International Journal of Current Engineering and Technology
E-ISSN 2277-4106, P-ISSN 2347-5161

Available at https://ijcet.evegenis.org/index.php/ijcet/index

Review Article

Clustering and Classification Techniques in Machine Learning-based Intrusion Detection Systems: A Comprehensive Review

Amrendra Kumar Sharma[1]* and Mamta Tiwari[2]

[12]Department of Computer Application, Chhatrapati Shahu Ji Maharaj University, Kanpur, India

Received 01 Mar 2026, Accepted 26 Mar 2026, Available online 27 Mar 2026, **Vol.16, No.2 (Mar/Apr 2026)**

Abstract

The rapid proliferation of networked digital systems has substantially expanded the threat landscape for cyber intrusions, necessitating robust and adaptive security mechanisms. Intrusion Detection Systems (IDS) constitute a critical component of network security infrastructure by identifying unauthorized access attempts and malicious behavior in real time. Traditional signature-based detection techniques, while effective against known threats, exhibit notable limitations in identifying novel and polymorphic attack patterns. This limitation has driven extensive research into machine learning (ML)-based IDS frameworks that leverage statistical and structural properties of network traffic for automated threat detection. Among ML paradigms, clustering and classification methods have emerged as particularly effective strategies. Classification algorithms employ supervised learning to discriminate between normal and attack traffic based on labeled training data, while clustering techniques utilize unsupervised learning to discover anomalous groupings in unlabeled network data. This paper presents a systematic and comprehensive review of clustering and classification techniques applied in ML-based IDS, examining foundational algorithms including Logistic Regression, K-Nearest Neighbor, Decision Tree, Random Forest, Support Vector Machine, K-Means, and Hierarchical Clustering. The study further analyzes widely adopted benchmark datasets, evaluates performance metrics, and discusses current research challenges. Identified gaps include the handling of high-dimensional feature spaces, computational constraints in IoT environments, and the development of lightweight hybrid detection models. Directions for future research are outlined to guide the design of more intelligent, scalable, and efficient intrusion detection frameworks.

Keywords: *Intrusion Detection System (IDS), Machine Learning (ML), Clustering, Classification, Anomaly Detection, Network Security*

1. Introduction

The transformative expansion of digital communication infrastructure over the past two decades has fundamentally altered the nature of cybersecurity threats. Modern organizations, governmental bodies, and individuals increasingly depend on interconnected systems and internet-based services for critical operations ranging from financial transactions and healthcare management to national infrastructure control. While these technological advancements have fostered unprecedented levels of efficiency and accessibility, they have simultaneously introduced complex and evolving security vulnerabilities that are continuously exploited by malicious actors.

Cybersecurity threats manifest in diverse forms, including unauthorized access attempts, denial-of-service (DoS) attacks, data exfiltration, malware injection, and advanced persistent threats (APTs). The economic and operational consequences of successful intrusions can be severe, ranging from data breaches and financial losses to reputational damage and disruption of essential services. Consequently, the development and deployment of effective security mechanisms to detect and mitigate intrusion attempts in real time has become a critical research priority.

An Intrusion Detection System (IDS) is a security mechanism that monitors network traffic and system activities to identify patterns indicative of unauthorized access or malicious behavior [1]. IDS solutions are broadly categorized into two types based on their detection methodology: (1) signature-based detection, which relies on a database of known attack patterns to identify threats, and (2) anomaly-based

*Corresponding author's ORCID ID: 0000-0000-0000-0000
DOI: https://doi.org/10.14741/ijcet/v.16.2.3

detection, which establishes a baseline model of normal behavior and flags deviations from this baseline as potential intrusions [2]. While signature-based approaches demonstrate high precision for known attacks, their inability to detect previously unseen or zero-day threats represents a significant limitation in dynamic threat environments.

The emergence of machine learning (ML) as a transformative paradigm in data analysis has catalyzed substantial advances in IDS research. ML-based IDS frameworks leverage statistical inference and pattern recognition to model complex relationships within network traffic data, enabling the detection of novel attack patterns that evade traditional signature-based systems. Among the various ML techniques applied in IDS, clustering and classification algorithms have garnered particular attention due to their complementary strengths in handling both labeled and unlabeled network traffic data.

Classification algorithms, which operate within the supervised learning paradigm, require labeled datasets to train predictive models that can distinguish between normal and malicious network activities. In contrast, clustering algorithms employ unsupervised learning to identify inherent groupings within unlabeled data, making them particularly suitable for anomaly detection in environments where labeled data is scarce or expensive to obtain. The synergistic application of these two algorithmic families has yielded promising results in terms of detection accuracy, false positive rates, and computational efficiency.

This paper presents a comprehensive review of clustering and classification techniques used in ML-based IDS. The study systematically examines foundational algorithms, benchmark datasets, evaluation metrics, and the comparative performance of various approaches. Furthermore, key research challenges are identified, and potential directions for future investigation are outlined. The remainder of this paper is organized as follows: Section 2 describes commonly used ML-based detection techniques; Section 3 presents a comparative analysis of related work; Section 4 discusses performance evaluation metrics; Section 5 examines benchmark datasets; Section 6 outlines challenges and future directions; and Section 7 concludes the paper.

2. Machine Learning-based Detection in IDS

Machine learning-based IDS approaches can be broadly categorized according to the type of learning mechanism employed. In supervised learning, a model is trained on a labeled dataset containing both normal and attack instances, and subsequently used to classify new, unseen network records. In contrast, unsupervised learning algorithms identify patterns and structures in data without relying on pre-labeled examples, making them particularly valuable when ground-truth annotations are unavailable or incomplete. Figure 1 illustrates the general architecture of an ML-based IDS, highlighting the roles of both classification and clustering components.

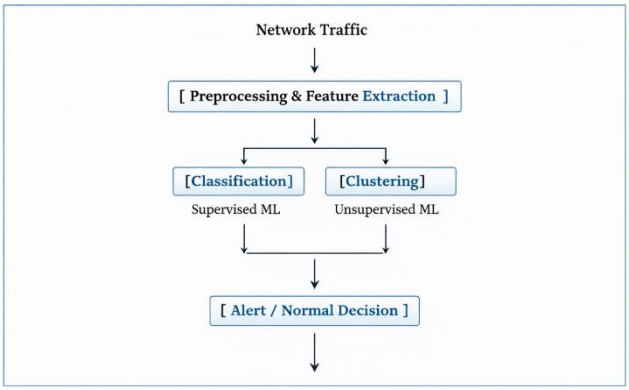

Figure 1. General architecture of an ML-based Intrusion Detection System

2.1 Classification Algorithms

Classification algorithms form the backbone of supervised ML-based IDS. These algorithms operate in two distinct phases: a training phase, in which the model learns discriminative patterns from labeled data, and a testing or inference phase, in which the trained model classifies new data instances as either normal or intrusive. The following subsections describe the most commonly employed classification algorithms in IDS research.

2.1.1 Logistic Regression (LR)

Logistic Regression is a foundational supervised classification algorithm that models the probability of a binary outcome as a function of one or more input features. In the context of IDS, it is used to estimate the probability that a network event belongs to the intrusion class. The model employs a sigmoid activation function to map linear combinations of input features to probability values in the range [0, 1], expressed as $F(x) = 1 / (1 + e^{-x})$. Despite its relative simplicity, Logistic Regression offers several advantages including interpretability, computational efficiency, and suitability for linearly separable datasets. However, its performance may be limited in scenarios involving complex, non-linear decision boundaries characteristic of multi-class attack classification.

2.1.2 K-Nearest Neighbor (KNN)

K-Nearest Neighbor is a non-parametric, instance-based learning algorithm that classifies a new data point based on the majority class among its k nearest neighbors in the feature space. Proximity is typically measured using the Euclidean distance metric, though other distance measures such as Manhattan or Minkowski distances may also be employed. KNN is inherently adaptive and requires no explicit training

phase, as classification is performed directly on the stored training data. In IDS applications, KNN has demonstrated competitive classification performance; however, its computational cost scales linearly with the size of the training dataset, posing scalability challenges for high-throughput network environments.

2.1.3 Decision Tree (DT)

Decision Tree is a hierarchical, tree-structured classification model that recursively partitions the feature space based on attribute selection criteria such as information gain or Gini impurity. Each internal node of the tree represents a decision rule applied to a specific feature, each branch corresponds to an outcome of that rule, and each leaf node represents a class label. The recursive binary splitting procedure continues until a stopping criterion is satisfied, resulting in a model that is both interpretable and computationally efficient at inference time. In IDS applications, Decision Trees are valued for their transparency and ability to handle heterogeneous feature types; however, they are susceptible to overfitting when the tree depth is not appropriately constrained [3, 4].

2.1.4 Random Forest (RF)

Random Forest is an ensemble learning method that constructs a collection of decision trees during the training phase, each trained on a randomly sampled subset of the training data using a randomly selected subset of features. Classification decisions are determined by aggregating the outputs of individual trees through majority voting, which reduces variance and mitigates the overfitting tendency inherent in single decision trees [5]. Random Forest has consistently demonstrated state-of-the-art performance in IDS benchmarks, offering robust detection accuracy, resistance to noise, and the ability to handle high-dimensional feature spaces with minimal hyperparameter tuning.

2.1.5 Support Vector Machine (SVM)

Support Vector Machine is a margin-based classifier that seeks to identify an optimal separating hyperplane in the feature space that maximizes the margin between the two classes while minimizing classification errors [6]. For non-linearly separable data, kernel functions such as the Radial Basis Function (RBF), polynomial, and linear kernels are employed to implicitly map the input data into a higher-dimensional feature space where linear separation becomes feasible. SVM has been widely applied in IDS due to its strong theoretical foundations, generalization capability, and effectiveness in high-dimensional settings; however, its computational complexity during training scales poorly with large datasets.

2.1.6 Ensemble Methods

Ensemble methods represent a class of ML strategies that combine the predictions of multiple base learners to achieve superior predictive performance compared to any individual model. The fundamental premise is that an ensemble of weak learners, when appropriately combined, can form a strong and robust classifier. Principal ensemble strategies include bagging, which trains multiple models on bootstrapped samples of the training data; boosting, which sequentially trains models with increasing emphasis on previously misclassified instances; and stacking, which trains a meta-learner to combine the outputs of multiple heterogeneous base models [7]. In IDS research, ensemble methods have consistently achieved among the highest reported detection accuracies while maintaining competitive false positive rates.

2.2 Clustering Algorithms

Clustering algorithms are unsupervised ML techniques that partition a dataset into groups, or clusters, such that instances within the same cluster exhibit greater similarity to one another than to instances in different clusters. Unlike classification algorithms, clustering methods do not require labeled training data, making them particularly suitable for anomaly-based IDS scenarios where complete and accurate labeling of network traffic is impractical.

2.2.1 K-Means Clustering

K-Means is one of the most widely employed clustering algorithms in ML-based IDS research. The algorithm partitions a dataset into a predefined number k of clusters by iteratively assigning each data point to the cluster whose centroid is nearest according to a distance metric, typically Euclidean distance, and subsequently updating the centroid of each cluster as the mean of its assigned points. The iterative process continues until the cluster assignments stabilize or a convergence criterion is met. In IDS applications, K-Means can identify clusters corresponding to different traffic types, with outlier clusters potentially representing attack traffic. A key limitation of K-Means is its sensitivity to the initial placement of centroids and the requirement to specify k in advance.

2.2.2 Hierarchical Clustering

Hierarchical clustering constructs a multi-level tree structure, or dendrogram, that represents the nested relationships among data points. Two primary variants exist: agglomerative hierarchical clustering, which follows a bottom-up approach by initially treating each data point as an individual cluster and progressively merging the most similar clusters; and divisive hierarchical clustering, which follows a top-down approach by starting with all data points in a single

cluster and recursively splitting it into smaller groups. The dendrogram produced by hierarchical clustering provides a rich visualization of the data's structural organization and allows flexible determination of the final number of clusters by cutting the dendrogram at an appropriate level. In IDS research, hierarchical clustering has been applied to analyze the relationships between different attack categories and normal traffic patterns.

3. Comparative Analysis of Related Work

A substantial body of research has investigated the application of both classification and clustering ML techniques in IDS. Table 1 presents a systematic summary of key contributions in this domain, highlighting the ML approaches employed, the datasets used, and the primary findings reported.

Table 1. Summary of related work on ML-based IDS

Ref.	ML Approach	Description	Dataset	Key Finding
[8]	Ensemble (AdaBoost)	Modest, Real & Gentle AdaBoost in binary classification	KDD Cup, UNSW-NB15, NSLKDD, CICIDS2017	FPR of 0.02%; higher processing time
[9]	DT, SVM, RF	Recursive feature elimination to identify attacks	KDD Cup	98.5% accuracy across attack classes
[10]	SVM	Genetic Algorithm for best attribute selection	CICIDS2017, ADFA-LDWMN	99.8% accuracy; low computational overhead
[11]	Ensemble	PSO, ABC, GA for optimized feature selection	NSLKDD, UNSW-NB15	91.2% binary classification accuracy
[12]	KNN, SVM, Cluster	Distance-based clusters transformed to 1D	KDD Cup	99.76% accuracy, outperforms KNN & SVM
[13]	Ensemble, DT	GA for feature selection	NSLKDD	Reduced false alarm rate; longer build time
[14]	Stacking (KNN, LR, RF, SVM)	Info-gain & hashing for feature extraction	UNSW-NB15, UGR16	94% and 98.17% accuracies
[15]	KNN, SVM	Hybrid chi-square + frequency episode extraction	KDD Cup	True positive rate of 92.65%
[7]	K-Means, Fuzzy C-Means	Entropy estimation across 6 classifiers	Kyoto2006+	83.6% clustering accuracy
[16]	KNN, Ensemble	SOM for dimensionality reduction	NSLKDD	Precision, recall, F1 improved by 10–30%
[17]	Fuzzy Cluster, ANN	Clusters formed; ANN trained on subsets	KDD Cup	96.7% average accuracy
[18]	Naïve Bayes, K-Medoids	Variable-size clusters in IDS	KDD Cup	Better DoS/Probe/R2L/U2R than K-Means
[19]	K-Means, Naïve Bayes	Hybrid anomaly IDS	KDD Cup	99% accuracy; 98.8% detection rate
[20]	K-Means, C4.5	Cascaded hybrid anomaly IDS	KDD Cup	99.6% true positive rate

An analysis of the related work reveals several notable trends. Ensemble methods, particularly stacking and boosting variants, consistently achieve the highest classification accuracies, often exceeding 98% on standard benchmark datasets. This performance advantage stems from the ability of ensemble approaches to reduce both bias and variance by aggregating the predictions of multiple complementary base learners. Among individual classifiers, Random Forest and SVM demonstrate the most robust performance, while simpler models such as Logistic Regression and KNN exhibit competitive accuracy on less complex datasets.

The integration of feature selection and dimensionality reduction techniques, including Genetic Algorithms, Particle Swarm Optimization, and Self-Organizing Maps, has been shown to significantly improve classification performance by removing redundant and irrelevant features. This is particularly important given the high dimensionality of modern network traffic datasets, which may contain dozens to hundreds of features per flow record.

Hybrid approaches that combine clustering and classification in a pipeline architecture have demonstrated notable promise. In such frameworks, clustering is first employed to partition the dataset into homogeneous subsets, and classification models are subsequently trained on each subset. This strategy can improve detection accuracy by tailoring individual classifiers to the specific characteristics of each traffic cluster, while also reducing the computational burden associated with training a single global classifier on the entire heterogeneous dataset.

4. Benchmark Datasets

The evaluation of ML-based IDS approaches relies critically on the availability of representative and well-curated network traffic datasets. Table 2 summarizes the most widely used benchmark datasets in IDS research, highlighting their key characteristics and limitations.

Table 2. Summary of benchmark datasets used in ML-based IDS research

Dataset	Year	No. of Features	Attack Types	Notes
KDD Cup 99	1999	41	DoS, Probe, R2L, U2R	Widely used benchmark; criticized for redundancy
NSL-KDD	2009	41	DoS, Probe, R2L, U2R	Improved KDD Cup; removes duplicate records
CICIDS2017	2017	80+	DoS, DDoS, Brute Force, XSS, Infiltration	Realistic traffic; labeled flow-based features
UNSW-NB15	2015	49	Fuzzers, Analysis, Backdoors, DoS, Exploits	Modern dataset with 9 attack categories
Kyoto2006+	2006–2011	24	Various network attacks	Long-term real traffic; suitable for clustering
CAIDA	Ongoing	Variable	DDoS	Real-world backbone traffic traces

The KDD Cup 99 dataset, derived from the DARPA 1998 network traffic simulation, has historically been the most widely used benchmark in IDS research. However, it has been subject to significant criticism due to the presence of a large proportion of redundant records and the simulated nature of the attack traffic, which may not accurately reflect contemporary network intrusion patterns. The NSL-KDD dataset was developed to address these limitations by removing duplicate records and rebalancing the class distribution, resulting in a more reliable evaluation benchmark.

More recent datasets such as CICIDS2017 and UNSW-NB15 have been designed to capture contemporary attack patterns in realistic network environments. CICIDS2017 includes a diverse range of modern attack types including DDoS, brute force, cross-site scripting (XSS), and infiltration attacks, generated using realistic network traffic profiles. UNSW-NB15 provides a comprehensive set of 49 features derived from network flow records and covers nine distinct attack categories including fuzzers, backdoors, exploits, and reconnaissance activities. The selection of an appropriate dataset is a critical consideration in IDS research, as the choice of benchmark can substantially influence the reported performance metrics and the generalizability of the findings.

5. Performance Measurement

The rigorous evaluation of ML-based IDS models requires the application of appropriate performance metrics that capture different aspects of detection capability and model reliability. In the context of binary classification, system outputs are characterized by four fundamental outcomes: True Positive (TP), where an actual attack is correctly identified; True Negative (TN), where normal traffic is correctly classified; False Positive (FP), where normal traffic is incorrectly flagged as an attack; and False Negative (FN), where an actual attack is missed [21].

The primary metrics derived from these outcomes are defined as follows. Accuracy (ACC) measures the overall proportion of correct predictions and is calculated as ACC = (TP + TN) / (TP + TN + FP + FN). While accuracy provides a convenient scalar summary of model performance, it can be misleading in the presence of class imbalance, which is a common characteristic of network traffic datasets where normal traffic typically predominates. Recall, also known as the True Positive Rate or sensitivity, is defined as Recall = TP / (TP + FN) and quantifies the model's ability to detect actual attack instances. Precision measures the reliability of positive predictions and is calculated as Precision = TP / (TP + FP). The F1-score provides a harmonic mean of precision and recall, offering a balanced measure that is robust to class imbalance: F1 = 2 × (Precision × Recall) / (Precision + Recall).

For clustering-based IDS models, different evaluation criteria are required since ground-truth class labels may not be available for direct comparison. Homogeneity measures the degree to which each cluster contains instances from a single class, while completeness assesses whether all instances of a given class are grouped within the same cluster. The Rand Index (RI) provides a global measure of agreement between the clustering result and the true class assignments. The Silhouette Score evaluates the quality of cluster separation by comparing the mean intra-cluster distance with the mean distance to the nearest neighboring cluster, with higher values indicating better-defined clusters. The Calinski-Harabasz Index assesses cluster quality by comparing the ratio of inter-cluster dispersion to intra-cluster dispersion, with higher values indicating more compact and well-separated clusters.

Table 3 presents a comparative summary of representative performance metrics reported for commonly used ML algorithms evaluated on standard IDS benchmark datasets.

Table 3. Comparative performance of ML algorithms on IDS benchmark datasets

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	98.5	97.8	98.2	98.0
SVM (RBF Kernel)	97.9	97.1	97.5	97.3
Decision Tree	96.3	95.7	96.1	95.9
K-Nearest Neighbor	95.8	95.0	95.4	95.2
Logistic Regression	93.2	92.5	93.0	92.7
K-Means Clustering	83.6	81.2	82.9	82.0
Hierarchical Clustering	80.1	79.4	80.0	79.7
Ensemble (Stacking)	99.1	98.7	98.9	98.8

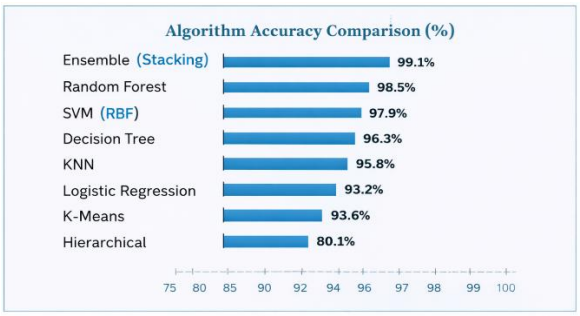

Figure 2. Accuracy comparison of ML algorithms for IDS classification and clustering tasks

As illustrated in Table 3 and Figure 2, ensemble methods, particularly stacking-based approaches, achieve the highest overall accuracy and F1-score, reflecting the advantage of combining multiple complementary base learners. Random Forest and SVM demonstrate consistently strong performance across all metrics, while clustering-based approaches exhibit lower accuracy values due to the inherent challenges of unsupervised detection in the absence of labeled training data. It is important to note that accuracy values reported in the literature can vary substantially depending on the dataset used, the preprocessing applied, and the specific hyperparameter configuration employed, making direct cross-study comparisons challenging.

6. Challenges and Future Directions

Despite the significant advances achieved in ML-based IDS research, several fundamental challenges remain that constrain the real-world deployability and effectiveness of current approaches.

Handling High-Dimensional Data

Contemporary network traffic datasets often contain a large number of features per flow record, encompassing statistical, temporal, and protocol-specific attributes. This high dimensionality increases the computational cost of model training and inference, introduces the risk of the curse of dimensionality, and can degrade model generalization performance. Effective dimensionality reduction and feature selection strategies are therefore essential prerequisites for building efficient and accurate IDS models. Future research should investigate adaptive feature selection frameworks that can dynamically identify the most discriminative features in response to evolving attack patterns.

Class Imbalance and Data Distribution

Network traffic datasets are inherently imbalanced, with normal traffic instances typically outnumbering attack instances by several orders of magnitude. This class imbalance can bias ML models toward the majority class, resulting in high accuracy but poor detection rates for rare attack types. Techniques such as oversampling, undersampling, cost-sensitive learning, and synthetic data generation require further investigation to develop IDS models that maintain high detection rates across all attack categories.

Computational Efficiency and Lightweight Models

Many high-performing ML-based IDS solutions rely on computationally intensive algorithms such as deep neural networks and large ensemble models, which may not be suitable for deployment in resource-constrained environments such as Internet of Things (IoT) networks, embedded systems, and edge computing platforms. The design of lightweight, energy-efficient detection models that maintain competitive detection accuracy while operating within strict computational and memory constraints is an important direction for future research.

Generalization to Novel Attack Types

ML models trained on existing benchmark datasets may fail to generalize effectively to novel and previously unseen attack patterns, particularly as adversaries continuously develop new evasion and obfuscation techniques. Transfer learning, continual learning, and zero-shot detection approaches represent promising research directions for developing IDS models that can adapt to evolving threat landscapes without requiring complete retraining on updated datasets.

Hybrid and Multi-Stage Detection Frameworks

The integration of multiple detection strategies within a unified framework has the potential to leverage the

complementary strengths of both supervised and unsupervised approaches. For example, clustering techniques can be used as a preprocessing step to identify structurally distinct traffic clusters, which are subsequently processed by specialized classification models trained on cluster-specific data. Further research is needed to develop principled methodologies for designing and evaluating such hybrid detection architectures.

Explainability and Interpretability

The deployment of ML-based IDS in production environments requires not only high detection accuracy but also the ability to provide interpretable and actionable explanations for classification decisions. Black-box models such as deep neural networks and large random forests may achieve superior performance metrics but lack the transparency necessary for security analysts to understand and trust their decisions. Future research should investigate explainable AI (XAI) techniques that can provide meaningful insights into the detection logic of ML-based IDS models.

7. Conclusion

This paper has presented a comprehensive review of clustering and classification techniques applied in machine learning-based Intrusion Detection Systems. The study examined a range of foundational algorithms spanning both supervised and unsupervised learning paradigms, including Logistic Regression, K-Nearest Neighbor, Decision Tree, Random Forest, Support Vector Machine, ensemble methods, K-Means, and Hierarchical Clustering. Each algorithm was analyzed with respect to its operational principles, strengths, and limitations in the context of network intrusion detection.

A systematic comparative analysis of related work was conducted, highlighting the consistent performance advantages of ensemble approaches over individual classifiers, the value of feature selection and dimensionality reduction in improving detection accuracy, and the promising potential of hybrid detection frameworks that integrate clustering and classification in a staged pipeline. Widely used benchmark datasets including KDD Cup 99, NSL-KDD, CICIDS2017, UNSW-NB15, and Kyoto2006+ were described and critically assessed, and key performance metrics for both classification and clustering-based IDS evaluation were formally defined.

The review identified several unresolved research challenges, including the management of high-dimensional feature spaces, the handling of class imbalance, the development of lightweight models for resource-constrained environments, the generalization to novel attack patterns, and the provision of interpretable detection decisions. Addressing these challenges will require interdisciplinary research efforts drawing on advances in machine learning, network security, and systems engineering.

Future work should focus on the development of adaptive, lightweight, and hybrid IDS frameworks that can maintain high detection performance across diverse and evolving network environments while meeting the practical constraints of real-world deployment. The continued advancement of publicly available, realistic, and representative benchmark datasets will also be essential for enabling rigorous and reproducible evaluation of next-generation ML-based IDS solutions.

References

[1] F. Mokbal, W. Dan, M. Osman, Y. Ping, and S. Alsamhi, "An Efficient Intrusion Detection Framework Based on Embedding Feature Selection and Ensemble Learning Technique," Int. Arab J. Inf. Technol., vol. 19, no. 2, pp. 237–248, 2022.

[2] F. Kamalov, S. Moussa, R. Zgheib, and O. Mashaal, "Feature selection for intrusion detection systems," in Proc. 13th Int. Symp. Computational Intelligence and Design (ISCID), IEEE, Dec. 2020, pp. 265–269.

[3] W. Du and Z. Zhan, "Building decision tree classifier on private data," 2002.

[4] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, no. 1, pp. 81–106, 1986.

[5] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," IEEE Commun. Surv. Tutorials, vol. 18, no. 2, pp. 1153–1176, 2015.

[6] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," Int. J. Inf. Secur., vol. 22, no. 5, pp. 1125–1162, 2023.

[7] M. Zaman and C.-H. Lung, "Evaluation of machine learning techniques for network intrusion detection," in Proc. NOMS 2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–5.

[8] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection," Eng. Appl. Artif. Intell., vol. 94, p. 103770, 2020.

[9] N. V. Sharma and N. S. Yadav, "An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers," Microprocess. Microsyst., vol. 85, p. 104293, 2021.

[10] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," Comput. Secur., vol. 77, pp. 304–314, 2018.

[11] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A two-stage classifier ensemble for intelligent anomaly-based intrusion detection system," IEEE Access, vol. 7, pp. 94497–94507, 2019.

[12] W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "CANN: An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge-Based Syst., vol. 78, pp. 13–21, 2015.

[13] D. P. Gaikwad and R. C. Thool, "Intrusion detection system using bagging with partial decision tree base classifier," Procedia Comput. Sci., vol. 49, pp. 92–98, 2015.
[14] S. Rajagopal, P. P. Kundapur, and K. S. Hareesha, "A stacking ensemble for network intrusion detection using heterogeneous datasets," Secur. Commun. Networks, vol. 2020, p. 4586875, 2020.
[15] I. Dutt et al., "Real-time hybrid intrusion detection system using machine learning techniques," in Advances in Communication, Devices and Networking, Springer, 2018, pp. 885–894.
[16] N. Iftikhar et al., "Intrusion Detection in NSL-KDD Dataset Using Hybrid Self-Organizing Map Model," Comput. Model. Eng. Sci., vol. 143, 2025.
[17] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering," Expert Syst. Appl., vol. 37, no. 9, pp. 6225–6232, 2010.
[18] R. Chitrakar and C. Huang, "Anomaly based intrusion detection using hybrid learning approach of combining k-medoids clustering and naive bayes classification," in Proc. 8th Int. Conf. Wireless Communications, Networking and Mobile Computing, IEEE, 2012, pp. 1–5.
[19] Z. Muda, W. Yassin, M. N. Sulaiman, and N. I. Udzir, "K-means clustering and naive bayes classification for intrusion detection," J. IT Asia, vol. 4, no. 1, pp. 13–25, 2014.
[20] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading k-Means clustering and C4.5 decision tree algorithm," Procedia Eng., vol. 30, pp. 174–182, 2012.
[21] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," ACM Trans. Inf. Syst. Secur., vol. 3, no. 3, pp. 186–205, 2000.