep learning models in this section can learn hierarchical feature representations automatically from raw or lightly processed network traffic. Most of these models are built under the supervised misuse detection paradigm, in which the classifiers are trained with labeled benign and malicious traffic samples. Early deep learning approaches mainly used either CNN or LSTM networks. CNN can extract local spatial patterns from packet payloads and traffic features well, but it is weak at modeling long-term temporal dependencies. LSTM is an improved version of the Recurrent Neural Network (RNN). It keeps the recurrent structure, but it also adds memory cells and gating mechanisms to solve the vanishing gradient problem [20]. As a result, LSTM can model long-term temporal dependencies in sequential network traffic more effectively [21]. However, LSTM is not good at capturing fine-grained spatial correlations. To make up for these two limitations, more and more recent studies use hybrid CNN-LSTM architectures [22]. In these models, the CNN layers first extract discriminative spatial features and reduce the feature dimension, and then the LSTM layers model the temporal dependencies across network flows. Experiments on representative IoT intrusion detection datasets, including TON_IoT and UNSW-NB15, show that CNN-LSTM models usually get higher detection accuracy and lower false-positive rates than a single CNN or LSTM network [22]. This is why CNN-LSTM has become one of the most widely used hybrid deep learning architectures for IoT intrusion detection.

Apart from supervised discriminative models, generative deep learning — chiefly VAEs and GANs — has become a common tool for unsupervised anomaly detection in IoT settings. The two are best seen as complementary rather than rival techniques, since one relies on explicit density estimation and the other on implicit. A VAE optimizes the evidence lower bound and learns a probabilistic latent representation of normal traffic [23], which keeps its training stable and its inference cheap. That combination fits reconstruction-based anomaly detection in real-time IoT intrusion detection systems. The trade-off is that reconstruction objectives tend to over-smooth the learned feature distribution, so complex traffic patterns are not always captured well. GANs approach the problem from the opposite direction. Through adversarial learning they can model intricate data distributions and generate high-fidelity synthetic samples [23], which makes them a strong option for augmenting minority attack classes and easing dataset imbalance. Their drawbacks are equally clear: training is often unstable, and there is no explicit inference step, so a GAN on its own is a poor fit for real-time anomaly detection. These contrasting strengths explain why recent work has increasingly turned to hybrid designs such as VAE-GAN, which pairs the stable latent-representation learning of a VAE with the generative power of a GAN and, in doing so, tends to lift both anomaly detection and data augmentation.

3.3. Federated Learning

FL offers a way around centralized raw-traffic collection and the privacy risks that come with it, working instead as a decentralized, collaborative training framework [25]. The approach was first set out systematically by McMahan et al. in 2017 [26]: a shared global model is trained across many clients, yet the training data never leaves the device it was collected on. Rather than uploading raw network traffic, each client sends only its locally computed model updates to a central aggregator, which keeps the data private while still drawing on resources spread across the network. Since FL makes no assumption about the underlying model, it accommodates both traditional ML algorithms and deep learning, and this flexibility is what makes it appealing for privacy-preserving IoT intrusion detection [25]. As the taxonomy in Figure 1 indicates, FL practice can be examined along two axes: the aggregation optimization algorithm, with FedAvg as the usual baseline, and the architecture of the local model, such as FedCNN and FedLSTM. Plain FedAvg, however, runs into trouble once it is taken out of the ideal setting. Non-IID data distributions, heterogeneous edge devices with limited resources, constrained communication bandwidth, and the threat of model-poisoning attacks all pull against it, and their combined effect can destabilise convergence and drag down detection performance. One response operates at the aggregation stage [27]: a dynamic ensemble strategy replaces plain averaging with Top-K client selection and performance-based weighting, while local SMOTE is added to ease class imbalance. On the Edge-IIoTset dataset this revised scheme attains high accuracy and holds up well even when the system scales to 500 clients [27]. In deployment the aggregation strategy works hand in hand with the local models: FedCNN and FedLSTM serve as the feature-extraction backbones on the edge clients, capturing spatial and sequential patterns in local traffic so that the dynamic ensemble algorithm can coordinate global updates in real IoT environments.

4. Comparative Performance Analysis and Model Selection

Numerous machine learning algorithms have been proposed for IoT intrusion detection, but their performance varies considerably across different datasets and deployment scenarios. To facilitate a systematic comparison, the five tables below summarize the representative performance of traditional machine learning and deep learning approaches on commonly used evaluation metrics.

4.1. Multidimensional Performance Comparison Table

4.1.1. Supervised Learning Paradigm

Table 2 and Table 3 summarizes the detection performance of different traditional machine learning algorithms on the NSL-KDD and CICIDS2017 benchmark, integrating experimental results from three independent studies.

Table 2. Performance comparison on the NSL-KDD dataset

Datasets	Algorithm	Optimizing Strategy	Classification Level	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Reference
NSL-KDD	SVM	Random feature subset selection	Binary	89.31	88.01	87.00	87.50	[12]
	KNN	Random feature subset selection	Binary	98.64	98.59	98.50	98.50	[12]
	RF	Random feature subset selection	Binary	99.48	99.50	99.50	99.50	[12]
	RF	RFE+SMOTE+Optuna	Binary	99.80	99.73	99.73	99.73	[15]
	XGBoost	RFE+SMOTE+Optuna	Binary	99.79	99.70	99.70	99.70	[15]

Table 3. Performance comparison on the CICIDS2017 dataset

Datasets	Algorithm	Optimizing Strategy	Classification Level	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Reference
CICIDS2017	SVM	Normalization + manual tuning	Multi-class (4 classes)	75.21	99.16	75.21	76.60	[14]
	KNN	Normalization + manual tuning + class weight balancing	Multi-class (4 classes)	99.52	99.49	99.52	99.49	[14]
	RF	Normalization + manual tuning + class weight balancing	Multi-class (4 classes)	99.30	99.09	99.30	99.12	[14]

Three separate studies on NSL-KDD and CICIDS2017 point in the same direction: ensemble tree methods and KNN clearly beat SVM at intrusion detection, and the margin between them grows as the datasets get harder.

The same pattern holds across every dataset tested, and it is sharpest when the classes are imbalanced. On NSL-KDD, RF reaches 99.48% accuracy while SVM trails by about 10 percentage points[12]. The gap widens on the heavily skewed CICIDS2017, where SVM accuracy falls away and its F1 drops below 77%, yet KNN and RF both stay at 99% [14]. The reason lies in how SVM draws its boundary: with margin maximisation, the abundant majority samples push the decision boundary toward the minority class, and the method's sensitivity to outliers further erodes recall on rare attacks. Its quadratic optimization cost is a second problem, since it makes SVM impractical at large scale. Tree ensembles behave differently—they split the feature space locally and decide by voting—while KNN works from distance-based local neighbourhoods, so neither is thrown off much by global skew. Taken together, these findings show that the edge SVM enjoys on balanced data simply does not carry over once the data is imbalanced; by leaning on local rather than global decision rules, tree ensembles and KNN prove both more robust and easier to scale.

4.1.2. Unsupervised Learning Paradigm

For public standard IoT datasets, unsupervised anomaly detection generally underperforms supervised misuse detection in overall accuracy. Table 4 focuses on the edge compression scenario unique to the IoT perception layer as supplementary analysis, comparing the performance attenuation of K-Means and Isolation Forest on a real-world IoT temperature sensor testbed (with unpublished datasets). The two algorithms represent spatial-distance-based and random-partition-based anomaly detection paradigms respectively.

Table 4. Comparison of K-Means and Isolation Forest under edge compression on IoT temperature sensor

Datasets	Algorithm	Optimizing Strategy	Classification Level	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Reference
Original (Uncompressed)	K-Means	None (Baseline)	Binary	99.95	99.99	99.95	99.97	[17]
Compressed (Post-edge)	K-Means	Edge Compression (K & e)	Binary	96.08	99.30	96.52	97.89	[17]
Original (Uncompressed)	Isolation Forest	None (Baseline)	Binary	100.00	100.00	100.00	100.00	[17]
Compressed (Post-edge)	Isolation Forest	Edge Compression (K & e)	Binary	99.52	99.78	99.72	99.75	[17]

For both K-Means and IF, edge compression pulls every performance metric below the uncompressed baseline, but IF loses far less ground. The difference traces back to how each model works: IF isolates anomalies through random cuts, so the local density distortions that averaging introduces during compression barely affect its partition-based mechanism. K-Means, resting on distance-based centroids, is more exposed, since the fine-grained temporal variation it depends on is largely washed out once the data is compressed. The numbers bear this out—IF's accuracy slips by only 0.48 percentage points, against a 3.87-point drop for K-Means. Recall tells the same story, with IF holding 99.72% on compressed data while K-Means falls to 96.52%, so IF still catches genuine anomalies even after the data is reduced [17]. Precision stays

high for both, 99.78% for IF and 99.30% for K-Means, yet the margin is wide enough to open a clear F1-score gap in IF's favour [17]. False alarms are less straightforward: IF records zero false alarms on the original dataset, but no false alarm rate is given for the compressed case, because the original experiments never report the post-compression sample sizes. That gap does little to change the overall picture, since IF's recall and precision on compressed data stay ahead of K-Means on every metric reported. Delivering high accuracy at a much smaller data volume, IF makes for a more dependable edge-cloud trade-off than K-Means, holding detection fidelity and real-time responsiveness together rather than trading one for the other.

4.2. Deep Learning

Table 5 reports the detection metrics of CNN-LSTM, CNN, and RNN on two IoT benchmarks, which gives a common ground for comparing the three models. In the study referenced here, all of them run through the same preprocessing pipeline, so the comparison stays fair.

Table 5. Performance comparison on the ToN_IoT and UNSW-NB15 datasets

Datasets	Algorithm	Optimizing Strategy	Classification Level	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FAR (%)	Reference
ToN_IoT	CNN-LSTM	GWO + SMOTE + Dual-Stage Scaling	Binary	99.68	99.71	99.64	99.68	0.92	[22]
ToN_IoT	CNN	GWO + SMOTE + Dual-Stage Scaling	Binary	99.52	99.48	99.55	99.52	1.65	[22]
ToN_IoT	RNN	GWO + SMOTE + Dual-Stage Scaling	Binary	99.45	99.41	99.48	99.45	1.87	[22]
UNSW-NB15	CNN-LSTM	GWO + SMOTE + Dual-Stage Scaling	Binary	92.05	91.84	92.28	92.06	12.67	[22]
UNSW-NB15	CNN	GWO + SMOTE + Dual-Stage Scaling	Binary	90.12	89.94	90.31	90.13	15.17	[22]

Table 5. (Continued)

Datasets	Algorithm	Optimizing Strategy	Classification Level	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FAR (%)	Reference
UNSW-NB15	RNN	GWO + SMOTE + Dual-Stage Scaling	Binary	89.45	89.22	89.68	89.45	16.10	[22]

Looking across the TON_IoT and UNSW-NB15 benchmarks, the three architectures fall into a stable ordering, with the hybrid CNN-LSTM staying ahead of the standalone CNN and RNN throughout. All three, however, lose noticeable accuracy once the setting shifts from the device-centric TON_IoT to the network-centric UNSW-NB15. The main reason is the difference in dataset complexity. TON_IoT separates its classes fairly cleanly, thanks to distinct IoT telemetry patterns, whereas UNSW-NB15 mixes more heterogeneous and overlapping network-level attack profiles that give simpler feature extractors trouble. On TON_IoT, CNN-LSTM reaches 99.68% accuracy [22], only slightly above CNN and RNN, but its FAR of 0.92% is about half that of either, so it keeps false alarms lower when the classes are already well separated. The gap grows on the harder UNSW-NB15, where CNN-LSTM records 92.05% accuracy against 90.12% for CNN and 89.45% for RNN. Its FAR of 12.67% is likewise better than the CNN's 15.17% and the RNN's 16.10% [22], which points to steadier behavior once class boundaries blur. Taken together, these numbers suggest that the preprocessing pipeline, built on GWO-based feature selection and SMOTE balancing, gives all the models a solid starting point, as their results on both datasets show. It is also worth noting that CNN-LSTM degrades less than the standalone CNN and RNN in moving from TON_IoT to the tougher UNSW-NB15, hinting that its combined spatial-temporal processing copes better with dataset shift. The practical read is that architectural choices matter more in messy, heterogeneous traffic than in clean, well-separated benchmarks, where such choices carry far less weight.

CNN and LSTM models detect attacks accurately, but they lean heavily on large amounts of labeled data, and that is a real weakness in IoT settings, where encryption and rare attacks make samples scarce and classes badly imbalanced. Generative augmentation is one way around this. Table 6 sets VAE and GAN, two representative strategies, side by side under a single evaluation framework. The original benchmark looks at Tor traffic, but the focus here is on how the two generators compare in general, a question that does not hinge on any one anomaly type.

Table 6. Comprehensive Three-Dimensional Evaluation between VAE and GAN for Encrypted Traffic Anomaly Detection

Datasets	Algorithm	Accuracy	FAR (%)	Fidelity (JS Div. ↓)	Privacy (MIA AUC ↓)	Training Time	Memory Usage	Reference
UNSW-NB15	VAE	99.23%	2.3%	0.213	0.612	18min	4.2GB	[24]
	GAN	99.17%	3.6%	0.188	0.824	42min	8.1GB	[24]

On the UNSW-NB15 dataset, VAE and GAN come out almost even on the usual utility metrics, with accuracy near 99.2% and precision, recall, and F1-score all sitting close to 99%. The two-part ways once privacy and efficiency enter the picture, and the split traces back to how each model learns. VAE's probabilistic latent regularization keeps the learned distribution smooth and holds overfitting in check, whereas GAN's adversarial training pushes its samples toward sharp boundaries, which encourages memorization and drives false alarms up. The figures make this concrete: VAE's false alarm rate is 2.3%, clearly below GAN's 3.6% [24]. The privacy contrast is sharper still. Under the membership inference attack (MIA), VAE's AUC of 0.612 is close to random guessing, while GAN reaches 0.824, a sign of serious privacy leakage. GAN does earn a marginally better Jensen-Shannon divergence on fidelity, but that small edge is costly: VAE trains in 18 minutes on 4.2 GB of memory, while GAN needs 42 minutes and 8.1 GB [24]. Weighing classification, fidelity, privacy, and efficiency together, VAE holds the better Pareto balance, keeping detection performance competitive while protecting privacy and using only moderate resources, which makes it the more trustworthy and practical choice for encrypted traffic synthesis.

4.3. Model Selection for Federated Learning

For centralized edge IDS deployment, tree-based ensembles (RF, XGBoost) are the optimal choice for balanced accuracy, efficiency and interpretability, while CNN-LSTM suits high-complexity scenarios with sufficient edge computing resources. For federated IoT intrusion detection, RF and XGBoost remain preferred local classifiers, as their low computational overhead and robustness to non-IID data reduce convergence fluctuations during training. Lightweight CNN are an option where edge resources are ample. SVM and K-Means, by contrast, are poor candidates for federated deployment, since they scale badly and are highly sensitive to shifts in data distribution.

4.4. Challenges

The biggest difficulty running through this comparison is that the reviewed studies share no common evaluation benchmark. They differ in how they preprocess data, how they tune hyperparameters, and how finely they classify traffic, and these gaps make any direct cross-dataset comparison unfair and hard to defend. Some narrower IoT settings, such as compressed traffic detection at the edge perception layer, still have no widely accepted public dataset, and part of the work reported there was carried out on private testbeds, which leaves the results difficult to reproduce. Many studies also stop at accuracy and F1-score and leave out metrics that matter just as much for IoT deployment, such as the false alarm rate and inference latency. Together these shortcomings limit how far the conclusions can generalize, and they point to a clear need for a standardized evaluation framework in IoT intrusion detection.

5. Conclusion

This review sorted machine learning based IoT intrusion detection into a taxonomy organized by training architecture, covering centralized traditional machine learning and deep learning as well as distributed federated learning. The comparison makes one thing clear: which model works best depends heavily on the setting. Within traditional methods, tree-based ensembles such as RF and XGBoost, along with KNN, come out ahead of SVM, and that lead grows as datasets get larger and more imbalanced, whereas Isolation Forest holds up better than K-Means once the data is compressed at the edge. On the deep learning side, the hybrid CNN-LSTM gives the best overall detection, while VAE trades off accuracy, privacy, and efficiency more favorably than GAN. For federated deployment, RF and XGBoost stay the preferred local classifiers, since their low overhead and tolerance of non-IID data keep training stable. What holds back firmer conclusions is that the reviewed studies rarely share a common benchmark, which limits how far any single result can be generalized. Closing this gap will call for a standardized evaluation framework, one that reports deployment metrics such as false alarm rate and inference latency rather than accuracy alone. Shared public datasets for underexplored cases like edge compressed traffic would help as well, giving reproducibility a firmer footing. On the federated side, non-IID data, tight edge resources, communication overhead, and resilience to model-poisoning attacks all remain open problems worth pursuing.

References

[1] Sinha, S. (2026, April 29). Cellular IoT market update 2026: Connections grew 13% in 2025, slowest since 2020. IoT Analytics. https://iot-analytics.com/cellular-iot-market-update-spring-2026/.

[2] StormWall. (2025). The evolving DDoS threat: StormWall's H1 2025 report. https://stormwall.network/resources/blog/ddos-attack-statistics-h1-2025.

[3] Bitdefender. (2025). New Bitdefender and NETGEAR report reveals rising threats across the connected home landscape. https://www.bitdefender.com/en-us/news/new-bitdefender-and-netgear-report-reveals-rising-threats-across-the-connected-home-landscape.

[4] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1–6). Ottawa.

[5] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). Canberra.

[6] Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In 4th International Conference on Information Systems Security and Privacy (pp. 108–116). Funchal.

[7] Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in

the Internet of Things for network forensic analytics: Bot-IoT dataset. Future Generation Computer Systems, 100, 779–796.

[8] Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustainable Cities and Society, 72, 102994.

[9] Ullah, I., & Mahmoud, Q. H. (2020). A scheme for generating a dataset for anomalous activity detection in IoT networks. In Advances in Artificial Intelligence (pp. 508–520). Ottawa.

[10] Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281–40306.

[11] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18, 1153–1176.

[12] Abrar, I., Ayub, Z., Masoodi, F., & Bamhdi, A. M. (2020). A machine learning approach for intrusion detection system on NSL-KDD dataset. In 2020 International Conference on Smart Electronics and Communication (ICOSEC) (pp. 919–924). Trichy.

[13] Alosaimi, S., & Almutairi, S. M. (2023). An intrusion detection system using BoT-IoT. Applied Sciences, 13, 5427.

[14] Maseer, Z., Yusof, R., Bahaman, N., Mostafa, S. A., & Foozy, C. F. M. (2021). Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset. IEEE Access, 9, 22351–22370.

[15] Hamidou, S. T., & Mehdi, A. (2025). Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks. Machine Learning with Applications, 22, 100738.

[16] Doghramachi, D. F., & Ameen, S. Y. (2023). Internet of Things (IoT) security enhancement using XGboost machine learning techniques. Computers, Materials & Continua, 77, 717–732.

[17] Liu, D., Zhen, H., Kong, D., Chen, X., Zhang, L., Yuan, M., & Wang, H. (2021). Sensors anomaly detection of industrial Internet of Things based on isolated forest algorithm and data compression. Scientific Programming, 2021, 1–9.

[18] Ahmed, M., Seraj, R., & Islam, S. M. S. (2020). The k-means algorithm: A comprehensive survey and performance evaluation. Electronics, 9, 1295.

[19] Kareem, M. S., & Muhammed, L. A. (2024). Anomaly detection in streaming data using Isolation Forest. In 2024 Seventh International Women in Data Science Conference at Prince Sultan University (WiDS PSU) (pp. 223–228). Riyadh.

[20] Perumal, T., Mustapha, N., Mohamed, R., & Shiri, F. M. (2024). A comprehensive overview and comparative analysis on deep learning models. Journal on Artificial Intelligence, 6, 301–360.

[21] Neto, E. C. P., Iqbal, S., Buffett, S., Sultana, M., & Taylor, A. (2025). Deep learning for intrusion detection in emerging technologies: A comprehensive survey and new perspectives. Artificial Intelligence Review, 58, 340.

[22] Ab Razak, M. F., Firdaus, A., Tahaei, H., & Gheisari, M. (2026). A hybrid CNN–LSTM model for IoT intrusion detection: A robustness analysis across datasets. Future Internet, 18, 345.

[23] Bond-Taylor, S., Leach, A., Long, Y., & Willcocks, C. G. (2022). Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44, 7327–7347.

[24] Cherukuri, A. K. (2026). Comparative evaluation of VAE, GAN, and SMOTE for Tor detection in encrypted network traffic. arXiv preprint arXiv:2601.01183.

[25] Nguyen, C., Ding, M., Pathirana, N., Seneviratne, A., Li, J., & Vincent Poor, H. (2021). Federated learning for Internet of Things: A comprehensive survey. IEEE Communications Surveys & Tutorials, 23, 1622–1658.

[26] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In Artificial Intelligence and Statistics (pp. 1273–1282). Fort Lauderdale.

[27] Baich, M., & Sael, N. (2025). A federated learning-based intrusion detection system using dynamic ensemble aggregation for IoT networks. IEEE Access, 13, 205826–205839.