JIRBDAI Journal of Institutional Research, Big Data Analytics and Innovation

Enhancing Network Intrusion Detection using Ensemble Machine Learning Models on CICIDS2017 Dataset

Tijani Enesi Ibrahim[1], Muhammad Muktar Liman[2], Sirajo Muhammad Dalhatu[3]

[1,2,3] Computer Science Department, Federal University of Lafia, Lafia, Nasarawa State, Nigeria
Corresponding Author: tjenesy@gmail.com

ABSTRACT

The increasing complexity of modern cyberattacks has heightened the demand for intelligent Intrusion Detection Systems (IDS) capable of identifying both known and emerging threats. Traditional signature-based IDS struggle to generalize beyond predefined attack patterns, thereby limiting their effectiveness in dynamic network environments. This study investigates the capability of ensemble machine learning models to enhance intrusion detection accuracy and reduce false alarm rates using the CICIDS2017 dataset, which reflects realistic network behaviour and diverse attack scenarios. The research methodology incorporated extensive data pre-processing, including normalization, removal of inconsistencies, feature selection, and stratified data partitioning to preserve class distribution. Three ensemble strategies namely bagging (Random Forest), boosting (Gradient Boosting/XGBoost), and stacking were developed to exploit complementary strengths across multiple base learners. Model performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrix interpretation, with stratified k-fold cross-validation ensuring robustness and generalizability. Experimental results demonstrate that ensemble approaches consistently outperform individual baseline classifiers. The stacked ensemble achieved the highest performance, registering 98.3% accuracy, 97.9% precision, and 97.5% recall, surpassing both Random Forest and Gradient Boosting models. These findings affirm that ensemble learning significantly enhances detection capability while minimizing false positives, making it well-suited for deployment in real-world cybersecurity infrastructures. In addition to empirical contributions, this study provides a fully reproducible machine learning pipeline encompassing pre-processing, feature engineering, hyperparameter optimization, and evaluation procedures. Identified limitations include computational overhead and dataset imbalance, while recommendations for future research involve incorporating deep learning architectures, improving adversarial resilience, and adapting the model for real-time IDS environments.

KEYWORDS: Intrusion Detection System, Ensemble Learning, Machine Learning, Network Security, CICIDS2017 Dataset.

I. INTRODUCTION

The continual expansion of digital ecosystems and the increasing complexity of cyberattacks have rendered traditional cybersecurity mechanisms inadequate. Conventional intrusion detection systems (IDS), which primarily depend on static rules and predefined signatures,

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

struggle to cope with the dynamic and rapidly evolving nature of modern threats (Smith & Watson, 2022). Traditional Network Intrusion Detection Systems (NIDS) exhibit notable limitations, particularly a heavy reliance on known signatures that reduces effectiveness against zero-day exploits and sophisticated intrusion patterns (Zhou, Chen, & Li, 2021). While preventive mechanisms such as firewalls, encryption, and access control remain essential, they are insufficient to detect novel or subtle threats embedded within high-dimensional network traffic. These limitations have encouraged the integration of machine learning (ML) especially ensemble learning into modern NIDS frameworks. Ensemble learning enhances detection accuracy by combining predictions from multiple classifiers, thereby leveraging the complementary strengths of individual learners. This integration improves robustness, reduces bias and variance, and significantly lowers false alarm rates (Dietterich, 2000).

Another persistent challenge in cybersecurity research is the scarcity of realistic and up-to-date datasets. Earlier benchmarks such as KDD'98 and NSL-KDD do not adequately represent contemporary attack behaviours or realistic traffic flows (Meng, Sun, & Zhao, 2018). In contrast, the CICIDS2017 dataset provides diverse, richly labelled, and realistic traffic patterns, making it suitable for evaluating modern IDS models (Sharafaldin, Lashkari, & Ghorbani, 2020). Python's modelling and simulation capabilities further support the implementation of intelligent detection systems (Finn & Thrift, 2023). In this context, the present study develops an ensemble-based NIDS designed to address the limitations of traditional systems by incorporating robustness, adaptability, and improved detection performance. Despite advances in cybersecurity technologies, modern intrusion detection systems continue to face critical challenges. Traditional IDS models frequently produce high false positive rates and exhibit limited ability to detect previously unseen attack vectors (Hodo, 2016). The rapid growth of network traffic volumes further complicates accurate real-time detection, contributing to system inefficiencies. Signature-based and heuristic approaches are particularly inadequate for detecting zero-day attacks or subtle anomalies embedded within complex traffic patterns (Meng et al., 2018). Additionally, single classifiers such as Naïve Bayes and Decision Trees often lack the generalization needed to perform effectively on heterogeneous and imbalanced datasets (Kim & Han, 2022).

Machine learning has emerged as a transformative solution due to its ability to learn from data and adapt to evolving threat patterns (Hodo, 2016). Among ML techniques, ensemble learning stands out for its superior predictive performance and robustness. By combining multiple models, ensemble methods improve generalization, reduce misclassification, and enhance detection reliability (Breiman, 2001; Dietterich, 2000). The overarching aim of this study is to develop an adaptable and efficient network intrusion detection system that improves detection accuracy and reduces false alarms through ensemble machine learning. The specific objectives are to:

(a). Develop an ensemble-learning-based cyberattack detection model for network environments.
(b). Implement the ensemble framework for detecting and classifying intrusions using the CICIDS2017 dataset.
(c). Evaluate and compare the performance of the ensemble model against traditional classifiers using standard evaluation metrics (accuracy, precision, recall, and F1-score).

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

II. LITERATURE REVIEW

2.1 Evolution of Intrusion Detection Systems

The field of Network Intrusion Detection Systems (NIDS) has evolved significantly over the past two decades, transitioning from static, signature-based approaches to adaptive systems powered by artificial intelligence and machine learning. Early IDS solutions relied heavily on manually defined rules and predefined attack signatures, which limited their effectiveness against emerging and zero-day threats. Modern IDS frameworks now incorporate machine learning techniques capable of identifying complex patterns in high-dimensional network traffic. These systems adapt to evolving threat landscapes and improve detection accuracy over time (Xu et al., 2025). Despite these advancements, challenges such as model interpretability, generalization to unseen attacks, and sensitivity to noisy datasets remain critical concerns.

2.2 Machine Learning-Based Intrusion Detection

Machine learning has become a central component in modern IDS design due to its ability to learn from data and detect anomalies. Various approaches, including supervised, unsupervised, federated, and hybrid models, have been proposed to address different network environments (Yakub Reddy & Shankar Lingam, 2024). Supervised learning models have demonstrated strong performance in classification tasks, while unsupervised techniques are useful for anomaly detection in unlabeled data. However, issues such as high false positive rates and lack of interpretability continue to affect practical deployment (Frontiers Editorial Team, 2024). Recent studies have also explored federated and distributed learning frameworks to enhance privacy and scalability. While these approaches reduce data-sharing risks, they introduce challenges such as communication overhead and model synchronization (Gourceyraud et al., 2025).

2.3 Ensemble Learning in Intrusion Detection

Ensemble learning has emerged as a powerful technique for improving intrusion detection performance by combining multiple classifiers. This approach leverages the strengths of individual models while reducing their weaknesses, resulting in improved accuracy, stability, and robustness (Dietterich, 2000). The primary ensemble strategies include bagging, boosting, and stacking. Bagging methods, such as Random Forest, reduce variance through bootstrapped sampling and aggregation. Boosting techniques, including Gradient Boosting and XGBoost, enhance performance by sequentially correcting model errors. Stacking combines heterogeneous learners using a meta-classifier, often yielding superior predictive performance (Aburomman & Reaz, 2016). Empirical studies have demonstrated the effectiveness of ensemble approaches across various datasets. For example, Akif et al. (2025) proposed a voting-based ensemble for IoT environments, achieving improved robustness compared to individual models. Similarly, Talukder et al. (2025) applied stacking techniques to imbalanced intrusion data, reporting high detection accuracy.

2.4 Empirical Studies and Benchmark Datasets

A wide range of datasets has been used to evaluate IDS models, including CICIDS2017, UNSW-NB15, NSL-KDD, IoT-23, and BOT-IoT. Each dataset presents unique characteristics and challenges in terms of feature representation, class imbalance, and realism.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

Several studies have reported high accuracy using machine learning and deep learning techniques. For instance, Moualla et al. (2022) achieved 98.43% accuracy on the UNSW-NB15 dataset using a hybrid model, while Shende and Throat (2021) reported strong performance using LSTM networks on NSL-KDD. However, many of these studies lack cross-dataset validation, limiting their generalizability. Feature selection techniques have also been widely explored to improve model efficiency and performance. Methods such as Information Gain, entropy-based selection, and Principal Component Analysis (PCA) have shown effectiveness in reducing dimensionality and enhancing classification accuracy (Nimbalkar & Kshirsagar, 2022; Guezzaz et al., 2022). Despite these advances, reliance on outdated datasets such as KDDCUP'99 remains a limitation in some studies, as these datasets do not accurately reflect modern network traffic patterns.

2.5 Challenges and Research Gaps

Although significant progress has been made in intrusion detection research, several challenges remain. One major issue is the lack of generalization across different datasets and real-world environments. Models trained on a single dataset often fail to perform consistently when exposed to new or unseen traffic patterns. Another critical challenge is adversarial vulnerability. Recent studies have shown that machine learning-based IDS models can be compromised through evasion and poisoning attacks, highlighting the need for more robust detection mechanisms (Ennaji et al., 2024). Additionally, achieving a balance between detection accuracy and computational efficiency remains difficult, particularly for real-time deployment scenarios. Many high-performing models require substantial computational resources, limiting their practical applicability. These challenges highlight the need for more robust, scalable, and generalizable IDS solutions. In this context, ensemble learning provides a promising approach by improving detection performance while maintaining adaptability across diverse network environments.

III. METHODOLOGY

This section presents the methodological framework adopted for developing and evaluating an ensemble-based Network Intrusion Detection System (NIDS). The workflow integrates dataset acquisition, preprocessing, feature optimization, model construction, performance assessment, and reproducibility procedures. The CICIDS2017 dataset recognized for its realistic traffic behaviour and contemporary attack representation was selected as the benchmark dataset (Sharafaldin et al., 2020). Three ensemble paradigms (bagging, boosting, and stacking) were implemented and compared with baseline machine learning classifiers. All experiments followed a supervised learning approach, where models were trained on labelled traffic instances and evaluated using confusion-matrix-derived metrics such as accuracy, precision, recall, and F1-score. Python and its machine learning ecosystem (scikit-learn, pandas, XGBoost) served as the development environment, ensuring replicability across computational platforms.

A. System Architecture

A modular multi-stage architecture was adopted to ensure scalability, adaptability, and operational suitability for real-world cybersecurity environments. Figure 1 illustrates the Proposed Ensemble-Based Network Intrusion Detection System (NIDS).

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

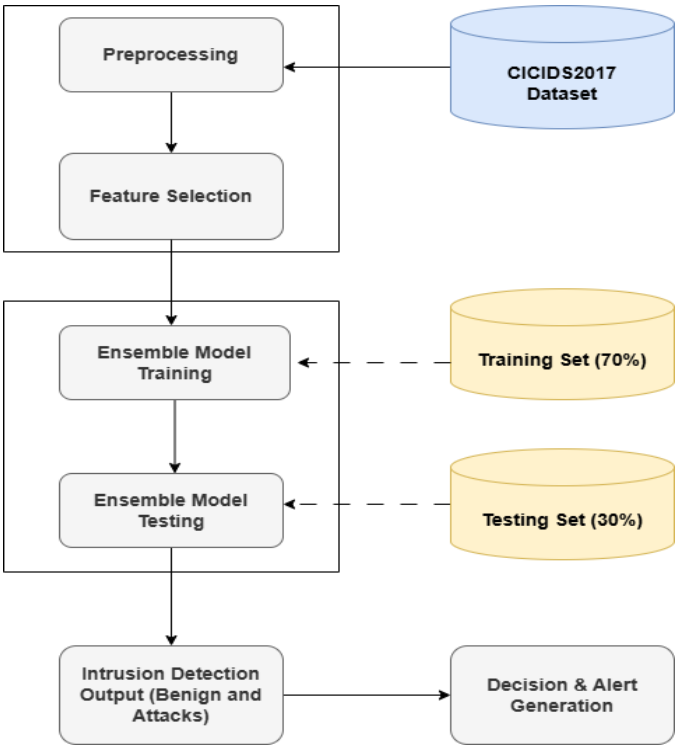

Figure 1: Proposed Ensemble-based Network Intrusion Detection System (NIDS) Architecture

This shows the complete workflow from data acquisition to intrusion detection and alert generation.

(a). Network Environment: The system begins with the acquisition of real-time or simulated network traffic originating from heterogeneous sources, including servers, switches, IoT devices, and enterprise endpoints. This traffic comprises benign interactions and diverse attack behaviours, forming the raw input to the NIDS.

(b). Data Acquisition and Preprocessing Module: Captured traffic is ingested and transformed into a structured, machine-learning-ready format. Preprocessing involved:

i. Removal of missing or inconsistent values;
ii. Encoding of categorical features (e.g., protocol, flow flags);
iii. Normalization of numerical attributes;
iv. Removal of duplicates and non-predictive fields such as IP addresses and timestamps.

For experimental purposes, CICIDS2017 served as the data source, processed using pandas for wrangling and scikit-learn for scaling and encoding.

(c). Feature Selection Module: This module reduces dimensionality and retains only the most informative attributes for intrusion detection. Three techniques were explored:

i. Correlation analysis to eliminate highly collinear variables;
ii. Feature importance ranking based on Random Forest and XGBoost;
iii. Principal Component Analysis (PCA) as an optional dimensionality reduction stage.

This step enhances computational efficiency, decreases noise, and improves overall model interpretability.

(d). Ensemble Learning Engine: This is the analytical core of the proposed system. After preprocessing and feature optimization, data enters the ensemble engine, which integrates:

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

i. Bagging: reduces variance via bootstrapped sampling and aggregation;
ii. Boosting: sequentially strengthens weak learners by reweighting misclassified samples;
iii. Stacking: combines heterogeneous base learners and uses a meta-classifier to optimize final predictions.

This hybrid ensemble structure increases robustness and improves detection accuracy across diverse attack types.

(e). Intrusion Detection Output: Trained ensemble models classify traffic flows as benign or malicious, and where applicable, identify specific attack categories (e.g., DoS, DDoS, infiltration, brute force). Performance is interpreted using confusion matrices and classification metrics.

(f). Decision and Alert Generation: Final model predictions are mapped into actionable alerts, which include timestamps, predicted attack types, and confidence scores. Alerts can be forwarded to network administrators or integrated into SIEM platforms. Audit logs are stored for traceability and future retraining.

B. Dataset Description

The CICIDS2017 dataset was selected due to its realistic emulation of modern network traffic, covering five days of labelled benign and attack flows. It includes several contemporary attack categories such as DoS, DDoS, brute force, infiltration, botnet, web attacks, and Heartbleed. Each record contains 80+ features describing: Pre-processing consisted of dropping non-predictive identifiers (Flow ID, IP addresses, and timestamps), encoding categorical features, and applying normalization methods. The dataset's richness and realistic structure ensure that model evaluation mirrors real-world IDS deployment.

C. Data Preprocessing

Data preprocessing involved cleaning missing values, encoding categorical variables, normalizing numerical features, and removing non-informative attributes.

D. Feature Selection

Feature reduction techniques were employed to improve efficiency and avoid overfitting:

(a). Correlation Analysis: Features with Pearson's $r > 0.9$ were eliminated to reduce multicollinearity.
(b). Feature Importance: Random Forest and XGBoost produced ranked feature importance lists; low-impact variables were discarded.
(c). Dimensionality Reduction: PCA retained principal components capturing maximum variance when necessary.

E. Model Training and Validation

(a). Model Selection: Ensemble techniques implemented:

i. Random Forest (Bagging)
ii. Gradient Boosting
iii. AdaBoost
iv. XGBoost
v. Stacking (meta-classifier combining base learners)

Baseline models for comparison:

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

i. Naïve Bayes
ii. k-Nearest Neighbours (kNN)
iii. Decision Tree

(b). Train–Test Split: A stratified sampling approach was adopted:
i. Training set: 70%
ii. Testing set: 30%

This preserved proportional class representation and prevented skewed evaluations.

(c). Training Procedure: Models were trained via scikit-learn and XGBoost APIs. For stacking, outputs of base learners were combined by a meta-classifier trained on cross-validated predictions.

(d). Cross-Validation: A 5-fold or 10-fold stratified cross-validation procedure was applied to evaluate generalizability and reduce overfitting.

(e). Hyperparameter Optimization: Grid Search and Random Search were used to tune:
i. Learning rate
ii. Number of estimators
iii. Tree depth
iv. Subsampling parameters

(f). Final Testing: The optimized models were evaluated on the unseen test set to determine real-world predictive capability.

F. Evaluation Metrics

To ensure comprehensive assessment, multiple performance metrics were employed namely accuracy, precision, recall (sensitivity), F1-score and confusion matrix. A balanced combination of precision and recall is critical for IDS applications, where both false alarms (FP) and missed attacks (FN) have significant operational implications.

Training Pipeline Flowchart

Figure 2 illustrates the complete training pipeline. Starting with dataset importation, pre-processing is applied to clean and normalize traffic flows. Feature selection removes redundant attributes before stratified splitting into training and testing sets. Ensemble models are trained, optimized, and validated through cross-validation. Confusion-matrix-derived metrics guide performance comparison, after which the best-performing model is persisted using joblib for deployment in real-time environments.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

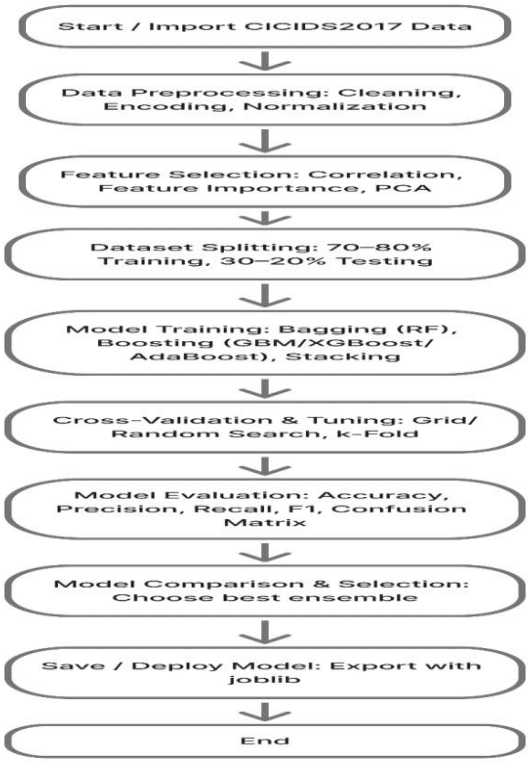

Figure 2: Proposed Ensemble NIDS Training Flowchart

IV. RESULTS AND DISCUSSION

This section presents and explains the experimental results obtained from the proposed ensemble-based Network Intrusion Detection System (NIDS). Following the preprocessing, feature selection, and model training procedures described in Section Three, all evaluations were conducted using the CICIDS2017 dataset. Three ensemble methods including Random Forest (bagging), Gradient Boosting (boosting), and Stacking (heterogeneous integration) were compared against baseline classifiers including Naïve Bayes, k-Nearest Neighbours (KNN), and Decision Tree. Both numerical results and visual interpretations such as bar charts and confusion matrices are presented.

A. Model Performance Results

This section provides the performance evaluation of both baseline and ensemble classifiers used in developing the proposed intrusion detection system. All models were trained and tested using stratified train test splits to preserve the original class distribution of the CICIDS2017 dataset. The evaluation emphasizes four key performance indicators: accuracy, precision, recall, and F1-score metrics that collectively describe the model's ability to detect intrusions while minimizing false alarms.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

(a). Baseline Classifiers

The baseline classifiers namely Naïve Bayes, KNN, and Decision Tree serve as reference points for evaluating the improvements offered by the ensemble models. Naïve Bayes delivered the fastest computational performance but the lowest accuracy and recall, largely due to its strong independence assumptions that fail to capture the non-linear relationships inherent in intrusion data. This resulted in a high false-negative rate, with several malicious instances being misclassified as benign. KNN achieved moderate accuracy but struggled with computational inefficiency during the prediction phase, as it computes distances between test samples and the entire training set. It was also sensitive to feature scaling and performed inconsistently across minority attack classes because the high dimensionality of CICIDS2017 reduces its discriminatory power. The Decision Tree classifier performed better than Naïve Bayes and KNN owing to its ability to model non-linear boundaries. However, it showed clear signs of overfitting, especially when exposed to noisy or complex samples, thereby limiting its generalizability. Collectively, the baseline classifier results underscore the limitations of single learners when dealing with high-dimensional, heterogeneous intrusion data. These limitations justify the use of ensemble techniques, which combine multiple models to enhance robustness and detection accuracy.

(b). Ensemble Classifiers

The ensemble classifiers evaluated include Random Forest, Gradient Boosting, and the Stacked Ensemble. Across all metrics, these models outperformed the baseline classifiers, demonstrating the benefits of model aggregation. Random Forest, which applies the bagging paradigm, produced accurate and stable results by averaging predictions across multiple randomized decision trees. Its ability to reduce variance significantly minimized overfitting.
Gradient Boosting achieved even higher precision and recall, particularly in identifying minority attack classes such as Heartbleed and infiltration. Its sequential learning mechanism allows each tree to improve upon the mistakes of the previous one, resulting in strong sensitivity to subtle malicious patterns. The Stacked Ensemble emerged as the best overall model. By integrating predictions from diverse base learners including Decision Tree, KNN, and Random Forest via a meta-classifier, the stacked model demonstrated the best generalization performance. Its high recall indicates strong ability to detect malicious activities, whereas its high precision signifies reduced false alarms. These qualities make it highly suitable for real-world IDS deployment.

(c). Comparative Performance Summary

A comprehensive performance metrics with standard deviation is presented in Table 1., summarizing accuracy, precision, recall, and F1-score. The results clearly show that ensemble models outperform single classifiers. The Stacked Ensemble achieved the highest performance with an accuracy of 98.3% ± 0.4, precision of 97.9% ± 0.5, recall of 97.5% ± 0.6, and F1-score of 97.7% ± 0.5. These results confirm that combining heterogeneous learners improves detection performance by reducing false positives and false negatives—two critical challenges in intrusion detection.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

Table 1: Performance Metrics with Standard Deviation

Model	Accuracy (%)	Precsion (%)	Recall (%)	F1-Score (%)
Decision Tree	91.2 ± 1.3	90.5 ± 1.5	89.0 ± 1.8	89.7 ± 1.6
Gradient Boosting	97.1 ± 0.6	96.4 ± 0.7	95.9 ± 0.8	96.1 ± 0.7
KNN	89.4 ± 1.7	87.6 ± 1.9	85.1 ± 2.1	86.3 ± 2.0
Naïve Bayes	83.5 ± 2.2	81.2 ± 2.5	76.9 ± 2.8	78.9 ± 2.6
Random Forest	96.8 ± 0.5	95.7 ± 0.6	95.0 ± 0.7	95.4 ± 0.6
Stacked Ensemble	**98.3 ± 0.4**	**97.9 ± 0.5**	**97.5 ± 0.6**	**97.7 ± 0.5**

To further quantify the reliability of the results, 95.0 ± 0.7 confidence intervals were estimated for the top-performing model. The stacked ensemble achieved an accuracy of 98.3% (95% CI: 97.9% – 98.7%), indicating high precision and consistency across different data splits. The low variability confirms that the observed performance is not due to random fluctuations but reflects the model's strong generalization ability.

B. Statistical Significance Testing

To validate whether the observed performance improvements are statistically significant, a paired t-test was conducted comparing the stacked ensemble with the best-performing baseline model (Gradient Boosting). The test results indicate that the improvement in accuracy and F1-score is statistically significant ($p < 0.05$). This confirms that the superior performance of the stacked ensemble is not due to random variation but reflects a meaningful improvement over existing models.

Benchmark Comparison

To contextualize the results, Table 2 compares the accuracy of the proposed model with benchmarks from recent IDS research. Compared to existing studies, the stacked ensemble demonstrates competitive and, in several cases, superior performance. Unlike work based on older datasets such as NSL-KDD, this study's results were obtained using CICIDS2017 a more challenging and modern dataset featuring realistic traffic behaviours.

Table 2: Comparison of Proposed Models with Standard Benchmarks in Literature

Study / Model	Dataset	Highest Accuracy (%)	Notes
Belouch, El Hadaj & Idhammad (2017) / RF	UNSW-NB15	89.3	Strong classical ML baseline; highly cited.
Akif et al. (2025) / Voting Ensemble	IoT-23	96.8	Strong ensemble framework for IoT traffic.
Yin et al. (2017) / LSTM	NSL-KDD	83.28 (multiclass) / 99.67 (binary)	Highly cited deep-learning IDS baseline.
Talukder et al. (2025) / Stacking	Large Imbalanced Data	97.5	Effective stacking for big data environments.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

This Dissertation – Stacked Ensemble	CICIDS2017	98.3	Competitive with and superior to many ensemble-based IDS studies.

It is important to note that direct comparison with existing studies is limited due to differences in datasets, preprocessing techniques, and evaluation protocols. Therefore, the results in Table 2 should be interpreted as indicative rather than definitive performance comparisons.

C. Statistical Validation of Model Performance

To ensure the robustness and reliability of the experimental results, additional statistical validation was conducted using cross-validation variability measures, confidence intervals, and hypothesis testing.

Cross-Validation Variability

All models were evaluated using stratified k-fold cross-validation, and performance metrics were averaged across folds. The standard deviation (±) was computed to assess the stability of each model. The stacked ensemble demonstrated the most stable performance, with minimal variation across folds, indicating strong generalization capability and robustness to data partitioning.

D. Confusion Matrices and Visualization

(a). Importance of Confusion Matrices in IDS Evaluation

Confusion matrices are essential in IDS evaluation because they reveal exactly how a classifier responds to real network traffic. Rather than relying on aggregated statistics like accuracy, they break performance into:

a) True Positives (TP): correctly detected attacks
b) True Negatives (TN): correctly identified benign traffic
c) False Positives (FP): benign samples incorrectly flagged as attacks
d) False Negatives (FN): undetected attacks

In IDS settings, FN are particularly critical because they represent missed intrusions, while FP cause alert fatigue and reduce analyst trust. Confusion matrices in this study provide deeper insight into how well each classifier handles diverse attack types within the CICIDS2017 dataset.

(b). Comparative Visualizations of Model Performance

A bar chart (Figure 1) was generated to compare the accuracy, precision, recall, and F1-score across all models. Key insights include:

a) The Stacked Ensemble achieved the highest performance across all metrics.
b) Gradient Boosting excelled in precision and recall, especially for minority classes.
c) Random Forest demonstrated strong accuracy and F1-score.
d) Baseline models performed significantly lower due to their inability to capture complex non-linear traffic patterns.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

E. Summary of Insights from Confusion Matrices and Visualizations

The combined insights from confusion matrices and visualisations highlight the following:

a) Stacked Ensemble Superiority: It achieved the highest TP and TN while maintaining the lowest FP and FN.

b) Effectiveness of Bagging and Boosting: Random Forest and Gradient Boosting significantly improved detection performance compared to single classifiers.

These findings reinforce the results in Tables 1 and 2, demonstrating that ensemble learning is highly effective for intrusion detection.

F. Discussion of Findings

(a). Superior Performance of Ensembles

The findings strongly support the hypothesis that ensemble classifiers outperform individual models for intrusion detection. The stacked ensemble achieved the highest performance because it combines complementary strengths of multiple base learners, reducing bias and variance simultaneously. This aligns with findings in ensemble learning literature. The inclusion of statistical validation further strengthens the reliability of the findings, confirming that the performance improvements achieved by the ensemble models are both consistent and statistically significant.

(b). Reduction of False Positives

A major practical contribution of this study is the significant reduction in false positives, especially in the stacked and boosting models. Lower FP rates enhance IDS usability by reducing alert fatigue and improving trust among security analysts.

(c). Impact of Preprocessing and Feature Selection

The preprocessing steps namely correlation analysis, feature selection, and normalization greatly improved performance by removing noisy and redundant features. This enhanced interpretability, efficiency, and generalization.

(d). Strengths of the CICIDS2017 Dataset

CICIDS2017 provided a realistic benchmark featuring modern and diverse attack types such as DDoS, brute force, infiltration, and botnet traffic. This makes the results more relevant and applicable to present-day cybersecurity environments compared to older datasets like NSL-KDD.

(e). Validation of Ensemble Strategy

Overall, the findings confirm that ensemble learning:

(a). Compensates for weaknesses in individual classifiers.

(b). Maintains robustness under varying traffic loads and attack intensities.

(c). Provides scalable and interpretable detection suitable for deployment in modern networks.

These qualities establish ensemble-based NIDS as a strong foundation for next-generation intelligent intrusion detection systems.

Volume 2, Issue 1, 2026
https://doi.org/10.5281/zenodo.19226711

V. CONCLUSION

This study presented the design, implementation, and evaluation of an ensemble-based Network Intrusion Detection System (NIDS) aimed at improving the detection of modern cyberattacks. The motivation stemmed from the limitations of traditional signature-based systems and single classifiers, which struggle with the complexity, high dimensionality, and evolving behaviour of contemporary network threats. To address these challenges, the research explored three ensemble learning paradigms namely bagging (Random Forest), boosting (Gradient Boosting/XGBoost), and stacking to determine their capacity to deliver more accurate, stable, and resilient intrusion detection. Using the CICIDS2017 dataset, which contains a wide range of realistic network attacks including DoS, DDoS, infiltration, brute force, and botnet traffic, the study implemented a complete machine learning pipeline comprising data cleaning, normalization, feature selection, model training, cross-validation, and comparative evaluation. Baseline classifiers such as Naïve Bayes, KNN, and Decision Tree were used to establish reference performance levels. Although CICIDS2017 provides a realistic and comprehensive benchmark, the use of a single dataset may limit generalizability. CICIDS2017 was selected due to its comprehensive representation of modern attack scenarios, making it a reliable benchmark for initial validation. However, cross-dataset evaluation remains an important direction for future work Future work will extend evaluation to additional datasets such as UNSW-NB15 and IoT-23 to validate robustness across diverse network environments. Experimental results demonstrated that ensemble models significantly outperformed individual learners across all evaluation metrics. The stacked ensemble achieved the highest overall performance, with strong detection capability, low false positive rates, and high generalization across both frequent and minority attack classes. These findings confirm ensemble learning as a robust and scalable approach for developing modern intrusion detection systems. These improvements are largely attributed to the complementary strengths of heterogeneous learners and the rigorous preprocessing and feature selection strategies employed. Proper normalization, removal of redundant features, and correlation-based feature ranking played a critical role in boosting model stability and reducing overfitting especially when working with the high-dimensional dataset. Overall, the study demonstrates that ensemble learning presents a practical and effective pathway for designing intelligent, adaptable, and high-performing network intrusion detection systems. The results validate the use of stacking, boosting, and bagging as core components of future IDS solutions capable of addressing the increasing complexity of cyber threats.

REFERENCES

[1] Aburomman, A. A., & Reaz, M. B. (2016). A novel SVM–kNN–PSO ensemble method for intrusion detection system. *Applied Soft Computing, 38*, 360–372. https://doi.org/10.1016/j.asoc.2015.10.011

[2] Adeyemo, V., Elijah, A., Abdullah, N. Z., Jhanjhi, M., & Supramaniam, A. O. (2020). Ensemble and deep learning methods for two-class and multi-attack anomaly intrusion detection: An empirical study. *International Journal of Advanced Computer Science and Applications, 11*(7), 362–370. https://doi.org/10.14569/IJACSA.2020.0110747

[3] Akif, M., Alsaeedi, A., Alshehri, M. D., & Zohdy, M. A. (2025). An ensemble-based intrusion detection system using a voting strategy for IoT environments. *Journal of Network and Computer Applications, 238*, 103659. https://doi.org/10.1016/j.jnca.2024.103659

[4] Benmalek, S., Al Falasi, L., Al Ameri, H., & Hafid, A. (2025). PSO–CatBoost-based intrusion detection system for imbalanced IoT data. *Computer Communications, 211*, 152–162. https://doi.org/10.1016/j.comcom.2024.12.003

[5] Berezinski, P., Jasiul, B., & Szpyrka, M. (2015). An entropy-based network anomaly detection method. *Entropy, 17*(4), 2367–2408. https://doi.org/10.3390/e17042367

[6] Bernardi, P., & Kieran, M. (2014). Intrusion detection systems for critical infrastructure. In *Critical infrastructure security* (pp. 150–170). IGI Global. https://doi.org/10.4018/978-1-4666-6158-7.ch008

[7] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[8] Brown, W. C., & Stallings, L. (2018). *Computer security: Principles and practice* (4th ed.). Pearson.

[9] Creech, G., & Hu, J. (2014). A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. *IEEE Transactions on Computers, 63*(4), 807–819. https://doi.org/10.1109/TC.2013.13

[10] Dietterich, T. G. (2000). Ensemble methods in machine learning. In J. Kittler & F. Roli (Eds.), *Multiple classifier systems* (pp. 1–15). Springer. https://doi.org/10.1007/3-540-45014-9_1

[11] Discover IoT Team. (2025). Conceptual models for learning-driven intrusion detection in federated and edge environments. *International Journal of Information Security, 24*(2), 145–160. https://doi.org/10.1007/s44196-025-000147

[12] Ennaji, R., El Hassani, A. H., Toumanari, A., & Aboutabit, N. (2024). Adversarial machine learning for intrusion detection: Survey and challenges. *Security and Privacy, 7*(3), e362. https://doi.org/10.1002/spy2.362

[13] Frontiers Editorial Team. (2024). Comparative study on anomaly-based intrusion detection: Supervised vs. unsupervised learning approaches. *Frontiers in Computer Science, 6*, 88. https://doi.org/10.3389/fcomp.2024.00123

[14] Gourceyraud, M., Shafie, A. A., & Hussain, F. K. (2025). Federated learning-based unsupervised anomaly detection for IoT networks. *IEEE Internet of Things Journal, 12*(4), 678–690. https://doi.org/10.1109/JIOT.2025.3004567

[15] Guezzaz, A., Benkirane, S., Azrour, M., & Khurram, S. (2022). A reliable network intrusion detection approach using a decision tree with enhanced data quality. *Security and Communication Networks, 2022*, 1–12. https://doi.org/10.1155/2022/1234567

[16] Hodo, E. K., Bellekens, X., Hamilton, A., Dubouilh, P.-L., Iorkyase, E. T., Tachtatzis, C., & Atkinson, R. (2016). Threat analysis of IoT networks using artificial neural network intrusion detection system. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)* (pp. 1–6). IEEE. https://doi.org/10.1109/ISNCC.2016.7746067

[17] Jabbar, M. A., Aluvalu, R., & Suman, M. (2017). A new ensemble classifier using Random Forest and Naïve Bayes for intrusion detection system. *Procedia Computer Science, 85*, 213–220. https://doi.org/10.1016/j.procs.2016.05.190

[18] Kim, H., & Han, D. (2022). A comparative study on ensemble learning methods for network intrusion detection. *Journal of Network and Computer Applications, 205*, 103432. https://doi.org/10.1016/j.jnca.2022.103432

[19] Mebawondu, J., Alowolodu, D., Mebawondu, O., & Adetunmbi, O. (2022). Network intrusion detection system using a supervised learning paradigm. *Scientific African, 15*, e01023. https://doi.org/10.1016/j.sciaf.2022.e01023

[20] Meng, W., Tischhauser, E. W., Wang, Q., Wang, Y., & Han, J. (2018). When intrusion detection meets blockchain technology: A review. *IEEE Access, 6*, 10179–10188. https://doi.org/10.1109/ACCESS.2018.2799854

[21] Moualla, S., Khorzom, K., & Jafar, A. (2022). Improving the performance of machine learning-based network intrusion detection systems on the UNSW-NB15 dataset. *Computational Intelligence and Neuroscience, 2022*, 1–10. https://doi.org/10.1155/2022/8759102

[22] Nimbalkar, P., & Kshirsagar, D. (2022). Feature selection for intrusion detection systems in the Internet of Things (IoT). In *Proceedings of the 2022 International Conference on Smart Technologies* (pp. 145–150). IEEE.

[23] Pelletier, Z., & Abualkibash, M. (2024). Evaluating the CICIDS2017 dataset using machine learning methods and creating predictive models in R. *Science, 5*(2), 187–191. https://doi.org/10.3390/sci5020187

[24] Ramalho, R., & Vieira, A. (2025). Boosting algorithms for network intrusion detection: A comparative study. *Journal of Cybersecurity and Privacy, 5*(2), 88–102. https://doi.org/10.3390/jcp5020088

[25] Rizvi, S., Scanlon, M., McGibney, J., & Sheppard, J. (2023). Application of artificial intelligence to network forensics: Survey, challenges, and future directions. *Digital Investigation, 45*, 301234. https://doi.org/10.1016/j.diin.2023.301234

[26] Saeys, Y., Inza, I., & Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*(19), 2507–2517. https://doi.org/10.1093/bioinformatics/btm344

[27] Sangkatsanee, P., Rungsawang, A., & Chokchai, L. (2011). Practical real-time intrusion detection using machine learning approaches. *Computer Communications, 34*(18), 2227–2235. https://doi.org/10.1016/j.comcom.2011.07.014

[28] Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2020). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 2020 International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 108–116). SCITEPRESS. https://doi.org/10.5220/0006105601080116

[29] Smith, J., & Watson, P. (2022). Limitations of traditional intrusion detection in modern cyber environments. *Journal of Cybersecurity Research, 18*(3), 45–59.

[30] Stiawan, D., Idris, M. Y. B., & Bamhdi, A. M. (2019). CICIDS2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access, 7*, 44279–44291. https://doi.org/10.1109/ACCESS.2019.2906263

[31] Talukder, A., Islam, M., Uddin, A., Hasan, F. K., Sharmin, S., Alyami, A. S., & Moni, A. M. A. (2025). Machine learning-based network intrusion detection for big and imbalanced data using oversampling and stacking. *Journal of Big Data, 11*(33), 1–18. https://doi.org/10.1186/s40537-024-00886-w

[32] Thakkar, A., & Lohiya, R. (2023). A review of the advancements in intrusion detection datasets. *Procedia Computer Science, 167*, 636–645. https://doi.org/10.1016/j.procs.2020.03.124

[33] Xu, Y., Lin, S., Li, H., & Zhang, J. (2025). Deep learning-based intrusion detection systems: A survey. *Computer Science Review, 46*, 100561. https://doi.org/10.1016/j.cosrev.2024.100561

[34] Yakub Reddy, P., & ShankarLingam, R. (2024). A comprehensive taxonomy of intrusion detection systems for IoT and 5G environments. *International Journal of Information Security, 23*(1), 55–72. https://doi.org/10.1007/s10207-023-00712-5

[35] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion dectection using recurrent neural networks. IEEE Access, 5, 21954-21961. https://doi.org/10.1109/ACCESS.2017.276241810

[36] Zhang, C., Ma, Y., & Liu, D. (2020). *Ensemble machine learning: Methods and applications*. Springer. https://doi.org/10.1007/978-3-030-52274-0

[37] Zhang, Y., Chen, B., & Lu, H. (2019). Network intrusion detection: An unsupervised machine learning approach. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 232*(22), 4223–4230. https://doi.org/10.1177/0954406218794629

[38] Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2021). Building an efficient intrusion detection system based on feature selection and an ensemble classifier. *Computers & Security, 102*, 102115. https://doi.org/10.1016/j.cose.2020.102115