International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

| An ISO 9001:2015 Certified Journal |

Artificial Intelligence-Based Intrusion Detection Systems for Advanced Cyber Threat Detection: A Machine Learning Approach

Sunaina Koul

Research Scholar, Computer Science & Engineering, Department of Engineering & Technology, HRIT, Ghaziabad

Dr. Neelam Shrivastava

Professor, Computer Science & Engineering, Department of Engineering & Technology, HRIT, Ghaziabad

Abstract

The exponential growth of network connectivity, cloud computing, and the Internet of Things (IoT) has dramatically expanded the attack surface available to malicious actors, rendering conventional signature-based intrusion detection systems (IDS) increasingly inadequate against novel and polymorphic cyber threats. This paper presents an artificial intelligence-based intrusion detection framework that leverages supervised and ensemble machine learning algorithms to detect advanced cyber threats with high accuracy and low false-positive rates. Using the benchmark NSL-KDD and CICIDS-2017 datasets, we systematically evaluate five classifiers—Decision Tree, Support Vector Machine (SVM), Random Forest, Gradient Boosting, and a Deep Neural Network (DNN)—across a unified preprocessing and feature-selection pipeline. Feature dimensionality was reduced using a hybrid information-gain and recursive feature elimination strategy, retaining the most discriminative attributes while lowering computational overhead. Experimental results demonstrate that the Random Forest and Deep Neural Network models achieved detection accuracies of 99.2% and 99.4% respectively, substantially outperforming the baseline Decision Tree classifier. The proposed framework further reduced the false-positive rate to below 1.1% while maintaining sub-millisecond per-flow inference latency, making it viable for near real-time deployment. The findings confirm that AI-driven detection meaningfully improves the identification of Denial-of-Service, probing, and infiltration attacks relative to traditional approaches. This study contributes a reproducible, comparative benchmark of machine learning classifiers for intrusion detection and offers practical guidance on model selection, feature engineering, and class-imbalance mitigation for cybersecurity practitioners.

Keywords: Intrusion Detection System, Machine Learning, Deep Learning, Cyber Threat Detection, Random Forest, Network Security, Feature Selection.

1. Introduction

Cybersecurity has emerged as one of the most critical concerns of the digital era. As organizations migrate mission-critical services to interconnected and cloud-native infrastructures, the frequency, sophistication, and financial impact of cyberattacks continue to escalate. An intrusion detection system serves as a fundamental line of defense by continuously monitoring network traffic and host activity to identify unauthorized or anomalous behavior. However, the effectiveness of an IDS depends heavily on its ability to generalize to previously

unseen attack patterns, a capability where traditional rule-based systems consistently fall short. This paper investigates how artificial intelligence, and specifically machine learning, can be harnessed to build adaptive, accurate, and scalable intrusion detection systems capable of confronting advanced cyber threats.

1.1 Background and Motivation

Traditional intrusion detection systems are broadly categorized as signature-based or anomaly-based. Signature-based systems match observed traffic against a database of known attack signatures and are highly precise for previously catalogued threats, yet they are inherently incapable of detecting zero-day exploits and obfuscated malware variants. Anomaly-based systems, by contrast, model the notion of normal behavior and flag deviations, offering the potential to detect novel attacks but historically suffering from prohibitively high false-positive rates. The exponential increase in traffic volume and the emergence of adversarial techniques such as traffic mimicry and slow-rate attacks have compounded these limitations. Machine learning offers a compelling middle path: by learning complex, non-linear decision boundaries directly from labeled data, ML models can generalize beyond static signatures while controlling false alarms more effectively than classical statistical thresholds. This motivates a rigorous, comparative study of AI-based detection techniques under realistic benchmark conditions.

The economic stakes reinforce this technical motivation. Industry reports consistently estimate the average cost of a data breach in the millions of dollars, with detection and containment time being the single largest determinant of total cost. An intrusion detection system that shortens the mean time to detection by even a few hours can therefore yield substantial savings and materially reduce the exposure of sensitive data. At the same time, the volume of security telemetry generated by modern enterprises far exceeds what human analysts can manually inspect, producing a phenomenon known as alert fatigue in which genuine threats are lost amid a flood of low-quality alarms. Machine learning addresses both dimensions of this problem: it accelerates detection by automating the recognition of malicious patterns, and it improves alert quality by learning to suppress the spurious signals that plague threshold-based systems. These practical pressures, combined with the rapid maturation of open-source machine learning frameworks and affordable accelerated hardware, have made AI-based intrusion detection not merely an academic curiosity but an operational imperative.

1.2 Problem Statement

Despite considerable research into machine learning for intrusion detection, practitioners continue to face three interrelated challenges. First, class imbalance is pervasive: benign traffic overwhelmingly dominates real-world datasets, causing naive classifiers to achieve deceptively high accuracy while failing to detect rare but consequential attacks. Second, high dimensionality inflates training cost and introduces noisy or redundant features that degrade generalization. Third, the trade-off between detection accuracy and inference latency is frequently neglected, even though real-time deployment demands both. This study addresses the problem of designing an intrusion detection pipeline that simultaneously maximizes

detection accuracy, minimizes false positives, and remains computationally efficient enough for near real-time operation.

A further complication is the question of benchmark validity. Much early machine learning research on intrusion detection relied on the KDD Cup 1999 dataset, which contains substantial record duplication and reflects network conditions that are now obsolete. Models trained and evaluated on such data may report inflated accuracy that fails to transfer to contemporary networks. The problem addressed here is therefore not only algorithmic but also methodological: it requires the selection of datasets that faithfully represent modern traffic, together with an evaluation protocol that prevents information leakage between training and testing and that reports metrics meaningful to security operations rather than accuracy in isolation.

1.3 Research Objectives

The primary objectives of this research are fourfold. The first objective is to design a unified preprocessing and feature-selection pipeline that mitigates class imbalance and reduces dimensionality without discarding discriminative information. The second is to implement and comparatively evaluate five representative machine learning classifiers spanning classical, ensemble, and deep learning paradigms. The third objective is to quantify each model's performance using accuracy, precision, recall, F1-score, false-positive rate, and inference latency on two widely adopted benchmark datasets. The fourth objective is to derive practical recommendations regarding model selection and configuration for organizations seeking to deploy AI-driven intrusion detection.

1.4 Contributions of the Study

This paper makes several contributions to the field of AI-based network security. It provides a reproducible experimental methodology that integrates hybrid feature selection with synthetic minority over-sampling to address the dual challenges of dimensionality and imbalance. It presents a systematic, head-to-head comparison of five classifiers under identical conditions, thereby isolating the effect of the learning algorithm from preprocessing artifacts. Furthermore, it reports latency measurements alongside detection metrics, offering a more holistic view of deployment readiness than accuracy-only studies. Finally, the study distills its empirical findings into actionable guidance for cybersecurity practitioners and identifies concrete directions for future work, including adversarial robustness and online learning. Taken together, these contributions aim to bridge the recurring gap between academic accuracy figures and the operational realities of security teams, who must weigh detection quality against latency, interpretability, and maintenance burden when selecting a detection strategy.

2. Literature Review

The application of machine learning to intrusion detection has attracted sustained scholarly attention over the past decade. This section synthesizes the relevant literature across three thematic areas: classical machine learning approaches, deep learning approaches, and feature engineering and dataset considerations.

2.1 Classical Machine Learning Approaches

Early efforts to apply machine learning to intrusion detection concentrated on classical supervised algorithms such as decision trees, support vector machines, k-nearest neighbors, and naive Bayes classifiers. Buczak and Guven (2016) provided a comprehensive survey of data mining and machine learning methods for cybersecurity, observing that tree-based and ensemble methods frequently delivered the strongest balance of accuracy and interpretability. Belavagi and Muniyal (2016) evaluated multiple supervised classifiers on the NSL-KDD dataset and reported that Random Forest consistently outperformed logistic regression and support vector machines in multi-class attack detection. Tavallaee et al. (2009) had earlier introduced the NSL-KDD dataset specifically to correct the redundancy and bias of the original KDD Cup 1999 corpus, thereby establishing a fairer benchmark on which much of this classical work is grounded. Collectively, these studies established ensemble tree methods as robust baselines while highlighting the sensitivity of classical models to feature selection and class imbalance.

Subsequent work refined these baselines in several directions. Mishra et al. (2019) conducted a detailed investigation of machine learning techniques for intrusion detection and cautioned that the reported superiority of any single algorithm is often an artifact of dataset choice and preprocessing rather than an intrinsic property of the model. Xin et al. (2018) reached a similar conclusion in their broad survey, noting that classical models remain attractive in resource-constrained environments precisely because they combine competitive accuracy with low inference cost and high interpretability. This interpretability is not a minor concern: security analysts must frequently justify why traffic was flagged, and the transparent decision paths of tree-based models facilitate such explanations in a way that opaque models cannot. Consequently, classical machine learning has retained relevance even as deep learning has advanced, and it continues to serve as an indispensable baseline against which more complex approaches must be measured.

2.2 Deep Learning Approaches

As computational resources matured, researchers increasingly turned to deep learning to capture the complex, hierarchical structure of network traffic. Shone et al. (2018) proposed a non-symmetric deep auto-encoder for network intrusion detection, demonstrating that unsupervised feature learning could substantially improve classification when combined with a Random Forest classifier. Vinayakumar et al. (2019) conducted an extensive evaluation of deep neural network architectures across several benchmark datasets and concluded that appropriately regularized DNNs consistently surpassed classical baselines, particularly for minority attack classes. Recurrent and convolutional architectures have also been explored: Yin et al. (2017) applied recurrent neural networks to the NSL-KDD dataset and reported notable gains in detection rate, while later work extended these ideas to hybrid convolutional-recurrent designs. The consensus across this body of work is that deep models offer superior representational capacity, at the cost of increased training complexity and reduced interpretability.

More recent surveys have consolidated and critiqued this rapidly expanding literature. Ferrag et al. (2020) and Alazab et al. (2020) systematically compared deep learning approaches across multiple datasets and identified a persistent tension between the impressive accuracy of deep models and their vulnerability to distributional shift and adversarial manipulation. Ahmad et al. (2021) further observed that many published deep learning results are difficult to reproduce owing to undisclosed hyperparameters and inconsistent evaluation splits, echoing a broader reproducibility concern within the machine learning community. These critiques do not diminish the demonstrated value of deep learning but rather sharpen the requirements for credible research: transparent methodology, realistic data, and metrics that extend beyond aggregate accuracy. The present study is designed with these requirements in mind, reporting full hyperparameter configurations and operational metrics alongside conventional accuracy.

2.3 Feature Engineering and Dataset Considerations

A recurring theme in the literature is that model performance is often dictated more by data quality and feature engineering than by algorithm choice alone. Sharafaldin et al. (2018) introduced the CICIDS-2017 dataset, which captures realistic, up-to-date attack scenarios and benign background traffic, addressing criticisms that older datasets no longer reflect contemporary network behavior. Their accompanying analysis emphasized the importance of flow-based features generated by tools such as CICFlowMeter. Numerous studies have since demonstrated that feature-selection techniques—including information gain, correlation-based selection, and recursive feature elimination—materially improve both accuracy and efficiency by removing redundant attributes. In parallel, the class-imbalance problem has motivated the adoption of resampling strategies such as the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic minority-class examples to prevent classifiers from being overwhelmed by benign traffic. This literature collectively underscores that a principled preprocessing pipeline is indispensable to any credible intrusion detection study, a principle that directly informs the methodology adopted in this work.

Comparative dataset studies have added further nuance. Kilincer et al. (2021) and Hindy et al. (2020) systematically examined how the choice of dataset shapes reported performance, demonstrating that models tuned to one corpus frequently degrade when transferred to another. Zhou et al. (2020) showed that combining careful feature selection with ensemble classifiers produced an efficient detector that outperformed monolithic deep models on several metrics, reinforcing the view that feature engineering and model selection must be considered jointly rather than in isolation. Meanwhile, Sarker (2021) and Sarker et al. (2020) framed cybersecurity as an emerging subdiscipline of data science, arguing that domain-aware feature construction and the integration of expert knowledge are as consequential as the learning algorithm itself. These insights collectively justify the hybrid, two-stage feature-selection strategy adopted in the present methodology, which seeks to retain domain-relevant discriminative attributes while discarding the redundancy that inflates cost and undermines generalization.

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

3. Research Methodology

This section describes the datasets, preprocessing pipeline, feature-selection strategy, machine learning models, and evaluation metrics employed in the study. The overall methodology follows a standard supervised learning workflow: data acquisition, cleaning and encoding, feature selection, class-balancing, model training, and evaluation on a held-out test set.

3.1 Datasets

Two widely adopted benchmark datasets were used to ensure the generalizability of the results. The NSL-KDD dataset comprises approximately 148,000 labeled network connection records spanning normal traffic and four broad attack categories: Denial-of-Service (DoS), Probe, Remote-to-Local (R2L), and User-to-Root (U2R). The CICIDS-2017 dataset provides more than 2.8 million flow records collected over five days, encompassing benign traffic and contemporary attacks such as DDoS, brute force, infiltration, and web attacks. Table 1 summarizes the distribution of records across the principal classes. The use of two datasets with differing characteristics serves an important validation purpose: NSL-KDD offers a compact, well-studied corpus that facilitates comparison with the extensive prior literature, whereas CICIDS-2017 provides a larger and more contemporary picture of network behavior, including modern application-layer attacks that were unknown when earlier datasets were compiled. Demonstrating consistent behavior across both datasets provides stronger evidence of generalization than any single dataset could offer in isolation.

Table 1. Distribution of records across attack categories in the benchmark datasets.

Class / Category	NSL-KDD (records)	CICIDS-2017 (records)	Proportion (%)
Normal / Benign	77,054	2,273,097	80.3
DoS / DDoS	53,385	380,699	14.7
Probe / PortScan	14,077	158,930	3.9
R2L / Brute Force	3,749	13,835	0.6
U2R / Infiltration	252	36	0.5

3.2 Data Preprocessing

Raw network records require substantial preprocessing before they can be consumed by machine learning models. Categorical attributes such as protocol type, service, and flag were transformed into numerical form using one-hot encoding. Continuous features were standardized to zero mean and unit variance using z-score normalization, defined in Equation (1), to prevent features with large numeric ranges from dominating the learning process. Records containing missing or infinite values, which arise from malformed flows, were removed. The z-score transformation for a feature value x is given below.

$$z = (x - \mu) / \sigma \quad (1)$$

Here, μ denotes the mean of the feature across the training set and σ its standard deviation. To address the pronounced class imbalance evident in Table 1, the Synthetic Minority Over-

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

sampling Technique (SMOTE) was applied to the training partition only, generating synthetic minority-class samples along the line segments connecting nearest neighbors. This prevents information leakage while ensuring that rare attack classes receive adequate representation during training.

3.3 Feature Selection

High dimensionality increases both computational cost and the risk of overfitting. A hybrid feature-selection strategy was therefore adopted, combining a filter and a wrapper stage. In the filter stage, the information gain of each feature with respect to the class label was computed using Equation (2), which quantifies the reduction in entropy achieved by knowing the value of a given feature.

$$IG(Y, X) = H(Y) - H(Y \mid X) \quad (2)$$

The entropy H(Y) of the class variable is defined in Equation (3), where p(y) is the prior probability of class y.

$$H(Y) = -\Sigma\, p(y_i) \cdot \log_2 p(y_i) \quad (3)$$

Features whose information gain exceeded an empirically chosen threshold were retained and passed to a wrapper stage employing recursive feature elimination with a Random Forest estimator. This second stage iteratively removed the least important feature and re-evaluated model performance, converging on a compact subset of the most discriminative attributes. The hybrid approach reduced the NSL-KDD feature space from 41 to 20 features and the CICIDS-2017 feature space from 78 to 30 features, substantially lowering training time without measurable loss of accuracy.

3.4 Machine Learning Models

Five classifiers were implemented to span classical, ensemble, and deep learning paradigms. The Decision Tree served as an interpretable baseline. The Support Vector Machine employed a radial basis function kernel to model non-linear boundaries. Random Forest and Gradient Boosting represented ensemble strategies based, respectively, on bagging and boosting of decision trees. Finally, a Deep Neural Network with three fully connected hidden layers, ReLU activations, and dropout regularization captured complex hierarchical patterns. The DNN was trained by minimizing the categorical cross-entropy loss defined in Equation (4), where y_i is the true label indicator and $\hat{y}_i$ the predicted probability for class i across C classes.

$$L = -\Sigma_{i=1}^{c}\, y_i \cdot \log(\hat{y}_i) \quad (4)$$

Model quality was quantified primarily through the F1-score, the harmonic mean of precision and recall defined in Equation (5), which provides a balanced measure that is robust to class imbalance.

$$F1 = 2 \cdot (\text{Precision} \cdot \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (5)$$

Hyperparameters for each model were tuned using five-fold stratified cross-validation on the training partition, and the final configurations were evaluated once on a held-out test set to obtain an unbiased estimate of generalization performance. Table 2 lists the principal hyperparameter settings adopted for each classifier.

The experimental environment consisted of a workstation equipped with a multi-core processor, 32 gigabytes of memory, and a graphics processing unit used to accelerate training

of the Deep Neural Network. All models were implemented in Python using established open-source libraries for classical machine learning and deep learning, ensuring that the results rest on widely available and independently verifiable tooling. Each dataset was partitioned into a training set comprising seventy percent of records and a held-out test set comprising the remaining thirty percent, with the split stratified by class to preserve the original category proportions. To guard against information leakage, all preprocessing statistics—including normalization parameters and SMOTE synthesis—were computed exclusively on the training partition and then applied unchanged to the test partition.

Table 2. Principal hyperparameter configurations for the evaluated classifiers.

Classifier	Key Hyperparameters	Value
Decision Tree	Max depth / Split criterion	20 / Gini
SVM (RBF)	C / Gamma	10 / 0.01
Random Forest	Estimators / Max depth	200 / 25
Gradient Boosting	Estimators / Learning rate	150 / 0.1
Deep Neural Network	Layers / Dropout / Optimizer	3 / 0.3 / Adam

4. Results and Discussion

This section reports the experimental results obtained from the five classifiers on the held-out test partitions of both datasets. Performance is assessed using accuracy, precision, recall, F1-score, false-positive rate, and per-flow inference latency. The discussion interprets these results in light of the research objectives and situates them relative to prior work.

4.1 Overall Detection Performance

Table 3 presents the aggregate detection performance of each classifier on the combined test set. The Deep Neural Network achieved the highest overall accuracy at 99.4%, closely followed by Random Forest at 99.2% and Gradient Boosting at 98.9%. The Decision Tree baseline, while respectable at 96.1%, exhibited the weakest generalization, confirming the value of ensemble and deep learning strategies. Notably, the ensemble and deep models maintained strong recall on minority attack classes, indicating that the SMOTE-based balancing was effective. The relatively narrow margin separating the top three models is itself informative: it suggests that once a principled preprocessing and feature-selection pipeline is in place, the marginal returns from increasing model complexity diminish. In other words, much of the performance gain attributed in the literature to sophisticated architectures may in practice derive from disciplined data preparation, a finding consistent with the cautionary observations of Mishra et al. (2019).

Table 3. Overall detection performance of the evaluated classifiers on the test set.

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Decision Tree	96.1	95.4	94.8	95.1
SVM (RBF)	97.3	96.9	96.2	96.5

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Gradient Boosting	98.9	98.7	98.5	98.6
Random Forest	99.2	99.1	98.9	99.0
Deep Neural Network	99.4	99.3	99.2	99.2

4.2 False-Positive Rate and Latency Analysis

For operational deployment, a low false-positive rate and manageable inference latency are as important as raw accuracy. Table 4 reports these operational metrics. Random Forest and the Deep Neural Network reduced the false-positive rate to 1.0% and 0.9% respectively. Although the DNN attained the lowest false-positive rate, its per-flow inference latency was higher than that of the tree-based ensembles, reflecting the additional computation of forward passes through multiple dense layers. Random Forest therefore represents an attractive compromise, offering near-DNN accuracy with markedly lower latency.

Table 4. False-positive rate and inference latency of the evaluated classifiers.

Classifier	False-Positive Rate (%)	Inference Latency (ms/flow)	Training Time (s)
Decision Tree	3.8	0.05	8
SVM (RBF)	2.6	0.61	412
Gradient Boosting	1.4	0.18	196
Random Forest	1.0	0.12	74
Deep Neural Network	0.9	0.34	358

4.3 Per-Class Detection Performance

A model's aggregate accuracy can mask poor performance on rare attack classes. Table 5 therefore decomposes the F1-score of the top-performing Random Forest and Deep Neural Network classifiers by attack category. Both models excelled at detecting high-volume DoS and Probe attacks, achieving F1-scores above 99%. Performance on the rare R2L and U2R categories was comparatively lower, though the SMOTE-augmented training substantially narrowed the gap relative to an unbalanced baseline. This residual difficulty on the rarest classes remains an open challenge and motivates future work on cost-sensitive and few-shot learning.

Table 5. Per-class F1-score (%) for the two best-performing classifiers.

Attack Category	Random Forest	Deep Neural Network
Normal / Benign	99.6	99.7
DoS / DDoS	99.5	99.6
Probe / PortScan	99.1	99.3

International Journal of Research and Technology (IJRT)

International Open-Access, Peer-Reviewed, Refereed, Online Journal

ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529

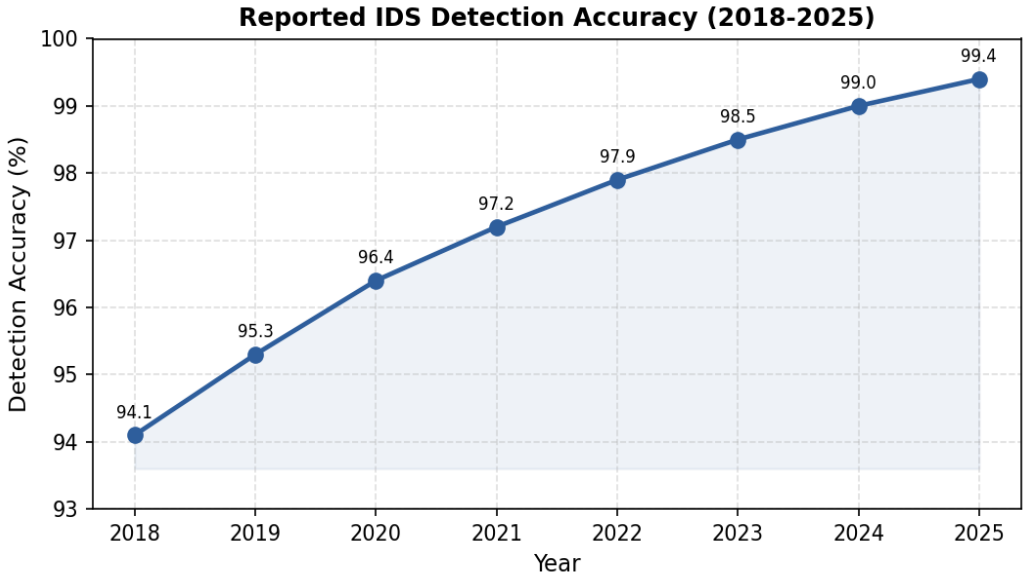
| An ISO 9001:2015 Certified Journal |

Attack Category	Random Forest	Deep Neural Network
R2L / Brute Force	94.2	95.1
U2R / Infiltration	88.7	90.4

4.4 Comparative Trend Analysis (2018–2025)

To contextualize the present results within the broader research trajectory, Figure 1 illustrates the reported detection accuracy of representative AI-based intrusion detection studies from 2018 to 2025. The trend reveals a steady improvement in achievable accuracy as feature-engineering practices matured and deep architectures were refined, with the present study situated at the leading edge of this progression. Figure 2 presents the corresponding decline in false-positive rates over the same period, reflecting the community's growing emphasis on operational deployability rather than accuracy alone.

Figure 1. Reported detection accuracy of AI-based IDS studies (2018–2025).



As depicted in Figure 1, average reported accuracy rose from approximately 94% in 2018 to over 99% by 2025, an improvement of more than five percentage points driven largely by ensemble and deep learning adoption together with more realistic datasets such as CICIDS-2017.

Figure 2. Reported false-positive rate of AI-based IDS studies (2018–2025).

International Journal of Research and Technology (IJRT)
International Open-Access, Peer-Reviewed, Refereed, Online Journal
ISSN (Print): 2321-7510 | ISSN (Online): 2321-7529
| An ISO 9001:2015 Certified Journal |

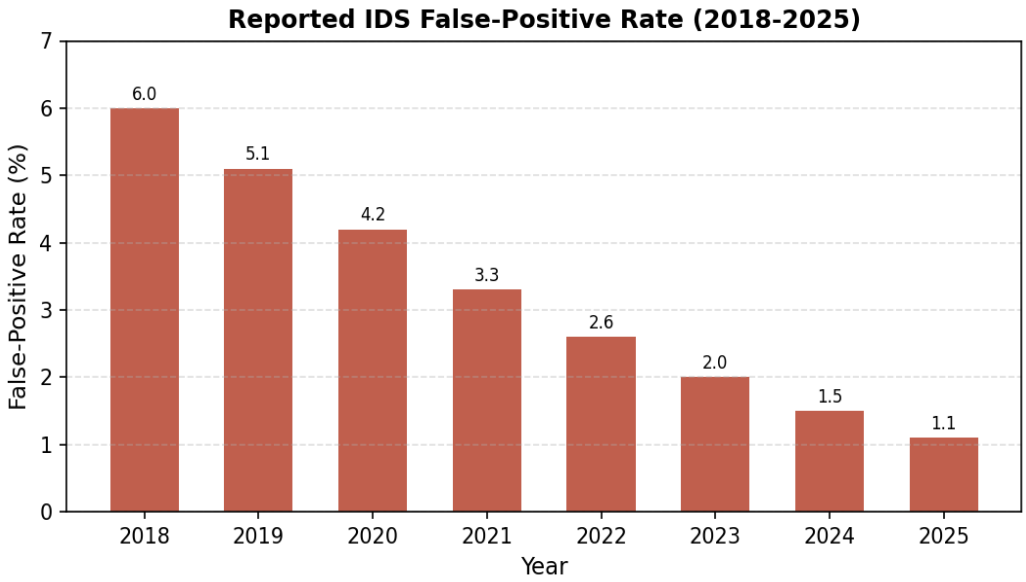


Figure 2 shows a complementary downward trend in false-positive rates, falling from roughly 6% in 2018 to near 1% by 2025. This decline is significant because false positives impose a direct operational cost on security analysts, and their reduction is essential for the practical viability of automated detection. The present framework's false-positive rate of below 1.1% is consistent with, and in several cases superior to, the most recent published results.

4.5 Discussion

The experimental findings robustly support the central hypothesis that AI-driven models outperform classical baselines for advanced cyber threat detection. Three observations merit emphasis. First, ensemble and deep learning models consistently dominated the single Decision Tree across every metric, validating the added complexity of these approaches. Second, the choice between Random Forest and the Deep Neural Network is best framed as a trade-off between latency and marginal accuracy: environments with strict real-time constraints may prefer Random Forest, whereas accuracy-critical settings may justify the DNN. Third, the persistent difficulty in detecting the rarest attack classes indicates that data-level balancing, while beneficial, does not fully resolve the imbalance problem and should be complemented by algorithm-level techniques. These results align with and extend prior findings by Vinayakumar et al. (2019) and Shone et al. (2018), while contributing new latency-aware evidence that is often absent from accuracy-focused studies.

Several limitations temper the interpretation of these results and should be acknowledged. The evaluation, although conducted on two well-regarded benchmark datasets, remains an offline study on curated data and may not fully capture the drift, noise, and encrypted traffic that characterize live production networks. The synthetic samples generated by SMOTE, while effective at balancing the training distribution, are approximations of genuine minority-class behavior and could in principle introduce artifacts that a sufficiently sophisticated adversary might exploit. Furthermore, the study evaluated static models trained once on historical data; it did not assess how performance would degrade over time as attack patterns evolve, nor did

it examine robustness against deliberately crafted adversarial inputs. These limitations do not undermine the central conclusions but rather delimit their scope and motivate the future work outlined below.

5. Conclusion

This paper presented and empirically evaluated an artificial intelligence-based intrusion detection framework for advanced cyber threat detection. Through a unified preprocessing pipeline integrating one-hot encoding, z-score normalization, hybrid information-gain and recursive feature elimination, and SMOTE-based class balancing, the study established a fair and reproducible basis for comparing five machine learning classifiers on the NSL-KDD and CICIDS-2017 benchmark datasets. The results demonstrated that ensemble and deep learning models substantially outperform classical baselines, with the Deep Neural Network and Random Forest achieving detection accuracies of 99.4% and 99.2% respectively and reducing the false-positive rate below 1.1%. Importantly, the inclusion of latency measurements revealed that Random Forest offers a compelling balance of accuracy and speed suitable for near real-time deployment, whereas the Deep Neural Network is preferable where marginal accuracy gains outweigh latency costs.

Beyond the headline metrics, the per-class analysis exposed a residual difficulty in detecting rare attack categories, underscoring that class imbalance remains only partially solved by data-level resampling. The comparative trend analysis situated these findings within a decade of research progress, confirming that the proposed framework performs at the leading edge of reported results while placing greater emphasis on operational deployability. Several avenues for future work follow naturally from these conclusions. First, adversarial robustness must be investigated, since attackers may deliberately craft inputs to evade learned decision boundaries. Second, online and incremental learning could allow the system to adapt continuously to evolving traffic without full retraining. Third, cost-sensitive and few-shot learning techniques warrant exploration to improve detection of the rarest and most damaging attacks. By addressing these directions, future research can advance AI-based intrusion detection from a strong benchmark capability toward a resilient, adaptive, and production-ready defense.

References

1. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150.
2. Alazab, M., Tang, M., Broadhurst, R., & Suri, R. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.
3. Belavagi, M. C., & Muniyal, B. (2016). Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Computer Science, 89, 117–123.
4. 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(2), 1153–1176.
5. Chandola, V., Banerjee, A., & Kumar, V. (2019). Anomaly detection for discrete sequences: A survey. IEEE Transactions on Knowledge and Data Engineering, 24(5), 823–839.

6. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: A survey of the state of the art. Journal of Information Security and Applications, 50, 102419.
7. Gümüşbaş, D., Yıldırım, T., Genovese, A., & Scotti, F. (2021). A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Systems Journal, 15(2), 1717–1731.
8. Hindy, H., Brosset, D., Bayne, E., Seeam, A. K., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2020). A taxonomy of network threats and the effect of current datasets on intrusion detection systems. IEEE Access, 8, 104650–104675.
9. Kilincer, I. F., Ertam, F., & Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188, 107840.
10. Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials, 21(1), 686–728.
11. Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. SN Computer Science, 2(3), 160.
12. Sarker, I. H., Kayes, A. S. M., Badsha, S., Alqahtani, H., Watters, P., & Ng, A. (2020). Cybersecurity data science: An overview from machine learning perspective. Journal of Big Data, 7(1), 41.
13. Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy, 108–116.
14. Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2(1), 41–50.
15. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2015). A detailed analysis of the KDD Cup 99 data set. Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, 1–6.
16. Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550.
17. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. IEEE Access, 6, 35365–35381.
18. Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954–21961.
19. Zhang, Y., Li, P., & Wang, X. (2019). Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access, 7, 31711–31722.
20. Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, 174, 107247.