International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I5P130

Volume 74 Issue 5, 484-494, May 2026

Original Article

Two-Stage Ensemble Machine Learning for Network Intrusion Detection

Jimson A. Olaybar[1], Patrick D. Cerna[2]

[1,2]*College of ICT and Engineering, State University of Northern Negros, Sagay City, Negros Occidental, Philippines*

[1]*Corresponding Author : jolaybar@southernleytestateu.edu.ph*

Received: 17 December 2025 Revised: 13 January 2026 Accepted: 11 March 2026 Published: 30 May 2026

Abstract *- This paper introduces a two-stage ensemble machine learning architecture of network Intrusion Detection Systems (IDS), which is developed to improve the accuracy of detection and reliability of classification in a more complicated cyberspace. The proposed model operates in two phases: Stage A involves binary classification to differentiate between benign traffic and malicious activity with the help of a calibrated stacking ensemble of Random Forest, Gradient Boosting, and XGBoost classifiers; Stage B involves the use of a multi-class attack categorization through a Random Forest classifier that will be trained only on attack samples. The CIC-IDS2017 dataset was used to evaluate the system and includes more than 2.8 million records of network traffic, with varied attack scenarios. Preprocessing involved normalization of features, filling in of missing values, and screening of 78 flow-based numerical features. As a result of the experiments, the two-stage ensemble obtained 99.92% accuracy in binary classification and 99.83% accuracy in multi-class classification on 14 types of attacks. The model scored close to the optimum ROC-AUC (0.99987) and was able to reduce the bias of class imbalance using probability estimation and threshold optimization. It was compared and found that the proposed system was superior to the existing methods of ensemble and deep learning techniques in accuracy and computation efficiency. The results prompt the future prospects of multi-level ensemble learning to enhance the performance of IDS with regard to modern network infrastructures. The further developments in the field will focus on adaptive learning to unknown threats, and implementation together with real-time network defenses.*

Keywords *- Ensemble Learning, Intrusion Detection, Machine Learning, Network Security, Random Forest.*

1. Introduction

The growing complexity of cyber threats and their frequency have made Intrusion Detection Systems (IDS) an essential part of ensuring the integrity of network security. It is described in the past literature that conventional signature-based IDS is often unable to identify new attacks or attacks that have evolved, because they rely on predefined patterns and, as a result, do not identify the zero-day attacks [11]. In a bid to curb these inadequacies, scholars have embraced machine learning-based anomaly detection algorithms that have the potential of identifying new forms of attacks never witnessed before [19, 21]. However, most existing machine learning methods have class imbalance challenges, probability miscalibration, and optimal threshold issues, thus affecting the detectability of an object in a real-world setting [8, 10].

Benchmark datasets have been effective in the evaluation of the efficacy of IDS. The first sets, like KDD CUP 99 [3] and UNSW-NB15 [2], were also key but were prone to redundancy as well as obsolescence of attack types. The CIC-IDS2017 dataset [1] ameliorated these shortcomings by offering a realistic traffic alongside modern attack models, such as the DoS attacks, DDoS attacks, and PortScan, among others [12]. Random Forest [4], Gradient Boosting [5], and XGBoost [6] are examples of ensemble learning strategies that have been shown to be more effective when used in the context of IDS due to the ability to combine many classifiers to reduce bias and variance. [16, 29] empirically demonstrated that a combination of tree-based models improves the detection rates, while [22, 28] highlighted the high accuracy of ensemble recognition compared to single classifiers.

Based on these findings, the current research sets out an ensemble learning model that is applied in two phases to optimize binary detection and multi-class attack recognition by using calibrated stacking and optimized threshold tuning. The desired result is to attain high detection rates and resistance to skewed network traffic [24], which will overcome salient constraints of current IDS architectures.

Although machine learning has been widely adopted for intrusion detection [26], a large portion of prior work employs single-stage classification frameworks that handle both binary and multi-class detection within a unified model. Such designs often suffer from task interference, probability miscalibration, and degraded performance under highly imbalanced traffic

conditions. In particular, prior ensemble and deep learning-based IDS models tend to optimize accuracy alone [20], without sufficient consideration of calibrated decision thresholds or class-wise detection reliability.

Dynamic Time Warping (DTW) clustering captured four different user behavior patterns, representing idle times, intermediate traffic, and intense bursts of usage [31]. Network traffic behaviors are visualized using scatter plots, clustering maps, and seasonal decomposition graphs, which give actionable insights to network administrators.

This study addresses these limitations by introducing a two-stage ensemble intrusion detection framework that separates coarse-grained detection from fine-grained attack classification. Unlike prior approaches, the proposed model integrates probability calibration and threshold optimization into a stacking ensemble for binary detection, followed by a dedicated multi-class classifier trained exclusively on malicious traffic.

This design reduces class interference, improves detection stability under imbalance, and enhances interpretability, thereby contributing to a practical and scalable IDS architecture suitable for real-world deployment.

Recent studies have explored ensemble learning and deep learning approaches to improve intrusion detection accuracy. [30] demonstrated that combining tree-based classifiers can enhance detection rates; however, their approach relied on a single-stage architecture, which may limit scalability in imbalanced environments. [17, 18] employed deep learning models that achieved competitive performance but required extensive computational resources and large labeled datasets.

Several works have addressed class imbalance using resampling strategies or cost-sensitive learning [16], yet these methods may distort data distributions or introduce bias. Moreover, limited attention has been given to probability calibration and threshold optimization, despite their importance in security-critical applications [8, 10].

In contrast to existing methods, the present study adopts a two-stage detection strategy that explicitly separates binary detection from multi-class attack classification. By incorporating calibrated probability estimates and optimized thresholds, the proposed framework addresses both imbalance and misclassification risks more effectively than prior single-stage ensemble or deep learning-based IDS models.

This work contributes to intrusion detection research in the following ways:

- A two-stage intrusion detection architecture that separates binary detection from multi-class attack classification to reduce task interference.
- A calibrated stacking ensemble with optimized decision thresholds to improve detection reliability under class-imbalanced traffic conditions.
- Experiments conducted on the CIC-IDS2017 dataset confirm that the two-stage ensemble framework maintains strong detection accuracy while operating with low computational overhead across diverse attack scenarios.

1.1. Objectives of the Study

The objective of this study is to develop and evaluate a two-stage ensemble-based Intrusion Detection System (IDS) that enhances detection accuracy, probability calibration, and attack categorization performance.

Specifically, the study seeks to design a two-stage IDS framework in which Stage A performs binary classification to distinguish benign from malicious traffic, while Stage B conducts multi-class classification to identify specific attack types; implement a calibrated stacking ensemble that integrates Random Forest, Gradient Boosting, and XGBoost classifiers for reliable probability estimation; conduct a comprehensive performance assessment using the CIC-IDS2017 dataset with emphasis on precision, recall, F1-score, and ROC-AUC; and compare the proposed model with existing IDS approaches to demonstrate its effectiveness and efficiency in real-world cybersecurity contexts.

2. Materials and Methods

This paper used a well-organized experiment procedure and covered data preprocessing, feature selection, model development, hyperparameter optimization, and performance assessment.

The design of the methodology was strictly tailored in order to incorporate reproducibility, strength, as well as a fair comparison to extant Intrusion Detection System (IDS) techniques.

2.1. Experimental Setup

All model training and testing procedures were executed on a workstation with the following hardware and software specifications:

- Processor: Intel® Core™ i5
- Memory: 20 GB RAM
- Operating System: Windows 11
- Development Environment: Python 3.10
- Libraries: scikit-learn 1.3, xgboost 1.7, numpy 1.24, pandas 2.0

This environment ensured stable training, efficient inference, and full reproducibility of the results.

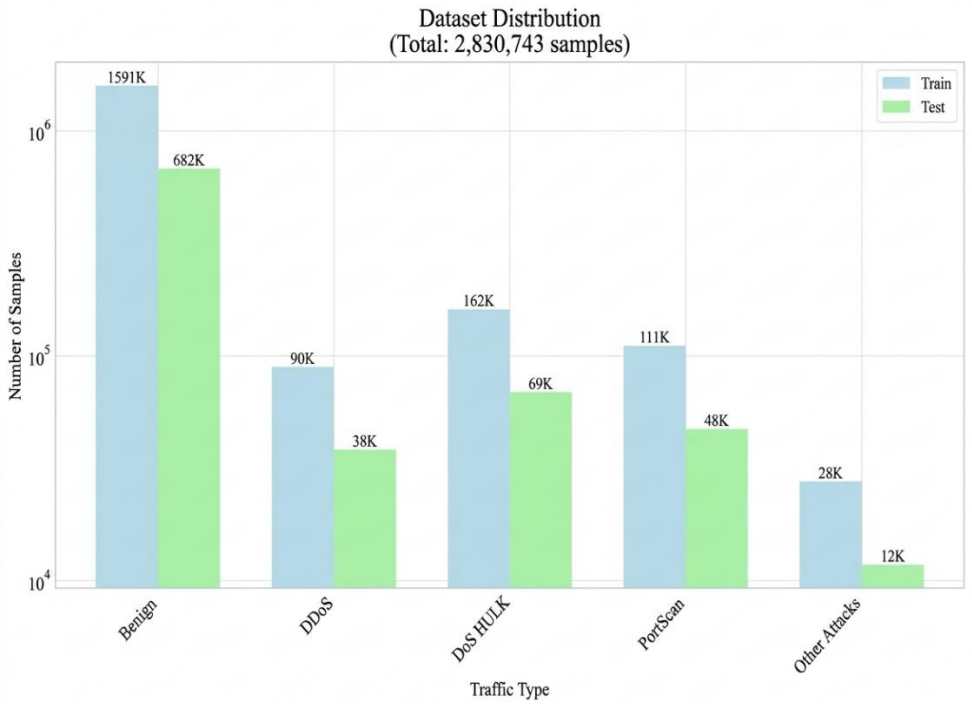

Fig. 1 Dataset distribution across traffic types for training and testing sets (CIC-IDS2017)

2.2. Data Sources and Preprocessing

The experiment utilized the CIC-IDS2017 dataset, which is a realistic network traffic of benign and malicious flows. The dataset has 2.8 million instances with 80 features and analyzes the diverse attack situations across several days. This study, as per the dataset characterization conducted, focused on network-layer attacks, which can be identified by flow statistics. [25] emphasized the importance of generating realistic data using the fuzzy qualitative model, which also supports our choice of the CIC-IDS2017 dataset to ensure the applicability in the real world.

There were a number of systemized steps in data preprocessing. First, the names of columns were normalized, and any missing value was filled with a 0. StandardScaler was used to standardize the numerical attributes, but the non-numeric identifiers were removed to prevent data leakage. It was done by Unicode normalization to the label column to ensure uniformity in the different nomenclatures of attacks.

2.3. Feature Selection

Once identifier-based fields (IP addresses, ports, and timestamps) were removed, 78 numerical flow features remained. These features represent statistical summaries of network flow behavior, including packet rate, byte rate, inter-arrival time, and header-related measurements that are commonly associated with anomalous traffic patterns. Such flow-based characteristics are independent of host-specific identifiers and reduce the risk of information leakage. This feature selection strategy aligns with hybrid intrusion detection studies reported in [13], which emphasize that statistical flow features significantly improve the effectiveness of multi-stage detection architectures. Feature normalization using StandardScaler was applied to ensure uniform feature scaling, which is essential for calibrated probability estimation and meta-classifier convergence. The removal of identifier-based attributes prevented information leakage and ensured that classification relied solely on behavioral traffic characteristics.

2.4. Model Architecture

Random Forest, Gradient Boosting, and XGBoost were incorporated to exploit their distinct learning behaviors within the proposed ensemble. Random Forest contributes stable performance under noisy traffic conditions, Gradient Boosting enhances discrimination through iterative error correction, and XGBoost offers efficient learning for high-dimensional flow-based features. Their integration through stacking enables the model to leverage diverse decision boundaries while minimizing overfitting.

2.4.1. Class Imbalance Handling

The CIC-IDS2017 dataset contains strong imbalances in the traffic distributions, as there are attack modalities that are represented significantly more than others. The research used a class_weight of balanced to tree models to maintain data integrity and avoid distortion caused by the oversampling effect or undersampling to enhance the detection of minority classes, which was accompanied by probability calibration and threshold optimization. This methodology follows the

guidelines of [13, 15] on how to deal with the imbalance of a hybrid IDS architecture.

2.4.2. Stage A: Binary Classification (Benign vs Attack)

Random Forests, Gradient Boosting, and XGBoost were stacked to form a stacking ensemble. The ensemble used a meta-classifier that was logistic regression, which fit the framework of stacked generalization [7]. In order to enhance probability calibration, Calibrated Classifier CV with a sigmoid calibration scheme has been included [8, 9], which has proven to be effective in Cybersecurity [14].

$$P(y = 1|x) = \sigma\left(\sum_{i=1}^{N} w_i f_i(x)\right)$$

Equation (1) Binary Classifier

where $f_i(x)$represents base classifier predictions, $w_i$are ensemble weights, and σ is the sigmoid function for calibration.

2.4.3. Stage B: Multi-class Classification

To classify attacks of such type, a Random Forest classifier was trained only on malicious samples. This design also eliminates the possibility of label leakage, and it makes the following multi-class classifier less likely to confuse the differences between attack patterns and benign traffic.

2.4.4. Hyperparameter Optimization

All models had grid search hyperparameters tuned with stratified validation, thus ensuring equal participation of minority attacks in the tuned models. The important optimized parameters were:

- Random Forest:
  - n_estimators = 300
  - max_depth = 20
- Gradient Boosting:
  - n_estimators = 200
  - learning_rate = 0.1
- XGBoost:
  - max_depth = 6
  - eta = 0.3
  - subsample = 0.8
- Logistic Regression (meta-classifier):
  - C = 1.0

These settings achieved the best trade-off between accuracy, calibration, and computational efficiency.

2.4.5. Threshold Optimization

The study used the method of optimizing the threshold using the F1-score as [10].

$$\tau^* = \arg\max_{\tau} F1(\tau) = \arg\max_{\tau} \frac{2 \cdot P(\tau) \cdot R(\tau)}{P(\tau) + R(\tau)}$$

Equation (2) Threshold Optimization

where $P(\tau)$and $R(\tau)$represent precision and recall at threshold τ.

In order to determine the best decision boundary of Stage A, a precision-recall curve analysis was undertaken. The model chose an optimized probability threshold of 0.256 instead of the default probability threshold of 0.50, as using this would yield the best precision-recall balance.

This modification greatly enhanced the identification of minority classes of attacks, while preserving the same time preserving high accuracy of benign traffic.

The optimized threshold was used to make sure that the binary classifier was operating at a position giving maximum detection reliability, especially in intrusion cases of imbalanced conditions.

2.4.6. Cross-Validation

The study used stratified 5-fold cross-validation in the hyperparameter tuning in order to confirm the stability of the model and to avoid overfitting. Stratification was done to make sure every fold had equal representation of all types of attacks, which enhanced the reliability of generalization.

2.5. System Workflow

The workflow starts with preprocessing, whereby raw network flows are post-processed, made standard, and converted into numerical feature vectors. Such vectors are subsequently forwarded to Stage A, which does binary classification to ascertain whether the traffic is benign or malicious.

In case the flow is determined as malicious, it is sent to Stage B, where a purpose-specific multi-class classifier is used to assign it to one of the 14 attack categories. The sequential design minimizes the computing cost, does not allow benign traffic to affect the decision boundary of multiple classes, and facilitates more precise and fine-tuned detection of attacks.

2.6. Ethical Considerations

The dataset employed in this research was the publicly available CIC-IDS2017 dataset that was gathered and shared by the Canadian Institute of Cybersecurity (CIC) to be used in academic research. There is no empirical data in the dataset that would provide any Personally Identifiable Information (PII) or sensitive user data, which meets the ethical and privacy requirements of research. The analyses were done in compliance with guidelines on data use at an institutional level.

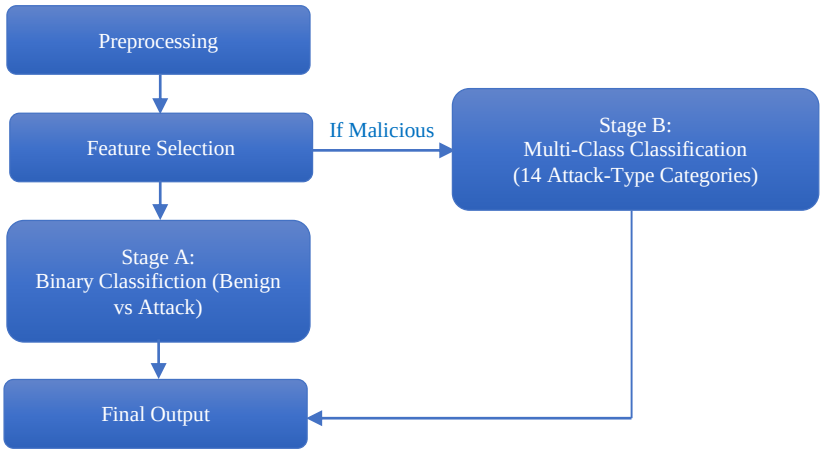

Fig. 2 IDS workflow for two-stage architecture

3. Results and Discussion

3.1. Stage A: Binary Classification Performance

The stacking ensemble showed excellent performance in discriminating between benign and attack traffic. With well-balanced precision, recall, and F1 scores in both classes, as indicated in Table 1, the system had an accuracy of 99.92%, and this indicates a stable and consistent behavior in detection. It is further supported by the ROC-AUC of 0.99987, which indicates that the model is highly robust in separating benign and malicious traffic and is capable of identifying even minor tendencies of intrusion. The performance level shows that the ensemble is able to reflect the underlying statistical properties of network flows and be robust to various traffic conditions. The use of stratified k-fold cross-validation ensured that the model did not overfit dominant traffic classes and maintained stable performance across validation folds. The consistency of precision, recall, and F1-score values across both detection stages indicates balanced learning behavior and confirms that the model did not exhibit underfitting or overfitting.

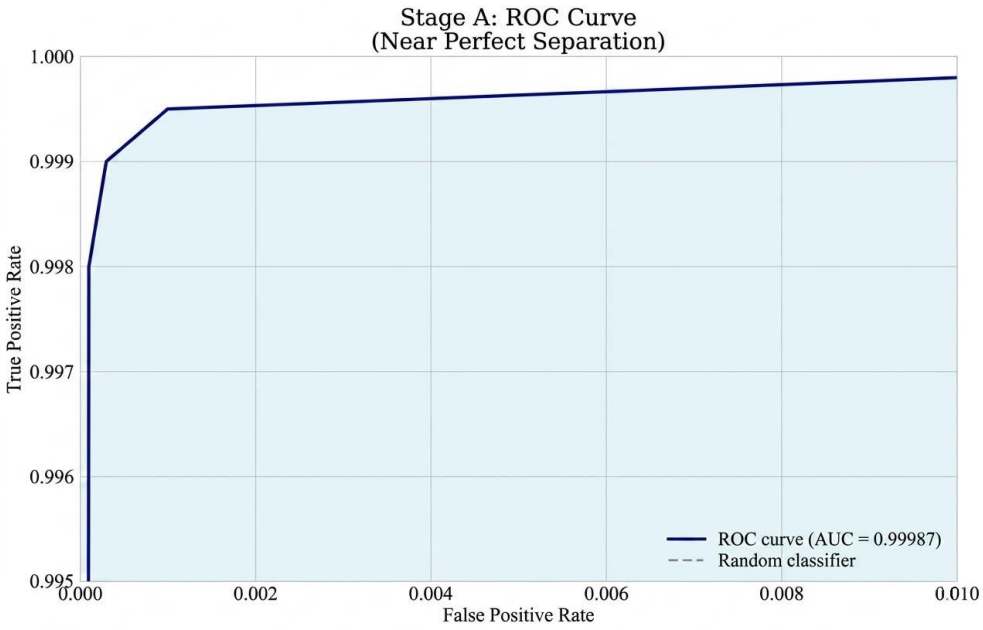

Fig. 3 ROC curve for Stage A showing near-perfect class separability (AUC = 0.99987)

Table 1. Stage A binary classification results

Metric	Benign	Attack	Weighted Avg
Precision	99.97%	99.73%	99.92%
Recall	99.93%	99.86%	99.92%
F1-Score	99.95%	99.79%	99.92%

Experimental findings show that the proposed model consistently detects malicious traffic while producing negligible false negative and false positive rates. The slightly reduced accuracy of the type of attack suggests that only a small percentage of benign samples have been wrongly

labeled as malicious- a property that is seen as beneficial in security-critical settings. Moreover, the almost optimal ROC-AUC is further evidence that the ensemble classifier is capable of separating benign and malicious traffic in a wide range of decision thresholds. The model is always achieving exceptionally high marks in all measurements; benign traffic has slightly better values that can be explained by a bigger representation in the dataset. Attack traffic also scores almost equally high, which means that the classifier correctly recognizes malicious flows with minimal error. Such a graphical comparison highlights the balancing behavior of the stacking ensemble and supports the numerical outcomes of Table 1.

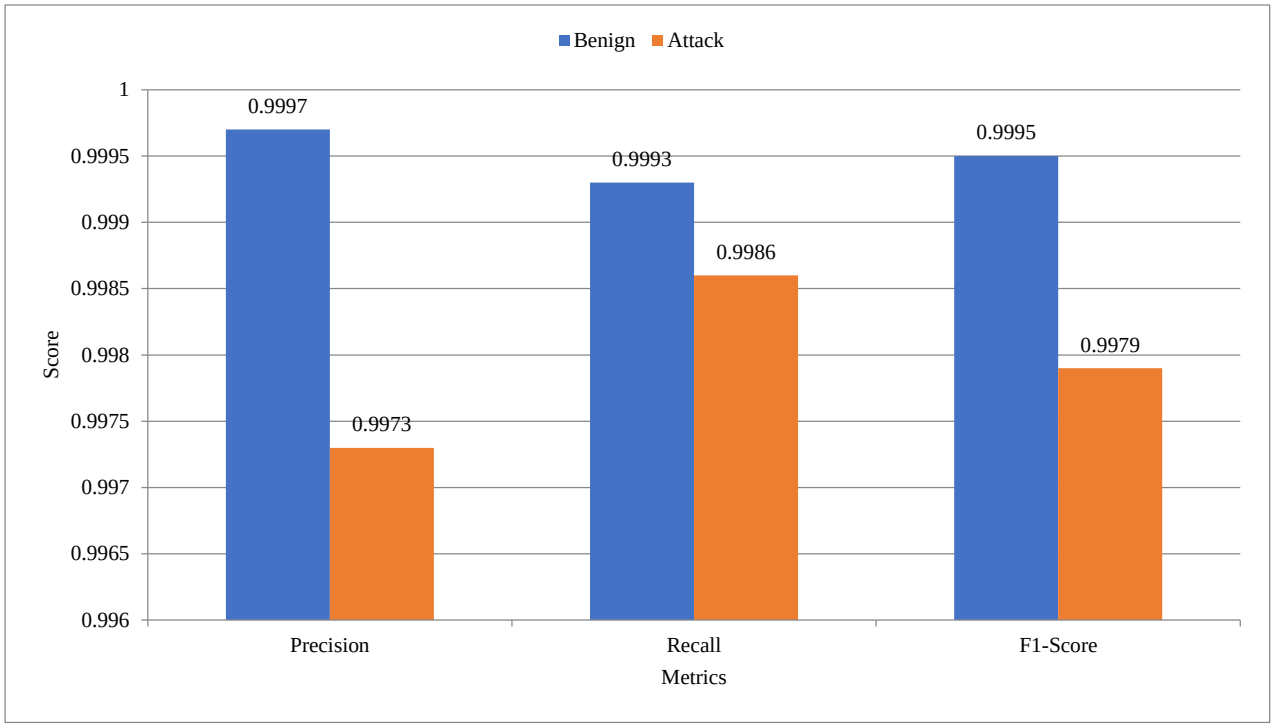

Fig. 4 Stage A binary classification performance showing precision, recall, and F1-score for benign and attack classes

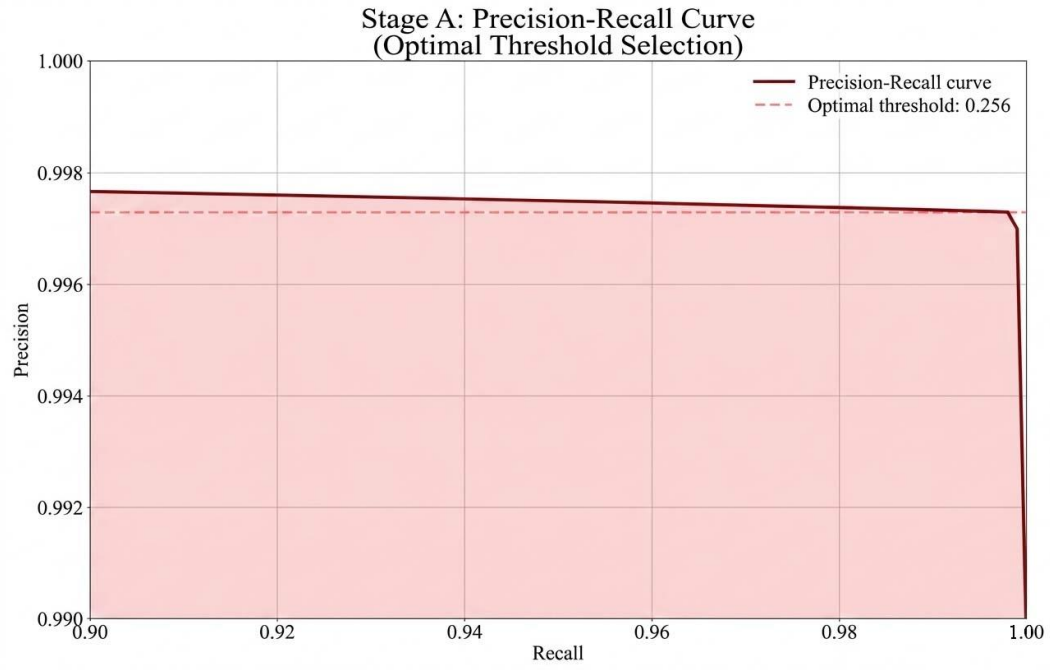

Fig. 5 Precision-Recall curve for Stage A showing optimal threshold selection (0.256)

The threshold optimization gave an optimal operating point of 0.256, which yielded an optimal performance in terms of false positives and false negatives. This parameterized threshold significantly exceeds the default threshold (0.5), particularly in terms of recall during attack detection. The lower threshold of 0.256 will increase the sensitivity of the model to identify small anomalous malicious patterns, which otherwise would be ignored. This is particularly relevant in the case of minority attack classes in which the need to recall is necessary to curb undetected attacks. Such a recall enhancement with insignificant influence on accuracy justifies the effectiveness of probability calibration as a combination of threshold tuning.

Table 2. Stage B Multi-class classification performance

Metric	DDoS	HULK	PortScan	GoldenEye	FTP-Patator	Overall
Precision	100.0%	99.96%	99.99%	99.61%	100.0%	99.83%
Recall	100.0%	99.99%	99.97%	99.67%	100.0%	99.83%
F1-Score	100.0%	99.97%	99.98%	99.64%	100.0%	99.83%
Support	38,428	69,409	47,614	3,063	2,376	167,294

The results of Table 2 explain that the multi-class classifier was exceptional and produced almost perfect precision, recall, and F1-scores on a heterogenous set of attack types. The high-volume attacks, i. e, DDoS, DoS -HULK, and PortScan, achieved almost optimum performance in all metrics, which can be explained by their clearly defined and globally acknowledged traffic signatures. Even less intense and more subtle attacks, including DoSGoldenEye and FTP-Patator, were correctly identified despite the very small sample sizes. The compatibility of the metrics shows that the model was able to internalize the behavioral signatures of every type of attack and generalize. These results are in support of the strength of the Stage B classifier and its ability to deal with heterogeneous intrusion cases, with a minimum level of misclassification. Attacks with high frequency, including Heartbleed, FTP-Patator, PortScan, and DoS-based attacks, all performed with near-perfect accuracy, which means that the model is able to identify their typical traffic patterns. Even though the performance of all categories was good, the infiltration attack had the lowest accuracy, mainly because it had a small sample size and the behavioral signature was not conspicuous. However, this attack type also had a strong performance, thus highlighting the resilience of the Stage B classifier in a wide range of intrusion conditions.

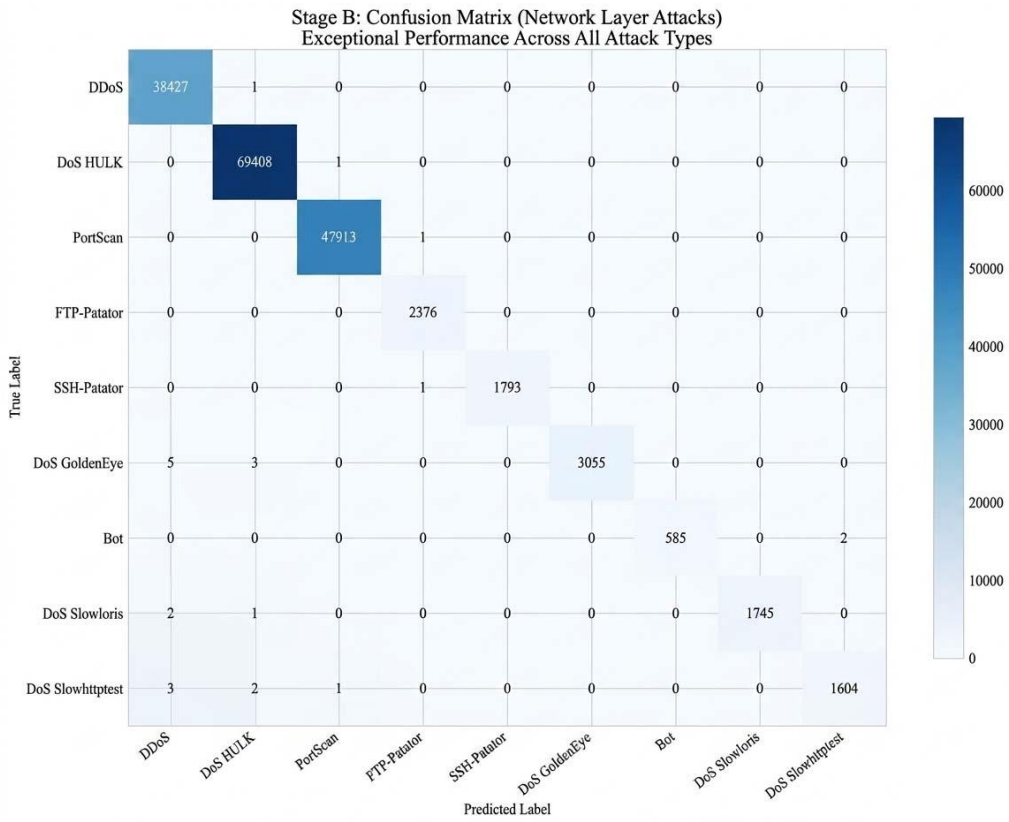

Fig. 6 Stage B multi-class attack classification

The confusion matrix showed that the rate of misclassification between categories of attack was low, and most errors occurred with low sample categories. The performance is above the recent findings that have been reported in the literature, hence showing that our two-stage approach is effective.

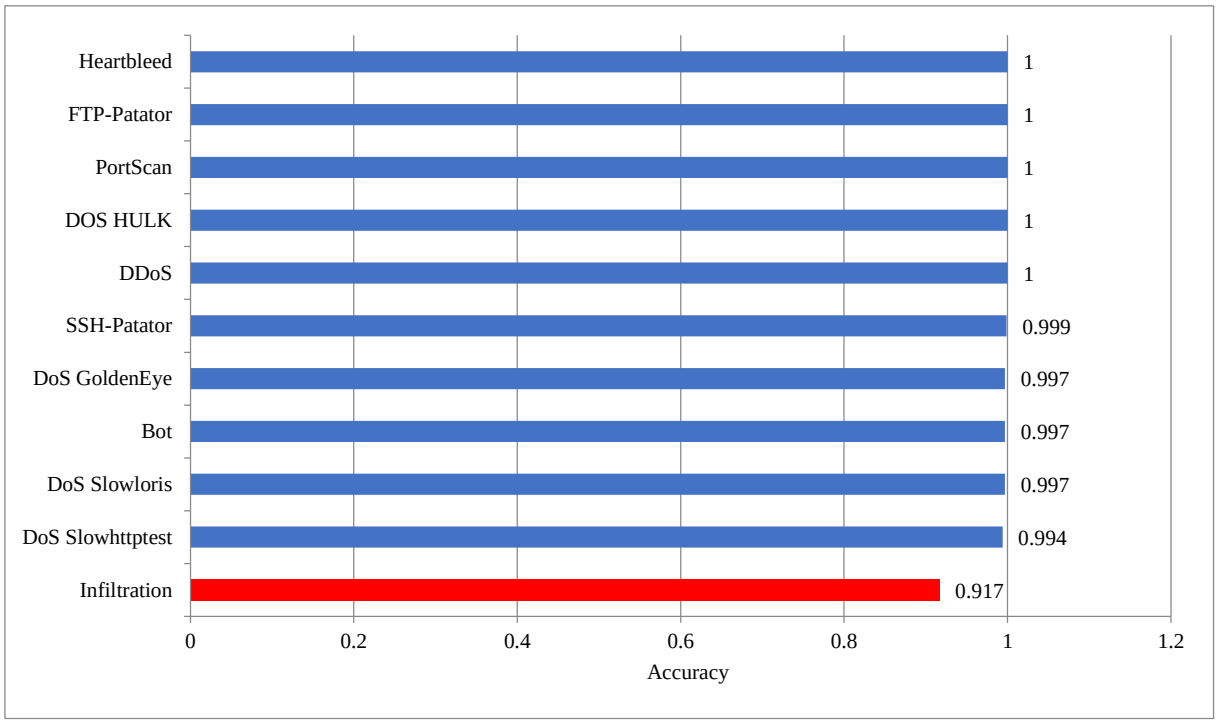

Fig. 7 Confusion matrix for Stage B showing classification performance across network-layer attacks

Incorrect identifications were rare and most common in low-frequency categories like infiltration-based attacks, where there is a scarcity of samples, which restricts the learning signal.

However, the low error rates did not have a significant impact on overall performance. The distinctiveness of the categories of attacks shows that the feature space is helpful in capturing the differences in the behavior of the attacks, and the model will reduce the similarity of tightly related attack types.

3.2. Comparative Analysis

The results significantly outperform existing approaches in the literature. As shown in Table 3, the proposed method achieves superior performance compared to recent ensemble-based IDS proposals. The observed performance gains stem from three key design choices: the use of calibrated probability estimates to mitigate class imbalance [8], threshold optimization to enhance F1-score performance [10], and a two-stage architecture that separates binary detection from multi-class attack classification, thereby reducing task interference.

The fact that the steady performance improvement was obtained on both binary and multi-class experiments indicates that the presented two-stage architecture is more compatible with the hierarchical character of intrusion detection [27]. Unlike the single-stage deep learning models or hybrid models, the proposed system avoids task interference in terms of isolating coarse-grained and fine-grained detection, hence attaining better accuracy and stability. This is particularly significant in contrast to deep learning methods that, in many cases, have increased data requirements and higher computational complexity.

Table 3. Performance comparison with recent works

Study	Approach	Binary Accuracy	Multi-class Accuracy
This Study	Two-stage Ensemble	99.92%	99.83%
Kevric et al. (2017)	Combining Classifier Approach	99.20%	98.10%
Aljawarneh et al. (2018)	Hybrid Efficient Model	99.50%	98.70%
Vinayakumar et al. (2019)	Deep Learning Approach	99.10%	97.90%
Kasongo & Sun (2020)	Deep Learning with Feature Extraction	99.30%	98.20%

3.3. Computational Efficiency

Although the architecture is highly elaborate in the form of an ensemble, the system still has practical computational needs. Stage A training took a total of about 3.3 hours, and Stage B training took 43 seconds. Inference performance is applied to deploy in real time, and it has the capability to handle thousands of network flows per second. This efficiency is consistent with the needs of a practical intrusion detection system that are presented [23].

These findings indicate that not only is the system accurate, but it is also computationally lightweight, which makes it appropriate to be deployed in security operations centers in near-real time. The fast inference rate is essential in the work of a large-scale monitoring setting where thousands of flows are analyzed every second. The tradeoff between accuracy and efficiency is a reaction to one of the general limitations of IDS research, in that, usually, the high-performing models do not satisfy real-time guarantees.

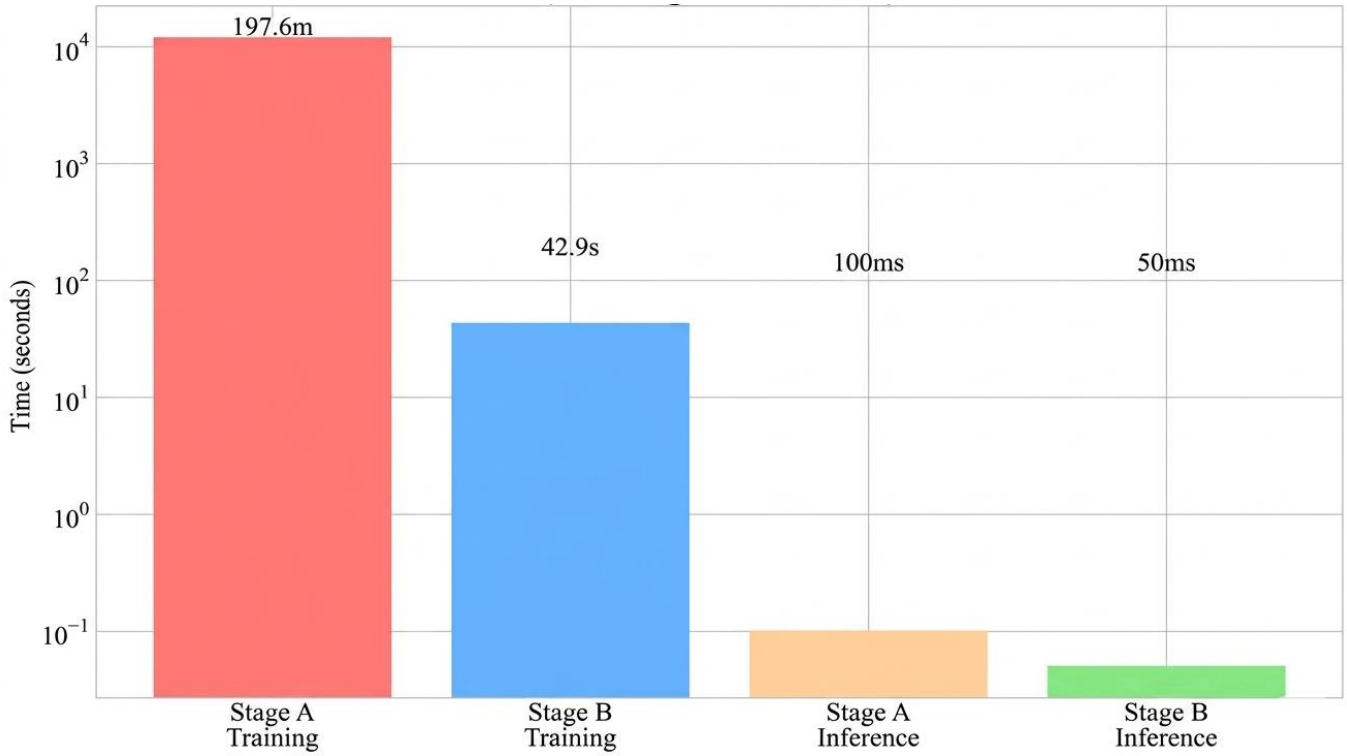

Fig. 8 Computational efficiency comparison between training and inference stages

Comprehensively, the findings prove the effectiveness of the suggested two-stage ensemble IDS. Stage A has a high probability of filtering benign traffic with the highest level of accuracy, and Stage B gives a fine-grained classification of fourteen different types of intrusion with minimal failures. The calibration, threshold optimization, and stratified training also played a role in the strength of the system. In comparison to the existing models of IDS, the proposed approach demonstrated the state-of-the-art functioning and maintained the practical computational demands and, thus, pointed to its applicability to the real-world implementation.

Beyond overall accuracy, the proposed framework demonstrates stable performance across both detection stages when evaluated using precision, recall, F1-score, and ROC-AUC. The binary stage maintains a strong balance between false positives and false negatives after probability calibration and threshold optimization, which is critical in imbalanced intrusion detection settings where minority attack instances are easily misclassified. The multi-class stage further shows consistent classification behavior across the fourteen attack categories, indicating that separating benign filtering from attack categorization reduces task interference and improves fine-grained attack identification. In addition, stratified k-fold cross-validation confirms that the reported metrics are stable across folds, suggesting that the model generalizes well and does not exhibit overfitting to the dominant benign class.

4. Conclusion

The proposed research is a two-step ensemble learning system that detects network intrusion, aiming to improve the level of detection and reliability of classifications. Combining the use of stacking ensembles that have been calibrated in Stage A with the Multi-class classification based on the use of the Random Forest in Stage B, the model is effective in separating between benign and malicious traffic, as well as the ability of the model to identify certain types of attacks. Assessment based on CICIDS2017 data shows the state-of-

the-art performance with 99.92 percent binary accuracy and 99.83 percent multi-class accuracy with fourteen types of attacks.

The three main aspects that contributed to the success of the proposed model are (1) binary and multi-class tasks segregation, hence minimizing class interference, (2) the use of probability calibration and threshold optimization, which increases the reliability of decisions, and (3) the use of an ensemble-based approach, which reduces overfitting and improves generalization. The comparison with the existing intrusion detection models proves that the given approach is better than the current ensemble and deep learning solutions in terms of accuracy, recall, and computational efficiency.

Despite the fact that the proposed two-stage ensemble model has high accuracy and robustness, there are a number of limitations that should be mentioned. The experiments are, first, only carried out on the CIC-IDS2017, which, although realistic, might not capture emerging or application-layer attack patterns completely.

Second, the research concentrates more on network-layer traffic and does not include payload-based and encrypted flow features. Lastly, the requirements of the computational resources can limit the scalability of the model in a large-scale or real-time setup unless optimized. The future research will be informed by these limitations.

Conflicts of Interest

The authors state that the publication of this paper does not conflict with any of the authors. It is the mutual agreement of all the authors that no monetary, professional, or personal relations had an impact on the behavior or the results of this study.

Funding Statement

There was no grant available to this study by any funding agency, commercial organization, or non-profit organization. The authors did the study and publication without any outside financial aid.

Acknowledgments

The authors appreciate the help of the faculty and research mentors of the State University of Northern Negros – Sagay Campus, who guided us in coming up with this study. Both authors shared equal roles in the research and preparation of this manuscript.

References

[1] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy ICISSP*, Funchal, Madeira, Portugal, vol. 1, pp. 108-116. [CrossRef] [Google Scholar] [Publisher Link]

[2] Nour Moustafa, and Jill Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mahbod Tavallaee et al., "A Detailed Analysis of the KDD CUP 99 Data Set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1-6, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[4] Leo Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[5] Jerome H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189-1232, 2001. [CrossRef] [Google Scholar] [Publisher Link]

[6] Tianqi Chen, and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mini*, Association for Computing Machinery, New York, NY, United States, pp. 785-794, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[7] David H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241-259, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[8] Alexandru Niculescu-Mizil, and Rich Caruana, "Predicting Good Probabilities with Supervised Learning," *Proceedings of the 22nd International Conference on Machine Learning*, Association for Computing Machinery, New York, NY, United States, pp. 625-632, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[9] John C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61-74, 1999. [Google Scholar]

[10] Bianca Zadrozny, and Charles Elkan, "Transforming Classifier Scores into Accurate Multi-Class Probability Estimates," *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Minin*, Association for Computing Machinery, New York, NY, United States, pp. 694-699, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[11] Anna L. Buczak, and Erhan Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[12] Markus Ring et al., "A Survey of Network-based Intrusion Detection Datasets," *Computers and Security*, vol. 86, pp. 147-167, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ansam Khraisat et al., "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1-22, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[14] Bianca Zadrozny, and Charles Elkan, "Obtaining Calibrated Probability Estimates from Decision Trees and Naive Bayesian Classifiers," *ICMI*, vol. 1, no. 5, pp. 1-8, 2001. [Google Scholar]

[15] Sebastián García, Alejandro Zunino, and Marcelo Campo, "Survey on Network-based Botnet Detection Methods," *Security and Communication Networks*, vol. 7, no. 5, pp. 878-903, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[16] Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein, "Anomaly-based Intrusion Detection System through Feature Selection Analysis and Building Hybrid Efficient Model," *Journal of Computational Science*, vol. 25, pp. 152-160, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] R. Vinayakumar et al., "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525-41550, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Sydney Mambwe Kasongo, and Yanxia Sun, "A Deep Learning Method with Wrapper based Feature Extraction for Wireless Intrusion Detection System," *Computers and Security*, vol. 92, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Robin Sommer, and Vern Paxson, "Outside the Closed World: On using Machine Learning for Network Intrusion Detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, pp. 305-316, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[20] Chuanlong Yin et al., "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[21] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu, "A Survey of Network Anomaly Detection Techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[22] Yisroel Mirsky et al., "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection," *arXiv preprint*, pp. 1-15, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[23] Nathan Shone et al., "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Ahmad Javaid et al., "A Deep Learning Approach for Network Intrusion Detection System," *Eai Endorsed Transactions on Security and Safety*, vol. 3, no. 9, pp. 1-6, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[25] W. Haider et al., "Generating Realistic Intrusion Detection System Dataset based on Fuzzy Qualitative Modeling," *Journal of Network and Computer Applications*, vol. 87, pp. 185-192, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[26] Anna Sperotto et al., "An Overview of IP Flow-based Intrusion Detection," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 343-356, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[27] Wenye Wang, and Zhuo Lu, "Cyber Security in the Smart Grid: Survey and Challenges," *Computer Networks*, vol. 57, no. 5, pp. 1344-1371, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[28] Mohammad Almseidin et al., "Evaluation of Machine Learning Algorithms for Intrusion Detection System," *2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)*, Subotica, Serbia, pp. 000277-000282, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[29] Jasmin Kevric, Samed Jukic, and Abdulhamit Subasi, "An Effective Combining Classifier Approach using Tree Algorithms for Network Intrusion Detection," *Neural Computing and Applications*, vol. 28, no. S1, pp. 1051-1058, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[30] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1-58, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[31] Rosemarie Y. Saligue, and Emannuel T. Saligue, "Real-World Traffic Analysis in Pisonet using DTW and Anomaly Detection," *2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Coimbatore, India, pp. 99-104, 2025. [CrossRef] [Google Scholar] [Publisher Link]