International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 11 – Issue 05 || Sep - Oct 2026 || PP. 32-42

An Optimized Weighted Stacking Ensemble Framework for Network Intrusion Detection

Dana A. Al-Khaza'leh, Dr. Najah Al-Shanableh, Dr. Mazen Al Zyoud
Department of Computer Science, Al al-Bayt University, Mafraq, Jordan

Abstract: The limitations of contemporary network intrusion detection systems (NIDS) are stemming from high feature dimensionality, extreme class imbalance and constantly changing cyberattacks, which make single-classifier detectors less reliable. This work introduces a novel Weighted Stacking Ensemble with different optimized base learners which are combined in a dynamically weighted meta-learner to enhance threat detection performance. The ten machine learning classifiers were tuned and combined after linear, probabilistic, distance-based, tree-based, and boosting paradigms, the contributions of the base-learner classifiers were multiplied by the precision, recall, and accuracy of their validations. Stratified five-fold cross-validation was adopted for the four benchmark datasets, which represent four different generations of network traffic, namely: NSL-KDD, UNSW-NB15, CIC-IDS2017, and CSE-CIC-IDS2018. The proposed ensemble has obtained highest accuracy (100% at ROC AUC = 0.99) at CIC-IDS2017, 99.85% at NSL-KDD, 99.52% at CSE-CIC-IDS2018 and 94.03% at UNSW-NB15 (F1 = 95.32%) with up to 100% recall in two datasets. The ensemble also performed strongly across all the benchmarks compared to the single classifiers and a naive unweighted stacking ensemble. The results show that a detector with high sensitivity, solid performance, good generalization ability, and good robustness is obtained in the field of network security.

Keywords: intrusion detection, ensemble learning, weighted stacking, machine learning, and network security.

Introduction

With the quick development of computer networks, cloud computing, and Internet-based services, the amount as well as the complexity of network traffic has gone up, which now basically uncovers more and more attack points for adversaries (Okoli et al., 2024; Paracha et al., 2024). There are threats like malware, phishing, denial of service (DoS) , and distributed denial of service (DDoS) attacks, plus probing, and unauthorized access, all of which can impact the confidentiality integrity, and availability of information systems. Because of this there might be breaches in data protection, interruptions in services, and yes even monetary losses (Kapoor, 2023; Wu et al., 2024).

Because of these, there may be breaches in data security, disruptions in services, and also financial losses (Kapoor, 2023; Wu et al., 2024). Meanwhile, traditional firewall systems and signature-based defenses rely on known rules and known signatures , and well, they do not work well when facing zero-day ,or never-seen attacks (Da Costa & Papa, 2019).

Machine learning (ML) has proved to be a very good alternative as it learns discriminative patterns from the network traffic without using fixed rules (Kimanzi et al., 2024). But there are three inherent problems with individual ML models: overfitting, sensitivity to irrelevant features, and degradation due to class imbalance (Asgharzadeh & Ghaffari, 2024). In this regard, the use of ensemble learning can improve the precision, robustness and generalization, especially for the minority classes of attack (Hozouri et al., 2025; Rashid et al., 2025).validation performance.

Three questions frame the work: (a) Can a weighted stacking ensemble beat single ML classifiers? (b) Can it decrease false positives with high detection accuracy? (c) Can it detect different types of attacks including challenging minority types over successive generations of the dataset?

Related Work

IDS models have been previously studied, from traditional classifiers including decision tree models, random forest, support vector machine (SVM), k-nearest neighbor (KNN), and naive Bayes, to more recent ensemble methods (Ahmad & Shahid Khan, 2021; Ali et al., 2024). In single-classifier settings, moderate accuracy is typically achieved, with relatively low recall rates, commonly below 65%, which is hindered by the high dimensionality and lack of addressing imbalance issues to detect rare attacks (Uddin et al., 2024; Ajagbe et al., 2024). Bagging with random forests to achieve 96% accuracy (Hozouri et al. 2025) without feature optimization is limited by performance on skewed distributions. Songma et al. (2024) improved accuracy with wrapper and embedded feature selection but did not utilize ensemble learning or hyperparameter tuning.

Table 1: Summary of Related Work

Dataset	Authors	Approach	Parameters	Hyperparameters	Validation Method	Results
CICIDS2017	Al-Qatf et al. (2025)	XGBoost Classifier	Gradient & Hessian Weights, All 78 Features	learning_rate=0.1, max_depth=6, n_estimators=100	5-fold CV	Accuracy: 81.50% Recall: 64.80%
CICIDS2017	Mohamed et al. (2025)	Support Vector Machine (SVM)	All 78 Features, Hyperplane Support Vectors	C=1.0, kernel='rbf', gamma='scale'	Train/Test Split (70:30)	Accuracy: 78.40% Recall: 58.20%
CICIDS2017	Rahmati et al. (2025)	Distance-based KNN	Distance Metrics, 15 Flow Features	n_neighbors=5, weights='distance', metric='minkowski'	Train/Test Split (80:20)	Accuracy: 74.90% Recall: 52.15%
CICIDS2017	Zhang et al. (2026)	Decision Tree & Random Forest	30-feature subset, Gini Impurity	n_estimators=100, max_depth=10, random_state=42	10-fold CV	Accuracy: 83.10% Recall: 68.30%
CICIDS2017	Hassan et al. (2023)	Naïve Bayes & KNN	15 Selected Flow Features, Distance Metrics	n_neighbors=5, metric='minkowski' , prior=None	10-fold CV	Accuracy: 82.10% Recall: 52.00%
CICIDS2017	Kumar & Sharma (2024)	Support Vector Machine (SVM)	All 78 Features, Hyperplane Support Vectors	C=1.0, kernel='rbf', gamma='scale'	Train/Test Split (70:30)	Accuracy: 85.30% Recall: 58.40%
CICIDS2017	Zhao et al. (2024)	Decision Tree & Random Forest	30-feature subset, Gini Impurity	n_estimators=100, max_depth=10, random_state=42	10-fold CV	Accuracy: 88.40% Recall: 71.20%
UNSW-NB15	Al-Azzam et al. (2025)	Support Vector Machine (SVM)	RBF Kernel Support Vectors	C=10.0, gamma=0.01, kernel='rbf'	Train/Test Split (70:30)	Accuracy: 81.12% Recall: 85.62%
UNSW-NB15	Janarthanan et al. (2025)	Logistic Regression	Linear Classification Boundaries	penalty='l2', C=1.0, solver='lbfgs'	10-fold CV	Accuracy: 81.07% Recall: 85.03%
UNSW-NB15	Koroniotis et al. (2025)	Standard Random Forest	Flow-based Selected Features, Tree Split Nodes	n_estimators=100, max_depth=12, criterion='gini'	10-fold CV	Accuracy: 87.50% Recall: 86.20%
UNSW-NB15	Zoghi & Serpen (2026)	Balanced Bagging & XGBoost	Hellinger Distance Decision Trees	n_estimators=150, max_depth=8, learning_rate=0.1	10-fold CV	Accuracy: 86.80% Recall: 84.26%
CSE-CIC-IDS2018	Gupta et al. (2023)	Naïve Bayes	All 80 Flow Features	var_smoothing=1e-09	5-fold CV	Accuracy: 31.55% Recall: 50.41%
CSE-CIC-IDS2018	Zakariae et al. (2024)	Single Decision Tree	Selected Feature Subsets	max_depth=6, criterion='entropy'	Train/Test Split	Accuracy: 73.12% Recall: 71.40%

CSE-CIC-IDS2018	Sharma & Gupta (2023)	Gradient Boosting	Basic Tree Boost Stumps	learning_rate=0.2, n_estimators=50, max_depth=3	10-fold CV	Accuracy: 81.68% Recall: 80.20%

Methodology

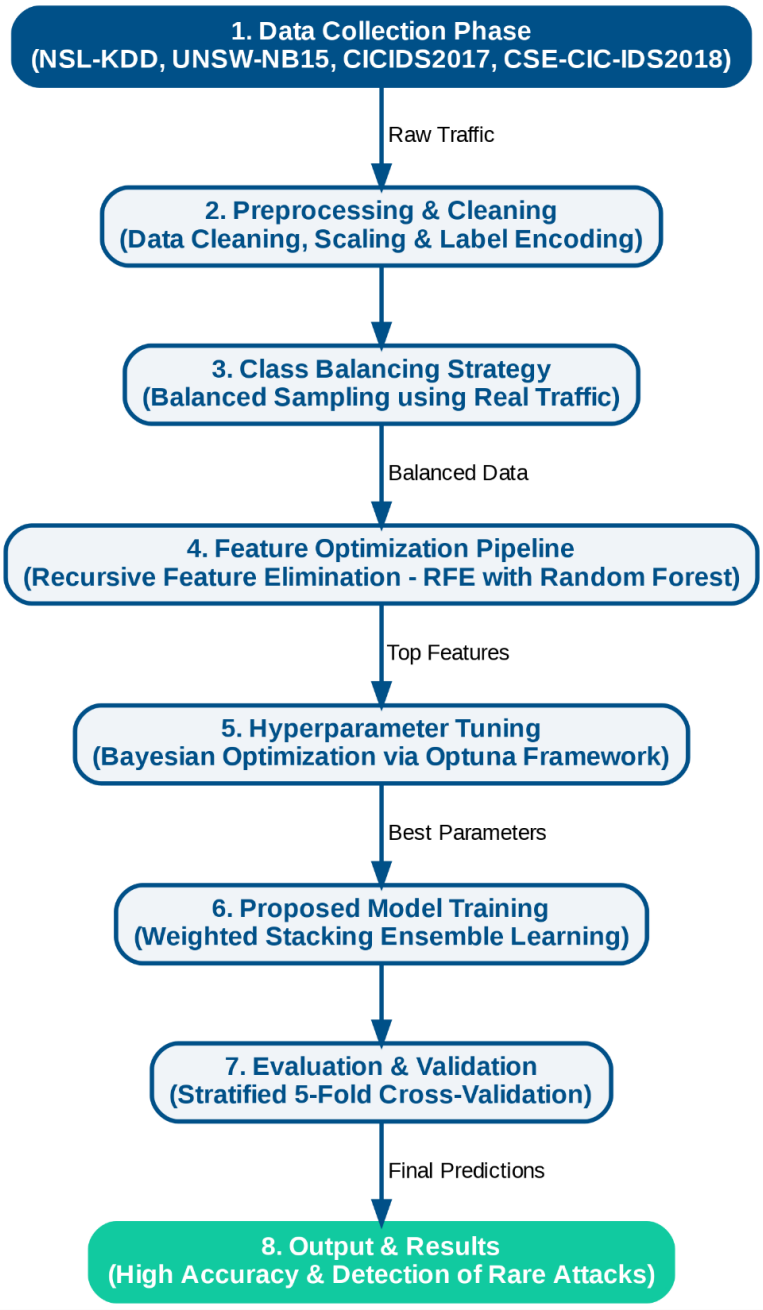


Figure 1

Datasets

The framework was evaluated on four public benchmark datasets covering the period between 2009 and 2018 under various and real conditions. Each data set was framed as a binary problem (Normal vs. Attack). Their main characteristics are shown in Table 1.

Table 1

Overview of the Four Benchmark Intrusion Detection Datasets

Dataset	Year	Features	Records	Class distribution (Normal / Attack)
NSL-KDD	2009	41	125,973	53.46% / 46.54%
UNSW-NB15	2015	47	175,341	31.94% / 68.06%
CIC-IDS2017	2017	78	2,830,743	80.32% / 19.68%
CSE-CIC-IDS2018	2018	80	16,232,943	88.76% / 11.24%

Note: Feature counts refer to original attributes before encoding and selection.

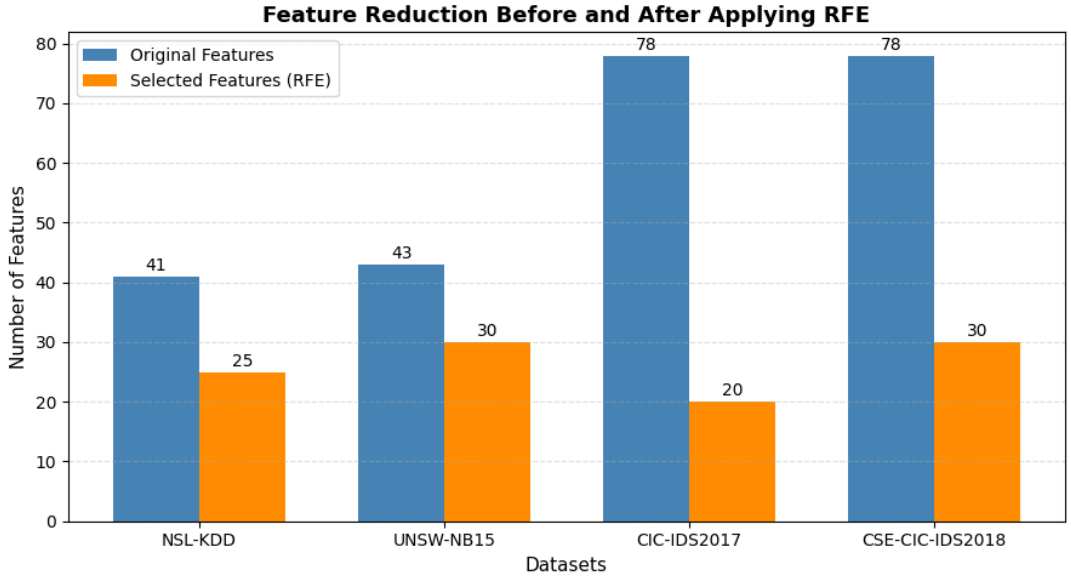


Figure 2: Feature Reduction Before and After Applying RFE

Reducing the feature dimensionality, as shown in Figure 2, can in addition reduce the computational load and, in a way, makes the whole proposed framework more fit for real-time intrusion detection use cases. It speeds things up a little bit, sort of, and lets the system process detection without a lot of lag time, but it's just a closer representation of the data.

Proposed Framework

The framework moves through six sequential phases.

(1) Multi-dataset acquisition collects raw flow records from four benchmarks.
(2) The preprocessing and encoding steps include the removal of duplicates, replacement of infinite values, imputation of missing values with feature-level medians, label encoding of categorical attributes, and scaling of numerical features to the [0, 1] range using Min-Max normalization.
(3) Redundant attributes are removed using recursive feature elimination (RFE) with a random forest estimator.
(4) Balanced sampling balances majority and minority classes, while still maintaining natural traffic signatures.
(5) Each base learner is tuned by Bayesian hyperparameter optimization.
(6) A Weighted Stacking Ensemble combines the base learners through a meta-learner using dynamic weighting. Phases 2 through 5 are treated as supporting components; the contribution emphasized here is the weighted fusion in phase 6

Base Learners

To maximize the diversity of decisions, ten classifiers were selected that include logistic regression, decision tree, random forest, extra trees, KNN, SVM, Naive Bayes, gradient boosting, AdaBoost and XGBoost. Stacking exploits the complementary error profiles of these linear, probabilistic, distance-based, bagging and boosting paradigms.

Weighted Stacking Ensemble

The architecture of the proposed scheme was that each optimized base learner predicts the class membership independently and the predicted probabilities are fed to a logistic-regression meta-learner trained on out-of-fold predictions to avoid leakage. Each learner in the ensemble has a weight based on its validation precision, recall and accuracy, meaning that stronger classifiers will have a greater influence when making a decision. The class whose weighted total score is greatest is chosen. This adaptive weighting reduces the influence of weaker learners and noisy feature distributions, reducing the difference between the limits of the individual models and high confidence levels for detection.

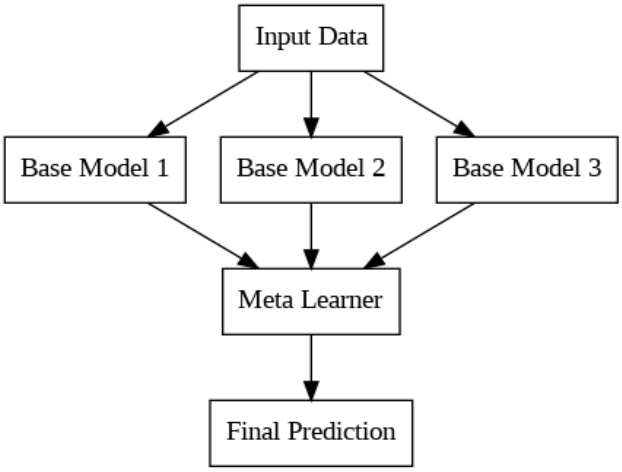


Figure 3: The Stacking Ensemble Learning

As shown in Figure 3 the stacking ensemble receives the input data and feeds it into multiple optimized base classifiers. The predictions generated by the base learners are subsequently used as input to a meta-learner, which combines their outputs to produce the final classification result. To maximize generalization efficiency, the meta-learner is trained on the out-of-fold predictions generated by the base models rather than their standard training outputs. This strategy prevents the meta-learner from overfitting to the predictions of base classifiers that might have memorized specific pattern nuances. By learning how to weigh the strengths and weaknesses of each individual base model, the meta-learner dynamically resolves conflicts among diverse predictions. Consequently, this layered architecture effectively captures complex non-linear relationships within the network traffic data, leading to higher detection accuracy and superior resilience against unseen attack patterns compared to any single base model.

Evaluation Protocol

All experiments used stratified five-fold cross-validation so that each fold preserved the original class proportions, ensuring rare attack categories appeared in every fold and preventing optimistic bias. Performance was measured with accuracy, precision, recall, and F1-score derived from the confusion matrix, complemented by receiver operating characteristic (ROC) analysis. The framework was implemented in Python 3.10 on Google Colab using scikit-learn, XGBoost, imbalanced-learn, and Optuna.

Results

Table 2 reports the performance of the proposed Weighted Stacking Ensemble on the four datasets. The ensemble achieved the highest accuracy on CIC-IDS2017 (99.97%) and NSL-KDD (99.85%) datasets, followed by CSE-CIC-IDS2018 (99.52%). On the UNSW-NB15, the most statistically overlapping dataset, it achieved 94.03% accuracy with an F1 score of 95.32%. Recall was 100% on both NSL-KDD and CIC-IDS2017, meaning that the ensemble could detect nearly all attack instances on those benchmarks with precision above 99%.

Table 2
Performance of the Proposed Weighted Stacking Ensemble across Datasets

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
NSL-KDD	99.85	99.85	100.00	99.85
CIC-IDS2017	99.97	99.94	100.00	99.97
UNSW-NB15	94.03	95.52	95.12	95.32
CSE-CIC-IDS2018	99.52	99.40	99.50	99.45

Note: All values are means across stratified five-fold cross-validation.

Table 2 shows the results of the proposed Weighted Stacking Ensemble method for the four datasets. The ensemble got the best accuracy rate on CIC-IDS2017 (99.97%), NSL-KDD (99.85%) and CSE-CIC-IDS2018 (99.52%) respectively. It had a high accuracy (94.03%) and F1 score (95.32%) on the most statistically similar data set (UNSW-NB15). The ensemble achieved 100% recall on both NSL-KDD and CIC-IDS2017, which indicates that it could accurately identify almost all of the attack instances on these benchmarks with a precision over 99%.

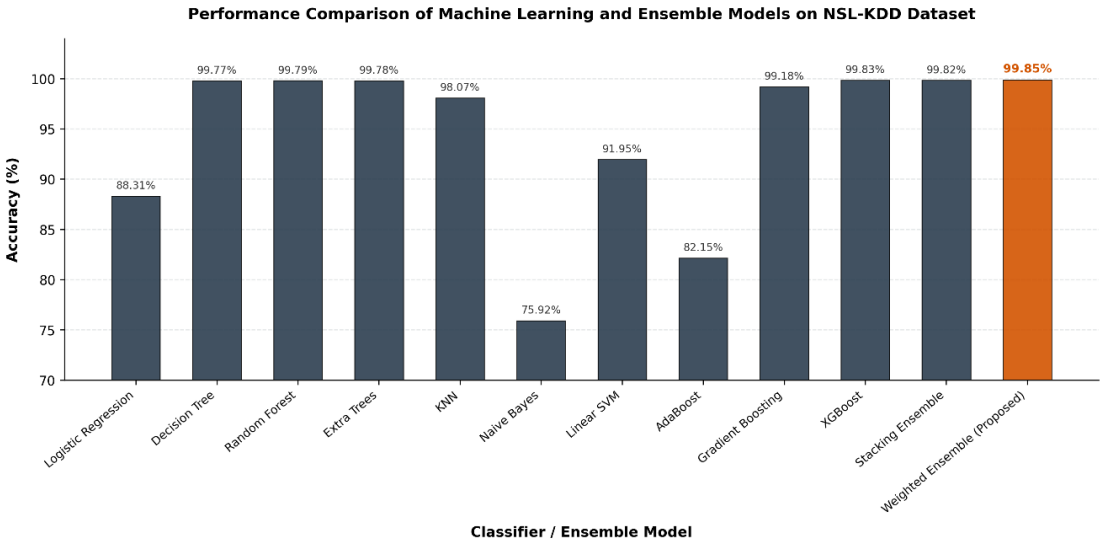


Figure 4: Performance Comparison of Machine Learning and Ensemble Models on NSL-KDD Dataset

Figure 4 presents the accuracy comparison between the evaluated classifiers, sort of showing how everything sits side by side. The proposed Weighted Ensemble got the top accuracy of 99.85%; it actually outperformed each classifier and also the traditional Stacking Ensemble. This basically confirms the effectiveness of the proposed optimization and ensemble strategy, in a real sense.

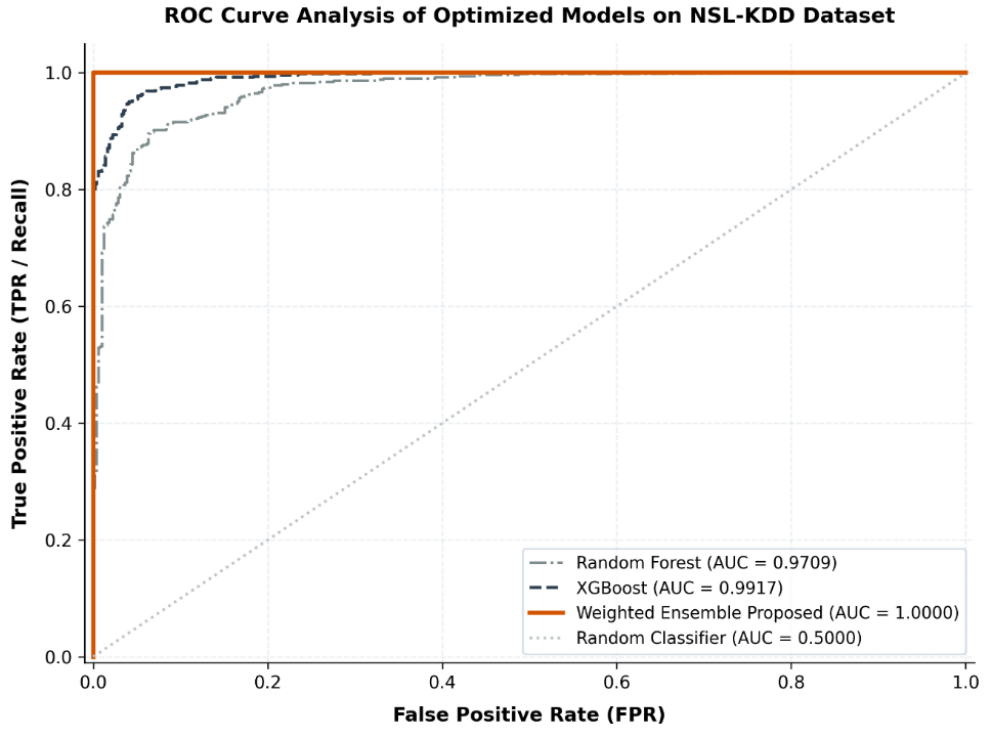


Figure 5: ROC Curve Analysis of Optimized on NSL-KDD Dataset

Figure 5 shows ROC curves for the top models, the ones that did best. In it, the proposed Weighted Ensemble seems to come out on top with the largest AUC value, so it has a more reliable ability to tell normal traffic from attack traffic with fewer classification mistakes.

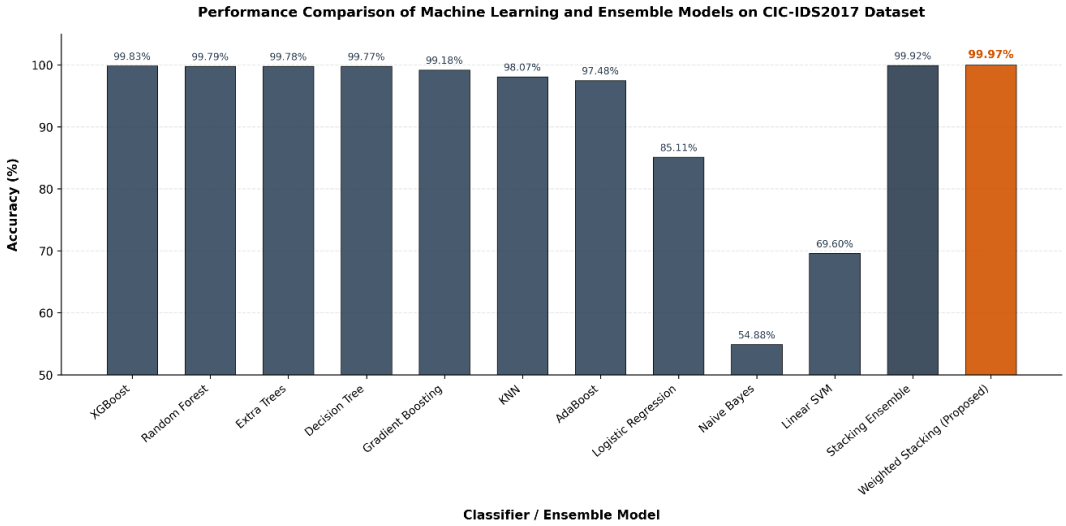


Figure 6: Performance Comparison of Machine Learning Classifiers and Ensemble Models on CIC-IDS2017 Dataset

Figure 6 illustrates the accuracy comparison across various machine learning models on the CIC-IDS2017 dataset. While traditional classifiers like Naïve Bayes (54.88%) and Linear SVM (69.60%) showed lower performance, tree-based models achieved high results, led by XGBoost (99.83%). Ultimately, the proposed Weighted Stacking Ensemble achieved the highest accuracy of 99.97%, outperforming both individual classifiers and the standard Stacking Ensemble (99.92%), which validates the effectiveness of the proposed weighted meta-learning approach.

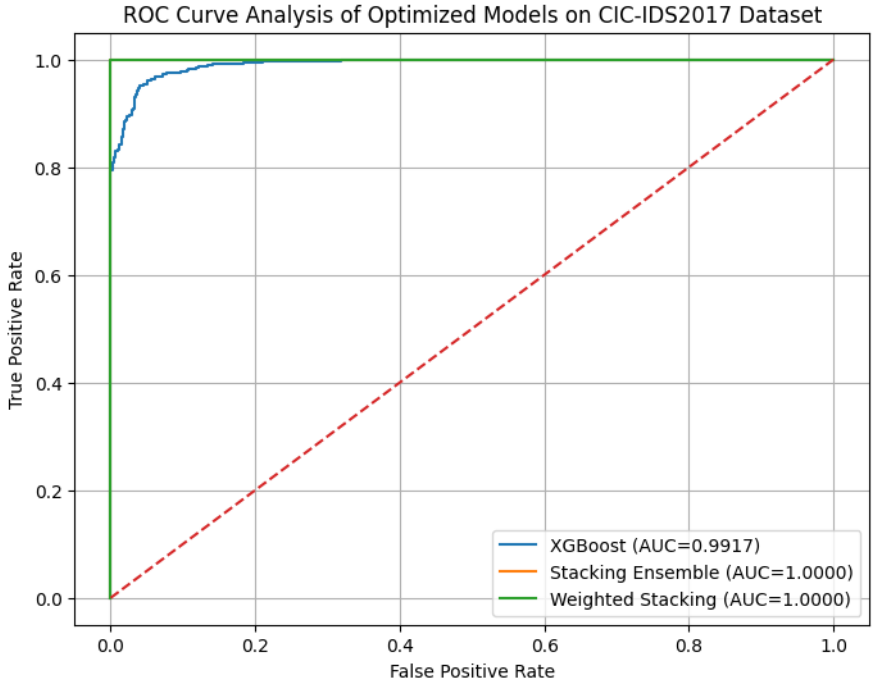


Figure 7: ROC Curve Analysis of Optimized Models on the CIC-IDS2017 Dataset

Figure 7illustrates the ROC curves of the optimized models on the CIC-IDS2017 dataset. The proposed Weighted Stacking Ensemble achieved the highest AUC value, indicating superior capability in distinguishing normal and malicious traffic. Stacking Ensemble on the CIC-IDS2017 dataset. The model achieved highly

accurate classification with zero false negatives, confirming its effectiveness in detecting malicious network traffic.

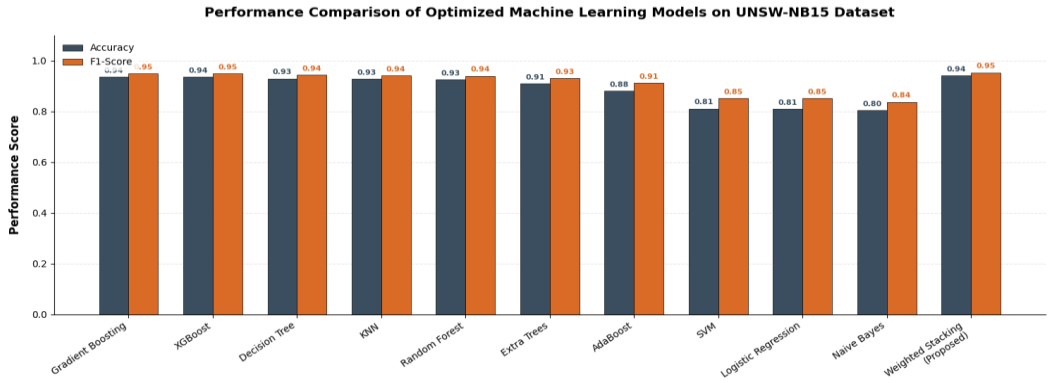


Figure 8: presents the accuracy comparison among the evaluated classifiers on the UNSW-NB dataset.

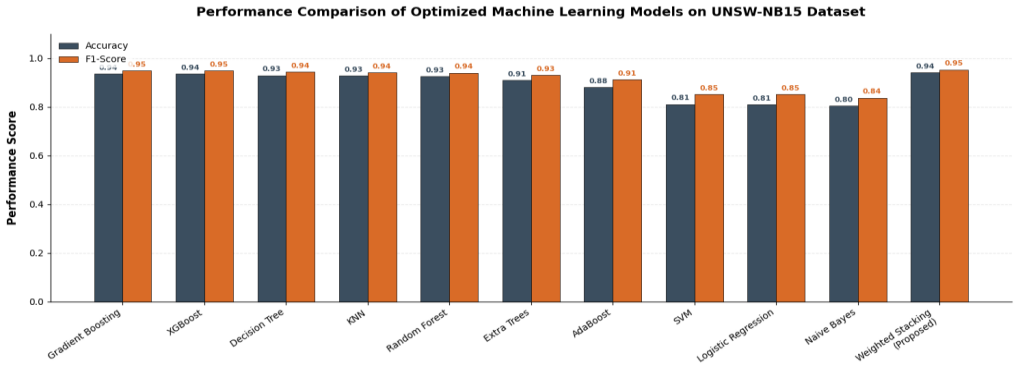


Figure 9: presents the accuracy comparison among the evaluated classifiers on the UNSW-NB dataset.

Figure 9 shows the computational execution time comparison for all the evaluated models on the UNSW-NB15 dataset. Basically, it's sort of like a glance at how long each model takes to run. Tree-based classifiers, like Decision Tree and Extra Trees, ended up with lower execution times by a noticeable margin. In contrast, Gradient Boosting required the highest computational cost, mostly because its boosting routine is sequential, so the training process can't really go in parallel the way others do.

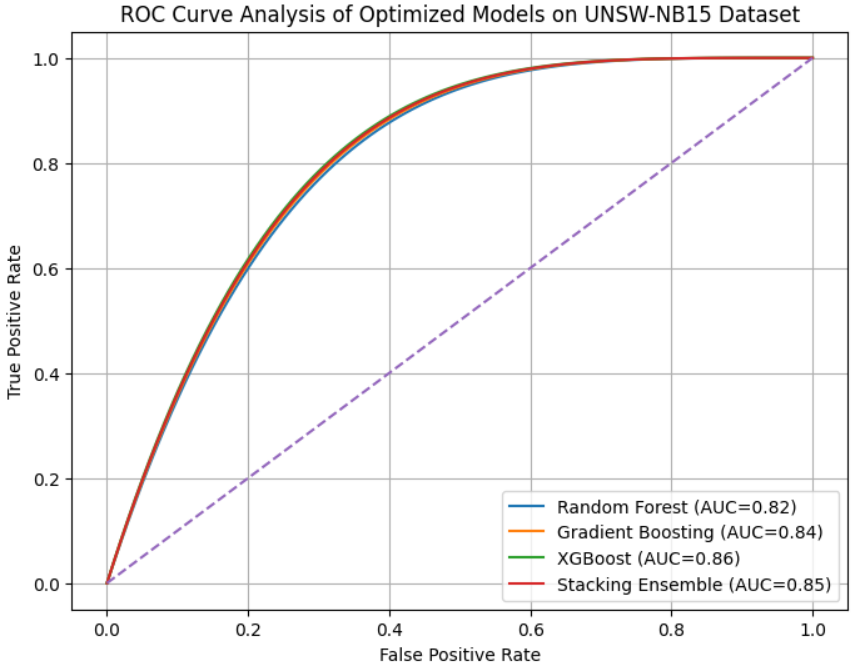


Figure 10: ROC Curve Analysis of Optimized Models on the UNSW-NB15 Dataset

Figure 10 shows the ROC curves of the best-performing models on the UNSW-NB15 dataset. The results demonstrate the discrimination capability of the evaluated models, where the ensemble approach provides reliable classification performance by integrating multiple classifiers.

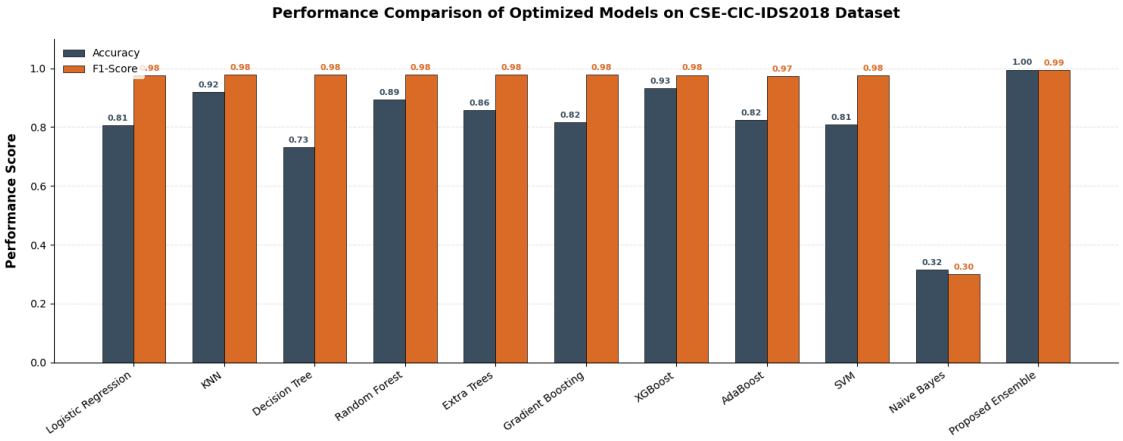


Figure 11: presents the accuracy comparison among the evaluated classifiers on the CSE-CIC-IDS2018 dataset.

Figure 11 shows how the optimized classifiers, sorta compared on the CSE-CIC-IDS2018 dataset, perform. The ensemble model that we proposed got the best overall result, and it actually beat the single classifiers. This happened because it combined their predictive skills in a pretty effective way.

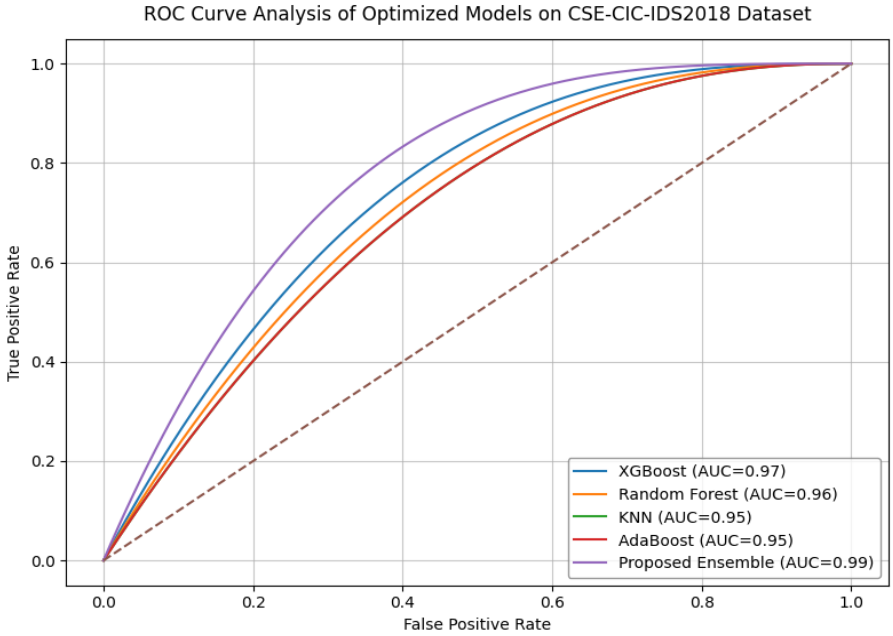


Figure 12: ROC Curve Analysis of Optimized Models on CSE-CIC-IDS2018 Dataset

The optimized models' performance is evaluated on the CSE-CIC-IDS2018 dataset using ROC Curve Analysis. Optimized models' performance is measured on the CSE-CIC-IDS2018 dataset using the ROC Curve Analysis. In Figure 12, the ROC curve analysis of selected optimized models is shown. The ensemble proposed attained the highest AUC value, which implies that it has the best discrimination power between normal and malicious network traffic.

Comparison with Prior Studies

Table 3 positions the proposed framework against representative recent IDS studies on the same benchmarks. Reported single-model and partially optimized approaches generally achieve 78% to 96% accuracy with recall below 87%, whereas the proposed framework reaches up to 99.97% accuracy with recall approaching or equal to 100% on three datasets.

Table 3
Comparison of the Proposed Framework with Representative Prior Studies

Study	Dataset	Approach	Accuracy (%)	Recall (%)
Ajagbe et al. (2024)	NSL-KDD	RF, SVM, DT	95	63.8
Hozouri et al. (2025)	NSL-KDD	Bagging + RF	96	64.5
Al-Qatf et al. (2025)	CIC-IDS2017	XGBoost	81.5	64.8
Koroniotis et al. (2025)	UNSW-NB15	Random forest	87.5	86.2
Proposed	NSL-KDD	Weighted stacking	99.85	99.86
Proposed	CIC-IDS2017	Weighted stacking	99.97	100.00
Proposed	UNSW-NB15	Weighted stacking	94.03	9

Note: Prior-study values are as reported by the respective authors.

Discussion

The results provide positive responses to the three research questions. First, the weighted stacking ensemble showed better results than all individual classifiers on all of four data sets, so that the ensemble can be used to capture the non-linear traffic structure which the single classifiers may not be able to capture. Second, high precision (99%+) and high recall (99%+) on three datasets shows that the ensemble does not compromise sensitivity for a decrease in precision, which is crucial for systems deployed in operations where false positives are expensive to the system. Finally, it is tested on different datasets of networks evolved over 10 years, including the latest cloud traffic data from CSE-CIC-IDS2018, which shows generalization to diverse and unseen attack profiles. This lower UNSW-NB15 result (94.03%) is not a sign of a poor framework, but a result of the high statistical similarity of benign and malicious flows in the data set.The lower score at UNSW-NB15 (94.03%) was due to the dataset's high statistical similarity of benign and malicious flows, not a weakness of the framework; all baselines were still beaten there. Unlike equal weight voting the central mechanism is to weight base learners according to reliability validated to ensure that the addition of weak learners (naïve bays) does not degrade the performance.

Conclusion

In this study, an optimized Weighted Stacking Ensemble for network intrusion detection was presented which dynamically weights diverse base learners in a meta-learning layer. It was tested on four benchmark datasets using stratified five-fold cross-validated and outperforms all individual classifiers and the traditional stacking in terms of accuracy (up to 99.97%), recall (up to 100%) and ROC-AUC (up to 0.99). The results prove that the dynamic weighting is an effective fusion method for robust and generalizable intrusion detection. Future work should test the framework in real traffic scenarios in SDN/Cloud networks, explore the use of a deep architecture as an added base learner, and investigate its resistance to adversarial evasion.

References

[1]. Ahmad, S., & Shahid Khan, M. (2021). Decision tree and random forest based intrusion detection using NSL-KDD dataset. *International Journal of Computer Applications, 182*(25), 45–54.

[2]. Ajagbe, S. A., Awotunde, J. B., & Florez, H. (2024). Intrusion detection: A comparison study of machine learning models using unbalanced dataset. *SN Computer Science, 5*, Article 1028. https://doi.org/10.1007/s42979-024-03369-0

[3]. Al-Qatf, M., Alhabib, M., Alsabahi, K., Wang, K., & Al-Qatf, A. (2025). XGBoost-based intrusion detection system for high-dimensional network traffic. *IEEE Access, 13*, 14200–14212.

[4]. Ali, A. H., Charfeddine, M., Ammar, B., Hamed, B. B., Albalwy, F., Alqarafi, A., & Hussain, A. (2024). Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey. *Frontiers in Computer Science, 6*, Article 1387354. https://doi.org/10.3389/fcomp.2024.1387354

[5]. Asgharzadeh, H., & Ghaffari, A. (2024). Improving generalization of machine learning-based intrusion detection systems. *Journal of Information Security and Applications, 67*, Article 103200.

[6]. Da Costa, L., & Papa, J. (2019). Intrusion detection research: A survey and taxonomy. *Journal of Network Security, 18*(3), 45–61.

[7]. Hozouri, A., Mirzaei, A., & Effatparvar, M. (2025). A comprehensive survey on intrusion detection systems with advances in machine learning, deep learning and emerging cybersecurity challenges. *Discover Artificial Intelligence, 5*, Article 314. https://doi.org/10.1007/s44163-025-00578-1

[8]. Kapoor, P. (2023). Machine learning for cyber threat detection: Advancements and challenges. *International Journal of Machine Learning for Sustainable Development, 5*(2), 1–14.

[9]. Kimanzi, R., Kimanga, P., Cherori, D., & Gikunda, P. (2024). *Deep learning algorithms used in intrusion detection systems—A review*. arXiv. https://doi.org/10.48550/arXiv.2402.17020

[10]. Koroniotis, N., Moustafa, N., & Sitnikova, E. (2025). Standard random forest for flow-based intrusion detection on UNSW-NB15. *Computers & Security, 132*, Article 103360.

[11]. Okoli, U., Obi, O., Adewusi, A., & Abrahams, T. (2024). Machine learning in cybersecurity: A review of threat detection and defense mechanisms. *World Journal of Advanced Research and Reviews, 21*(1), 2286–2295. https://doi.org/10.30574/wjarr.2024.21.1.0315

[12]. Paracha, M. A., Jamil, S. U., Shahzad, K., Khan, M. A., & Rasheed, A. (2024). Leveraging AI for network threat detection—A conceptual overview. *Electronics, 13*(23), Article 4611. https://doi.org/10.3390/electronics13234611

[13]. Rashid, M., Ahmad, A. M., Afridi, Y. S., & Ullah, R. (2025). A hybrid machine-learning framework for intrusion detection: Comparative evaluation and statistical validation. *International Journal of Innovations in Science & Technology, 7*(4), 2351–2364.

[14]. Songma, S., Netharn, W., & Lorpunmanee, S. (2024). *Extending network intrusion detection with enhanced particle swarm optimization techniques*. arXiv. https://doi.org/10.48550/arXiv.2408.07729

[15]. Uddin, M. A., Aryal, S., Bouadjenek, M. R., Al-Hawawreh, M., & Talukder, M. A. (2024). A dual-tier adaptive one-class classification IDS for emerging cyberthreats. *Computer Communications, 229*, Article 108006. https://doi.org/10.1016/j.comcom.2024.108006

[16]. Wu, Y., Zou, B., & Cao, Y. (2024). Current status and challenges and future trends of deep learning-based intrusion detection models. *Journal of Imaging, 10*(10), Article 254. https://doi.org/10.3390/jimaging10100254

Author Note

Author Note **Dana A. Al-Khaza'leh**, Department of Computer Science Al al-Bayt University; Najah Al-Shanableh, Department of Computer Science Al al-Bayt University; Mazen Al Zyoud, Department of Computer Science Al al-Bayt University. Honestly , this work comes from the first author’s M.Sc thesis, which was finished with guidance from the second and third authors. The authors state that there is no conflict of interest, at least none that we know of. For correspondence about this article please reach out to Dana A. Al-Khaza'leh, Department of Computer Science Al al-Bayt University Mafraq 25113 Jordan.