RESEARCH & INNOVATION DIVISION
IJIST
P ISSN : 2618-1630
E ISSN : 2909-6130

An Intelligent Machine Learning-Based Framework for IDS Using Clustering-Based Feature Engineering and Soft Voting Ensemble

Noor ul Ain Fatima[1*], Rashid Amin[1], Engr. Tahir Khan[2], Rahma Gantassi[3]
[1]Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan
[2]Department of Electronics Engineering, University of Engineering and Technology, Taxila, Pakistan
[3]Communication System Laboratory Sys'Com (ENIT), University of Tunis El Manar (UTM), Tunis, Tunisia.
***Correspondence:** 24-MS-CS-20@students.uettaxila.edu.pk, rashid.sdn1@gmail.com, tahir.khan@uettaxila.edu.pk, rahmagantassii@gmail.com

Citation | Fatima. N. U. A, Amin. R, Khan. T, Gantassi. R, "An Intelligent Machine Learning-Based Framework for IDS Using Clustering-Based Feature Engineering and Soft Voting Ensemble", IJIST, Vol. 8 Issue. 9 pp 3812-3830, October 2026
Received | Oct 03, 2026 **Revised** | Oct 18, 2026 **Accepted** | Oct 23, 2026 **Published** | Oct 28, 2026

With the growing complexity of network traffic and the ever-changing landscape of cyber threats, the demand for robust Intrusion Detection Systems (IDSs) powered by machine learning has grown significantly. The rise in network traffic complexity and the ever-changing nature of cyber threats have made Intrusion Detection Systems (IDSs) with machine learning capabilities more necessary than ever before. High-dimensional feature spaces, class imbalance issues, and the correct class identification of minority attack categories are still significant challenges, however. To solve these problems, the proposed solution in this study is the development of an intelligent intrusion detection framework that combines data preprocessing, clustering-based feature engineering, Principal Component Analysis (PCA) and soft-voting ensemble learning. To enhance the uniformity of features and overcome class imbalance, the framework uses feature standardization and random oversampling. Mini-Batch-K-Means clustering is used to assign clusters and add centroid-distance features to extract more traffic characteristics while PCA is used to reduce the dimensionality of features by transforming the expanded feature space into a compact representation. Next, Multiclass intrusion detection is performed by soft voting with probabilities on top of Random Forest (RF), Extra Trees (ET), and XG Boost (XGB) classifiers. The proposed structure is tested on CICIDS2017 dataset via classification performance metrics, class-wise analysis and 6-stage ablation study. The reported results show that the accuracy was 99.92%, macro precision was 96.04%, macro recall was 96.25% and macro F1 score was 96.13%. The results of the ablation study show continuously improved performance in the framework components, with accuracy improving from 99.41% to 99.92% (+0.51 percentage points), and macro F1-score rising from 90.96% to 96.13% (+5.17 percentage points). In addition, underrepresented attack categories (such as Bot and Web Attack) are evaluated to look at detection performance in addition to aggregate accuracy. The results show that clustering-based features, dimensionality reduction, and heterogeneous ensemble learning can be used to enhance the multiclass intrusion detection system for CICIDS2017 benchmark. In any case, evaluation and testing of the proposed framework in operational network environments is required to prove the generalizability and practical reliability of the framework.

Keywords: Intrusion Detection System, Machine Learning, Clustering-Based Feature Engineering, Principal Component Analysis, Soft Voting Ensemble, Network Security, CICIDS2017.

IPIndexing Indexing Portal
CiteFactor Academic Scientific Journals
RESEARCHBIB ACADEMIC RESOURCE INDEX
IDEAS
Scientific Journal Impact Factor TOGETHER WE REACH THE GOAL
JOURNALS MASTER LIST
ROOTINDEXING JOURNAL ABSTRACTING AND INDEXING SERVICE
Scilit
INFOBASE INDEX
DRJI
Crossref
ResearchGate
WIKIDATA

Introduction:

The volume, diversity, and complexity of network traffic have dramatically increased due to the quick development of cloud computing, the Internet of Things (IoT), and sophisticated network infrastructures. This change has facilitated the development of advanced digital services but has also brought about severe security issues, especially the emergence of advanced cyberattacks on network systems. Intrusion Detection Systems (IDSs) are crucial in protecting network environments by monitoring traffic and detecting malicious activities and unauthorized access. Conventional IDS systems, particularly signature-based systems, are useful in identifying known attacks, but they are not effective in detecting zero-day and emerging threats because they are based on pre-defined patterns and rules [1]To address these shortcomings, studies have been more inclined towards ML-based IDS that allow models to learn patterns based on data and identify known and unknown attacks with greater adaptability and accuracy [2]These systems use data-driven methods to derive meaningful features and automatically detect complex attack patterns. Consequently, ML-based IDS systems have emerged as a promising solution to contemporary cybersecurity issues, especially in dynamic and large-scale network settings [3]Moreover, hybrid and ensemble learning methods have been proposed to improve the accuracy of detection and minimize false positives through the integration of several classifiers [4]Other recent developments are scalable ML-based IDS systems that can deliver high performance on the benchmark dataset, [5]Besides classification-based techniques, anomaly detection techniques have also received considerable interest in detecting unknown and low-frequency attacks. Local Outlier Factor, Isolation Forest, and One-Class SVM are some of the most popular models that are used to identify abnormal patterns without the need to have fully labelled data, [6]Nevertheless, they tend to have high false-positive rates and sensitivity to data distribution, particularly when imbalanced or noisy data are involved [7]To overcome these issues, recent studies have included dimensionality reduction methods and feature selection to enhance efficiency and performance[8], and the significance of model interpretability and robustness in practical applications [9]Modern IDS also need to work in environments that are always changing, where the patterns of network traffic are always changing. To address these issues, this paper develops a holistic machine learning-based intrusion detection system that incorporates preprocessing, dimensionality reduction, feature engineering, and ensemble learning to improve detection accuracy and strength.

Research Objectives:

The primary objective of this study is to develop and systematically evaluate a machine learning-based intrusion detection framework that integrates clustering-based feature engineering, principal component analysis (PCA), and ensemble learning for multiclass network intrusion detection.

The specific research objectives are as follows:

To design an intrusion detection system pre-processing and feature optimization technique that combines standardization, class balancing, clustering-based feature engineering and Principal Component Analysis (PCA) techniques to enhance intrusion detection performance.

To evaluate the individual contributions of the proposed framework components through systematic ablation using accuracy, macro-precision, macro-recall, and macro-F1 score.

To evaluate and compare the classification performance of Random Forest, Extra Trees, XGBoost, and the soft-voting ensemble for multiclass intrusion detection.

To assess the performance of individual attack categories in terms of class-wise precision, recall, F1 score and false positive rates, especially minority classes.

Novelty and Research Contributions:

The novelty of the proposed framework lies in the systematic integration and experimental evaluation of complementary data preprocessing, feature engineering, dimensionality reduction, and ensemble classification techniques for multiclass intrusion detection. Unlike approaches that rely exclusively on original traffic features or individual classification algorithms, the proposed framework incorporates clustering-derived information into the feature representation before applying PCA and a heterogeneous soft-voting ensemble. Specifically, Mini-Batch-K Means is employed to generate cluster-assignment and cluster-distance features, allowing the classifiers to incorporate information about the structural relationships among network traffic observations. PCA is subsequently used to reduce the expanded feature representation, while Random Forest, Extra Trees, and XG Boost are combined through probability-based soft voting. The individual techniques employed in this framework are established methods. Therefore, the contribution is not the invention of a new clustering algorithm, dimensionality reduction method, or ensemble classifier. Instead, the study investigates their complementary roles within a unified intrusion detection pipeline and evaluates the contribution of each component through comparative and ablation experiments.

Related Work:

This paper has presented a powerful machine learning model by [10] of network intrusion detection which aids in feature selection. Their contribution shows that a 2024 trend of reducing the number of features and maintaining high attack classification performance is significant, which is relevant to the study of ML-based IDS, since it shows that careful feature engineering can provide competitive results without overly complex designs. [11] proposed a machine-learning-based system to identify intrusions in IoT security attacks. The study is important because it focuses on the IoT setting, where various traffic and scarce resources make the traditional IDS methods less effective. The article contributes to the literature by showing that the design of ML-based IDS is shifting to smart-device ecosystems rather than traditional enterprise traffic. [12] suggested a multi-label network attack detection methodology that employed two-stage model fusion.

The work is noteworthy because it goes beyond single-label attack prediction to address the more realistic scenario where network traffic can exhibit overlapping attack characteristics. These two-stage fusion strategies can be used in modern IDS pipelines to enhance the granularity of detection. [13] developed a lightweight IDS that was built on a stochastic gradient descent classifier and utilized efficient feature subsets. Their study contributes to the literature by addressing the issue of lightweight detection that is particularly useful in cases when speed is as significant as accuracy. [14] proposed a zero-day IoT intrusion detection system that combines binary snake optimization with an attention-based bidirectional gated recurrent classifier. This study is particularly important because zero-day detection is still one of the hardest problems in intrusion detection.

The paper shows how optimization and attention-enhanced recurrent learning can be used together to better recognize threats that have never been seen before. [15] developed a new framework for an intrusion detection system utilizing a least-squaresSVM. Their contribution consists of re-examining kernel-based supervised learning for IDS and demonstrating that optimized SVM variants continue to be pertinent in contemporary intrusion detection research, particularly in efforts to enhance classification stability and minimize false predictions. [16] proposed a resilient machine-learning-based intrusion detection system utilizing straightforward statistical methods for feature selection in Scientific Reports. This work is beneficial for the literature as it illustrates that superior IDS performance does not necessarily necessitate advanced feature-selection pipelines; meticulously crafted

statistical screening can also yield resilient intrusion detection models with reduced complexity. [17]created a multi-stage ML-based IDS and feature-selection ensemble framework for cloud-assisted vehicular ad hoc networks. This study expands IDS research into VANET environments and underscores the efficacy of multi-stage learning in enhancing detection within highly dynamic networks, where topology changes and mobility complicate attack identification. [18]put forward MFEI-IDS, an intrusion detection system that uses multi-level feature extraction and inductive learning.

Their framework combines inductive learning with FCN Transformer-based feature extraction. This is important to current IDS literature because it combines deep representation learning with a classification mechanism that is meant to improve generalization on complex attack data. [19]described a lightweight intrusion detection system that uses image encoding and ECANet for networked autonomous vehicles. This article contributes to the literature by targeting lightweight deployment for self-driving cars, fast inference and reliable recognition are significant in applications that are critical to safety. [5] discussed FC-Trans, a deep-learning algorithm to locate network data breaches in big data environments. This research is important because it addresses the issue of scalability, which is a major issue in modern IDS studies and shows how deep architectures can be flexible. large-scale traffic analysis and not just small benchmark data. [20]suggested a graph-based federated learning approach to intrusion.

IoT network detection in Scientific Reports. The paper is particularly significant as it fuses federated learning and graph-based modelling. This allows intrusion detection to be more advantageous. with decentralized training and retains more structural information on how traffic flows between IoT nodes. [9] created a framework for IoT intrusion detection that uses rule induction to explain models in Sensors. This study introduces a significant aspect of explainability to IDS research by demonstrating that ML-based detection can be integrated with rule induction to enhance interpretability, which is essential for analysts who need to trust and comprehend model decisions. [21] suggested a hybrid deep-learning method for protecting IoT networks from DDoS attacks in Sensors. Their research enhances the understanding of DDoS behavior in IoT traffic and supports the prevailing literature assertion that hybrid deep learning models more effectively capture intricate temporal and spatial attack patterns compared to single-model methodologies. [22] presented a thorough, real-time network intrusion detection system that makes use of similarity embedding learning and incremental feature encoding. This paper is significant as it directly tackles real-time operation and generalization, two enduring deficiencies in IDS research, and proposes that embedding-based methodologies can enhance model adaptability in response to fluctuating traffic conditions. [23] reviewed of supervised ML intrusion detection and a multi-criteria evaluation. The importance of this paper is that it not only summarizes the past supervised models of IDS but also evaluates them based on a set of criteria that enable a more balanced selection of a model that goes beyond accuracy [24] proposed a lightweight cascade-based system to detect zero-day attacks in real time on computers. Their input is in the form of creating a two-stage cascade architecture of CPU-only environments, making the work applicable to real-world deployment environments where high-end hardware might not be available. A new hybrid graph neural network and transformer model to detect intrusions is proposed by [25]They combine GNNs, Transformer learning, greyscale image transformation, GLCM texture extraction, and Borderline-SMOTE, which is a complex innovation that demonstrates the growing overlap of graph learning and attention processes in IDS studies. [26]developed GConvTrans, a graph-based convolutional network and transformer architecture to identify network intrusions. The paper shows that converting tabular traffic records to graph representations improves the model to capture local structural information and global contextual dependencies to better intrusion detection in distributed settings.

Table 1. Comparative Analysis of Existing Intrusion Detection Systems and the Proposed Framework

Study	Proposed Method	Dataset	Acc (%)	Limitations / Research Gap
[13]	Lightweight IDS using stochastic gradient descent (SGD)	KDD-CUP-1999, BotIoT-2018, N-BaIoT-2021	92.69 (average)	Requires further evaluation of minority-class detection and efficiency trade-offs.
[18]	Multilevel feature extraction using FCN-Transformer and inductive learning	ISCX 2012	98.50	Deep learning architecture introduces additional computational requirements.
[25]	Hybrid GNN–Transformer with image-based feature extraction	UNSW-NB15, IEC61850, DAPT2020	99.24 (best reported)	Requires graph construction and additional feature transformations.
[26]	GConvTrans integrating graph convolution and Transformer learning	CIC-IDS2018	96.94	Graph-based representation increases implementation complexity.
Proposed Framework	Clustering-based feature engineering, PCA, and RF–ET–XG Boost soft-voting ensemble	CICIDS2017	99.92	Independent-dataset validation is required to establish generalizability.

The comparative analysis in Table 1 highlights the effectiveness of different machine learning and deep learning approaches for intrusion detection.[13], [18][25] and [26]reported accuracies of 92.69%, 98.50%, 99.24%, and 96.94%, respectively. However, these approaches involve challenges related to computational complexity, feature processing, and minority-class detection. In comparison, the proposed framework combines clustering-based feature engineering, PCA, and soft-voting ensemble learning, reporting an accuracy of 99.92% on CICIDS2017. Since the studies use different datasets, direct performance comparisons are limited, and independent validation is required to establish generalizability.

Proposed Methodology:

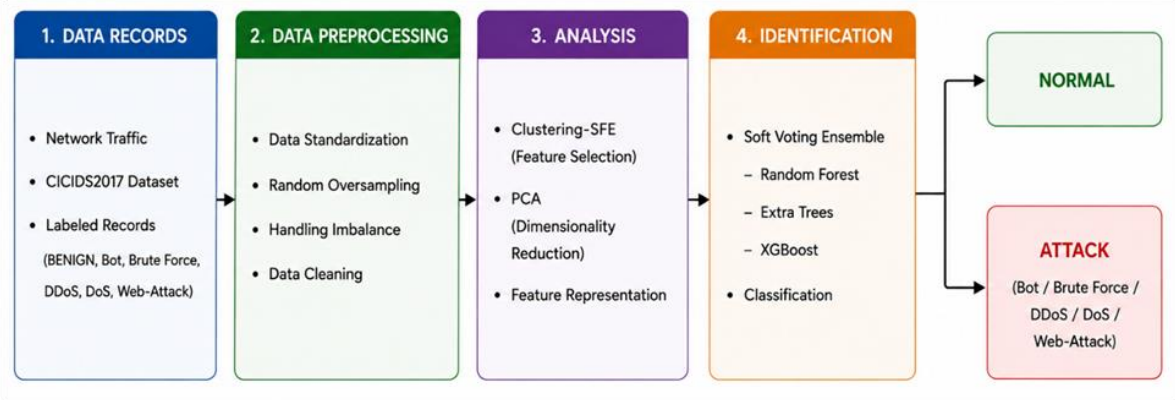

Figure 1. Data processing and attack classification flowchart

The proposed intrusion detection system in Figure 1 employs state-of-the-art preprocessing, feature engineering, dimensionality reduction, and ensemble learning methods to simplify the detection of known and unknown cyberattacks. The methodology begins with the CICIDS2017 dataset and proceeds through data cleaning, standardization, and random oversampling to address noise and class imbalance. After that, clustering-based feature

engineering (SFE) is applied to enhance the processed data and PCA is applied to reduce the number of dimensions to make it more efficient and more representative of the data. Lastly, a soft voting ensemble is a combination of multiple machine learning models (Random Forest, Extra Trees, and XGBoost) to properly classify network traffic as normal or attack. This ensures that intrusion detection is powerful and sound.

Dataset Description:

The CICIDS2017 dataset [27]for intrusion detection studies is one of the most popular benchmark datasets that represents modern network traffic and cyberattacks realistically. It not only has labelled network flow records collected with CIC Flow Meter but also has a set of normal traffic and several categories of attacks, including Bot, Brute Force, DDoS, DoS and Web Attacks. Each record contains statistical and behavioral characteristics of the network flows, such as flow information, packet statistics, timing properties, and communication characteristics, but not the actual payloads of the packets. These features also allow machine learning models to successfully differentiate between non-threatening and threatening traffic. The dataset was pre-processed in this study to ensure the high quality of the input data for the model training and evaluation process, which is done by cleaning, normalizing, and optimizing the features.

Table 2. Dataset Statistics (CICIDS2017)

Category	Number of Records	Description
Benign	2,273,097	Legitimate network traffic representing normal user activities without malicious behavior.
DDoS	128,027	Distributed Denial-of-Service attacks that overwhelm network services using traffic from multiple sources.
DoS	252,673	Denial-of-Service attacks intended to disrupt network services or resources.
Brute Force	13,850	Repeated authentication attempts intended to gain unauthorized access.
Other Attacks	36,000+	Additional malicious activities, including web-based attacks, infiltration, and botnet traffic.

The table presents the distribution of network traffic categories in the CICIDS2017 dataset, highlighting the imbalance between benign and malicious traffic. Benign traffic represents the largest category, containing 2,273,097 records, followed by DoS attacks with 252,673 records and DDoS attacks with 128,027 records. Brute Force attacks account for 13,850 records, while other attack categories, including web-based attacks, infiltration, and botnet activities, collectively contain more than 36,000 records. This uneven distribution highlights the challenge of class imbalance, which may affect the detection of less frequent attacks. Therefore, appropriate preprocessing and class-balancing techniques are important for improving multiclass intrusion detection performance.

Data Preprocessing and Leakage Prevention:

To ensure reliable model evaluation and minimize the risk of data leakage, all data-dependent preprocessing operations were performed after partitioning the CICIDS2017 dataset into training, validation, and testing subsets. Stratified sampling was employed to preserve the relative distribution of attack categories across the three subsets. The preprocessing pipeline consisted of missing-value imputation, feature standardization, random oversampling, clustering-based feature engineering, and Principal Component Analysis (PCA). Initially, missing-value imputation and feature standardization were fitted exclusively on the training subset.

The resulting imputation statistics and scaling parameters were subsequently applied to the validation and testing subsets without refitting. To address class imbalance, random oversampling was performed only on the transformed training data, while the validation and testing subsets retained their original class distributions. Next, clustering-based feature engineering was performed using Mini Batch K Means with eight clusters, a batch size of 10,000, ten initializations, and a fixed random state of 42.

The clustering model was fitted exclusively on the balanced training data. Nine clustering-derived features were generated for each observation, comprising one cluster-assignment feature and eight centroid-distance features. These features were combined with the 69 original numerical features, producing a 78-dimensional feature representation. Subsequently, PCA was fitted exclusively on the training-derived feature representation to reduce dimensionality from 78 to 20 principal components. The fitted clustering model and PCA transformation were then applied to the validation and testing subsets without updating their learned parameters.

Consequently, no validation or testing observations were used to estimate preprocessing statistics, generate oversampled training records, determine cluster centroids, or calculate principal components. This training-only fitting strategy-maintained separation between the data subsets, reduced the risk of information leakage, and supported a consistent evaluation of the proposed intrusion detection framework. CICIDS2017 dataset was partitioned into training, validation, and testing subsets before fitting data-dependent preprocessing transformations. Stratified sampling was employed to preserve the relative distribution of attack classes across the subsets. Missing-value imputation and feature standardization were fitted exclusively on the training data.

The fitted transformations were subsequently applied to the validation and testing subsets without recalculating their parameters. To address class imbalance, random oversampling was performed only on the training subset. Validation and testing samples were not oversampled, allowing model performance to be assessed on their original class distributions. Clustering-based feature engineering was implemented using Mini Batch K Means with eight clusters, a batch size of 10,000, ten initializations, and a fixed random state of 42. The clustering model was fitted using the balanced training data. For each observation, nine clustering-derived features were generated: one cluster-assignment feature and eight features representing distances to the cluster centroids. These features were combined with the original processed features, producing a 78-dimensional representation. PCA was then fitted exclusively on the training representation to reduce the feature space to 20 principal components. The same fitted clustering and PCA transformations were applied to validation and testing data. This procedure prevents validation and testing observations from influencing the fitted preprocessing, clustering, and dimensionality reduction parameters.

Experimental Data Partitioning:

The dataset CICIDS2017 was split into three mutually exclusive subsets by stratified random sampling, with 70% for training, 15% for validation and 15% for testing purposes, to ensure a consistent, unbiased evaluation. A fixed random state of 42 was used to assure repeatability. The relative distribution of network traffic categories within the subsets remained the same as stratification. The classification models and the training data were used to fit the missing-value imputation, feature standardization, feature engineering with clustering, and Principal Component Analysis (PCA). Random oversampling was only applied on training set to overcome class imbalance; no duplicated observations were found in validation and test sets. The validation and testing data were not re-fit with the fitted preprocessing transformations. The validation subset was used only for model comparison and tuning the hyperparameters, and the test subset was used only for final performance

evaluation. To guarantee inter-classifier comparison and avoid data leakage, identical data partitions and evaluation metrics were kept for the individual classifiers and the proposed ensemble.

Classifier Configuration and Soft-Voting Ensemble:

Random Forest (RF) and Extra Trees (ET) classifiers were implemented using 220 estimators each, with unrestricted maximum tree depth, a minimum of two samples required for node splitting, and balanced subsample class weighting. A fixed random state of 42 was used to support reproducibility. The experimental code also specifies an XGBoost configuration with 220 estimators, a maximum tree depth of 8, a learning rate of 0.08, subsampling and column-sampling ratios of 0.90, and the multiclass probability objective (multi: softprob). However, XGBoost was disabled in the executed experiment and was therefore excluded from the reported model comparison. The soft-voting ensemble combined the class probabilities predicted by the available constituent classifiers using equal weights. For each observation, the probability estimates for a given class were averaged, and the class with the highest resulting probability was selected as the final prediction. Model performance was evaluated on a common validation set, and the best-performing configuration was selected primarily according to macro F1-score, followed by balanced accuracy and overall accuracy. The selected model was subsequently evaluated on the independent test set.

Table 3. Hyper parameter Configuration of the Classification

Classifier	Hyperparameter	Value
Random Forest	n_estimators	220
	max_depth	None (unrestricted)
	min_samples_split	2 (default)
	class_weight	balanced_subsample
	random state	42
	n_jobs	-1
Extra Trees	n_estimators	220
	max_depth	None (unrestricted)
	min_samples_split	2 (default)
	class_weight	balanced_subsample
	random_state	42
	n_jobs	-1
XGBoost	n_estimators	220
	max_depth	8
	learning_rate	0.08
	subsample	0.90
	colsample_bytree	0.90
	objective	multi:softprob
	eval_metric	mlogloss
	tree_method	hist
	random_state	42

Classifier	Hyperparameter	Value
Soft-Voting Ensemble	voting	soft
	weights	None (equal weights)
	n_jobs	-1

Training and Testing:

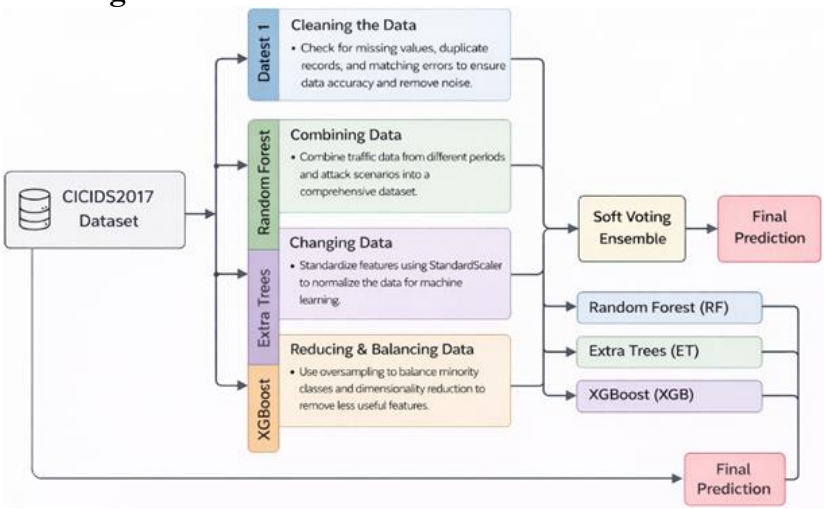

Figure 2. Training and Testing of the proposed IDS

Figure 2 shows the training and testing process for the proposed intrusion detection system. To enhance data quality and learning efficiency, data in the CICIDS2017 dataset will be pre-processed with data cleaning, data standardization and data class balancing. The processed data is then fed to three machine learning classifiers RF, ET, and XGB. Random Forest introduces the concept of bootstrap aggregation to boost the stability of classification, Extra Trees introduces the concept of extra randomness in the construction of trees to enhance the generalization, and XGB introduces the concept of gradient boosting to learn complex attack patterns. The trained models predict classes for network traffic records, and the ensemble strategy of soft voting is applied to combine the multiple predictions to gain the best detection performance and robustness. For each classifier, a mapping function is learnt in Equation 2:

$$[\boldsymbol{F(x) = y}] \qquad 2$$

The input feature vector is represented by where (x)and the predicted traffic class by (y). The goal of training is to learn patterns that are discriminative and that correctly classify benign and malicious network traffic.

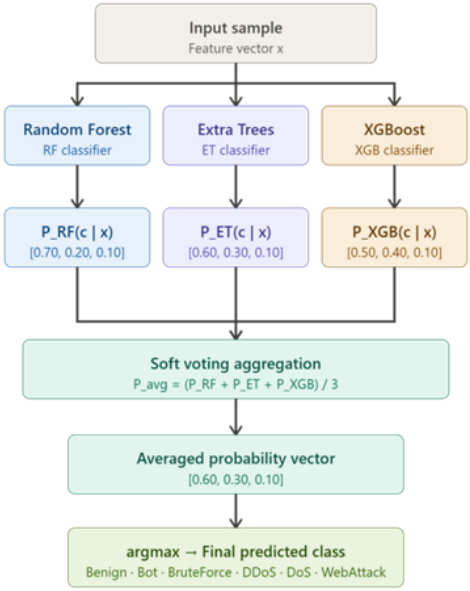

Figure 3. Soft Voting Ensemble Learning Architecture for Multi-Class Intrusion Detection

The proposed framework adopts the soft voting ensemble mechanism as shown in Figure 3. The classifiers (Random Forest, Extra Trees, and XGBoost) provide independent probabilities for each attack class, given an input network traffic sample. These probabilities are combined by taking their average and resulting in a single probability distribution. The final prediction is chosen based on the class that has the maximum averaged probability. Soft voting is a voting technique that combines the classifiers' confidence scores in addition to majority voting, which provides more stable and reliable predictions as compared to hard voting. The ensemble works based on the fact that the combination of the complementary abilities of different classifiers minimizes the errors made by the individual classifiers and enhances the overall intrusion detection performance.

Phase of Detection:

In the detection stage, we use the selected model (or ensemble model) to test network data that hasn't been seen before. The model is given each new network flow after it has been passed through the feature extractor. A soft voting ensemble method is used to make detection even better. This approach combines the probabilities of the different models (RF, ET, and XGB) are summed, then the prediction is made using the class with the highest probability. The ensemble prediction can be formally articulated as:

$$\boldsymbol{P(y \mid x) = \frac{1}{n}\sum_{i=1}^{n} P_i\,(y \mid x)} \qquad \boldsymbol{3}$$

The soft voting ensemble method uses Equation 3. The aim is to combine the predictions of several machine learning algorithms and combine them into a single probability. In this equation, x is the sample data, in your paper, that is one network traffic record or one flow instance. y stands for a class label, such as benign, DDoS, DoS, Brute Force, or another type of attack. $P\ i\ (y \mid x)$. The ith the model predicts the probability of class y when the input is x. If you are using three models, like Random Forest, Extra Trees, and XGBoost, each one will give you a different probability for each class. n is the number of models in the ensemble. If you use three classifiers, then n = 3.

The sign ∑i=1n∑i=1n means "add the probabilities from all models from model 1 to model n." We divide the sum by n, which gives us the average probability across all models. This equation shows us that the final probability for a class is just the average of the probabilities that all the individual models predicted. For instance, if three models are given one traffic record, they might predict the chance of a DDoS attack like this:

The scores are 0.80 for Random Forest, 0.70 for Extra Trees, and 0.90 for XGBoost. The last soft voting chance is:

$$P(y \mid x) = \frac{0.80 + 0.70 + 0.90}{3}$$

$$P(y \mid x) = \frac{2.40}{3} = 0.80$$

This means that the ensemble system says there is an 80% chance that the DDoS class will happen. The same math is done for all the other classes, and the one with the highest final probability is chosen as the final prediction. The best thing about this equation is that it doesn't rely on just one model. It brings together the best parts of all the models. If one model makes a prediction that is a little off or wrong, the other models can make up for it. Because of this, the final system is more stable, reliable, and accurate than just one classifier. The last prediction is then:

$$\hat{y} = argmax P(y \mid x) \qquad 4$$

After using equation 2 to figure out the odds, equation 4 is used to make the final prediction(decision). Here, y^ shows the last predicted class. This means what your model gives you, like whether the network traffic is benign, DDoS, DoS, brute force, etc.

The notation $P(y \mid x)$denotes the likelihood of each class contingent upon the input data x. The soft voting ensemble (the equation above) has already calculated these probabilities. The term "argmax" refers to "argmax" - choose the class with the highest chance. It looks at all probabilities and chooses the class with the highest probability. This method uses the best bits of multiple models to ensure that mistakes made in one model don't affect the stability of the classification.

Experimental Environment:

The CICIDS2017 dataset was used for experiments which includes both benign and malicious network traffic data with realistic cyberattacks. The dataset was split into three parts: training set, validation set, and testing set for model building, hyperparameter tuning, and final performance evaluation, respectively. The implementation was done in Python and Kaggle Notebooks with the support of CPU and GPU for the training of the model. Data preprocessing, model development and model analysis were done using common machine learning libraries such as NumPy, Pandas, Scikit-learn, XGBoost, Matplotlib and Seaborn. The proposed approach is to use Standard-Scaler for normalizing the features, Random Oversampling for balancing the classes, clustering to engineer features, PCA for dimensionality reduction and a soft voting ensemble of Random Forest, Extra Trees and XGBoost classifiers.

Evaluation Metrics:

For the evaluation of the proposed intrusion detection framework, the various widely used metrics for multiclass classification problem were used, namely Accuracy, Precision, Recall and F1-score. Precision scores represent the percentage of attack instances correctly classified and accuracy scores represent the overall classification performance. You can calculate the Recall measure, which indicates how well the model can identify (or detect) actual attacks, and the F1 score is a measure that gives a balanced view of the Precision and Recall. Furthermore, the robustness and effectiveness of the proposed framework have been evaluated by using the ROC curves, Precision-Recall curves, confusion matrix analysis, and ablation studies.

$$\left[Accuracy = \frac{TP+TN}{TP+TN+FP+FN}\right] \left[Precision = \frac{TP}{TP+FP}\right] \left[Recall = \frac{TP}{TP+FN}\right] \left[F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision+Recall}\right] \qquad 5$$

(Where in equation 5 (TP), (TN), (FP), and (FN) denote true positives, true negatives, false positives, and false negatives, respectively.)

Performance Analysis:

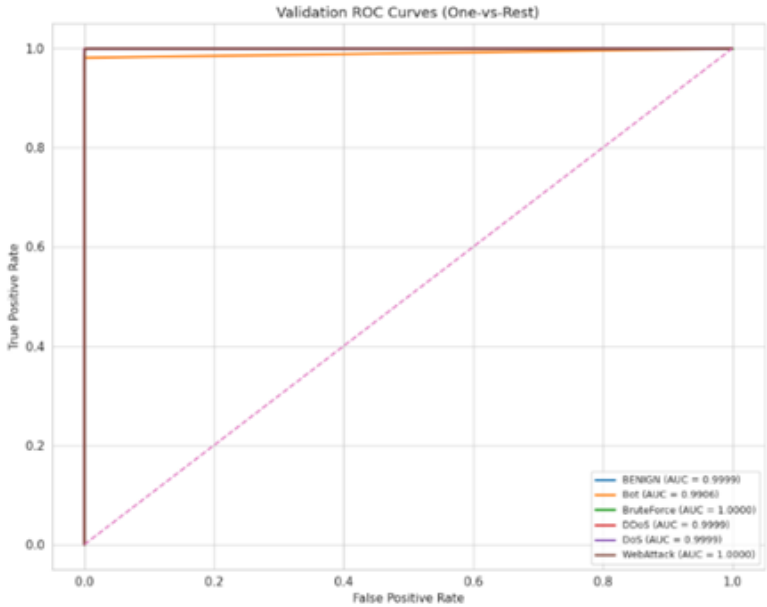

Figure 4. Validation ROC Curves (One-vs-Rest)

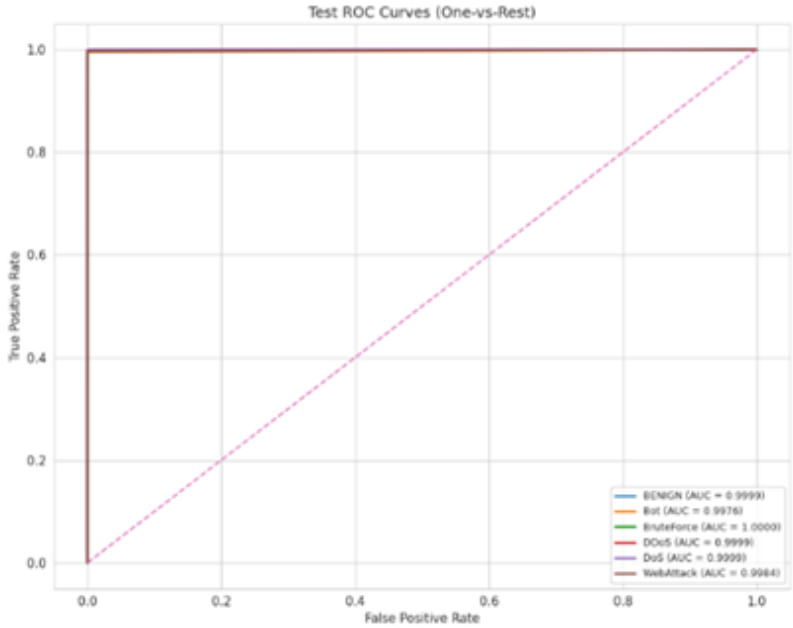

Figure 5. Test ROC Curves (One-vs-Rest)

The CICIDS2017 dataset was used for experiments which, includes both benign and malicious network traffic data with realistic cyberattacks. The dataset was split into three parts: training set, validation set, and testing set for model building, hyper parameter tuning, and final performance evaluation, respectively. The implementation was done in Python and Kaggle Notebooks with the support of CPU and GPU for the training of the model. Data preprocessing, model development and model analysis were done using common machine learning libraries such as NumPy, Pandas, Scikit-learn, XGBoost, Matplotlib and Seaborn. The proposed approach is to use Standard-Scaler for normalizing the features, Random Oversampling for balancing the classes, clustering to engineer features, PCA for dimensionality reduction and a soft voting ensemble of Random Forest, Extra Trees and XGBoost classifiers.

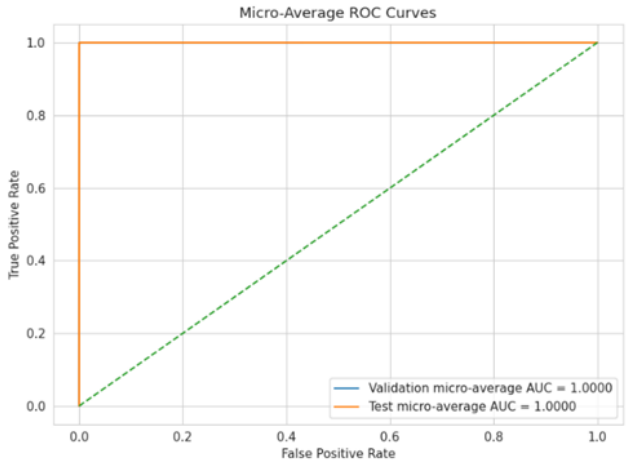

Figure 6. Micro-Average ROC Curves

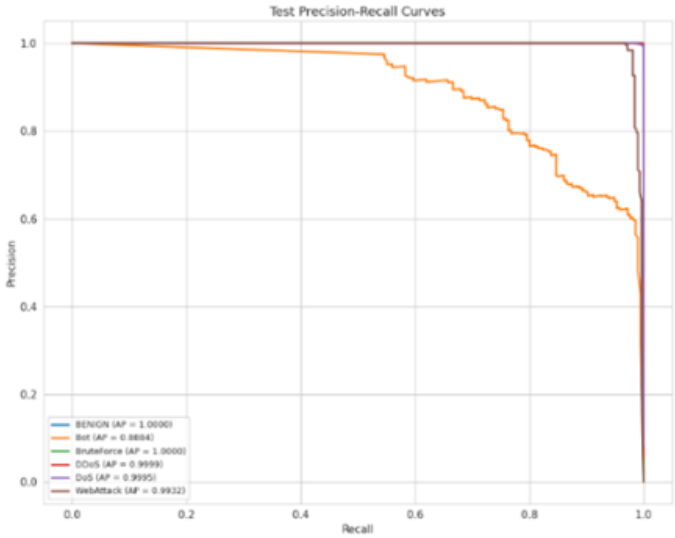

Figure 7. Test Precision-Recall Curve

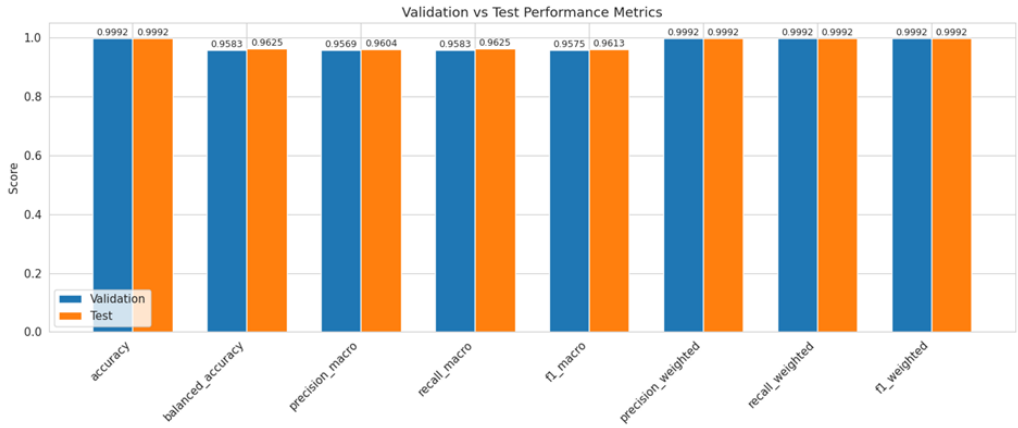

Figure 8. Validation vs Test Performance Metrics

The overall effectiveness, robustness, and generalization ability of the proposed intrusion detection framework are illustrated with figures 4 - 8. The validation and test ROC curves (Figures 4 and 5) demonstrate class discrimination ability with most AUC values near 1.0 for all attack categories, a value that suggests the model can successfully differentiate between the benign and malicious traffic with a very low false positive rate. The same applies to the micro-average ROC curves (Figure 6), which have an AUC of 1.0000 on both the

validation and test sets, indicating the model's overall capacity to classify and consistent performance across all classes. The effectiveness of the framework can also be verified by the Precision–Recall analysis in Figure 7, which further shows that most of the attack classes can obtain close to perfect average precision scores in the case of imbalanced multiclass intrusion detection. The detection performance of the Bot class is satisfactory as it has a relatively smaller number of samples and is more similar to the normal traffic. Also, Figure 8 provides a comparison of the performance of the validation and test sets, and illustrates that the differences between these two sets are very small.

The model exhibits an accuracy of 99.92%, macro-averaged precision, recall and F1-scores above 96% and a balanced accuracy of more than 95%, indicating a good generalization with no sign of overfitting. Overall, the results indicated that the proposed framework yielded high accuracy, stability, and reliability in intrusion detection performance, both in the majority and minority attack classes, and hence would be a good solution for real-world cybersecurity applications.

Comparison of Models:

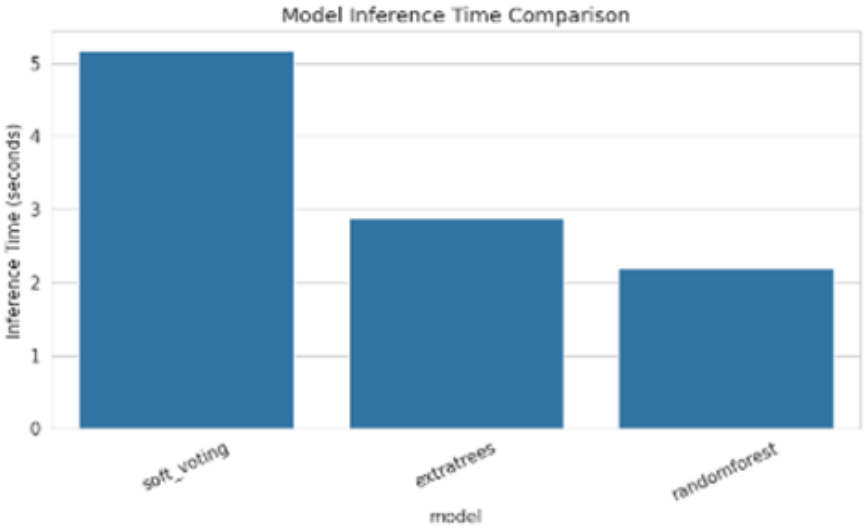

Figure 9. Model Interference Time Comparison



Figure 10. Model Training Time Comparison

To assess the effectiveness of the proposed soft voting ensemble, the performance of each single machine learning model, such as Random Forest (RF) and Extra Trees (ET), was compared. The results showed that the individual classifiers had good classification accuracy, but the ensemble model demonstrated superior robustness and detection performance by

leveraging the strengths of various classifiers. The computational analysis reveals a trade-off between performance and efficiency: the individual models are more efficient in the sense that they require less time during training and less inference time, while the ensemble model is more expensive because it is the combination of several predictions. The balanced accuracy and the F1-score were better for the ensemble approach, despite this overhead, showing the robustness of the ensemble approach in the cases of both majority and minority attack classes. The comparison also shows that single accuracy might not be a proper measure for IDS performance; instead, balanced accuracy and F1-score are better measures of the IDS effectiveness. Overall, the results validate that the proposed ensemble framework offers an optimal detection accuracy, robustness and usability for practical real-world IDS implementation.

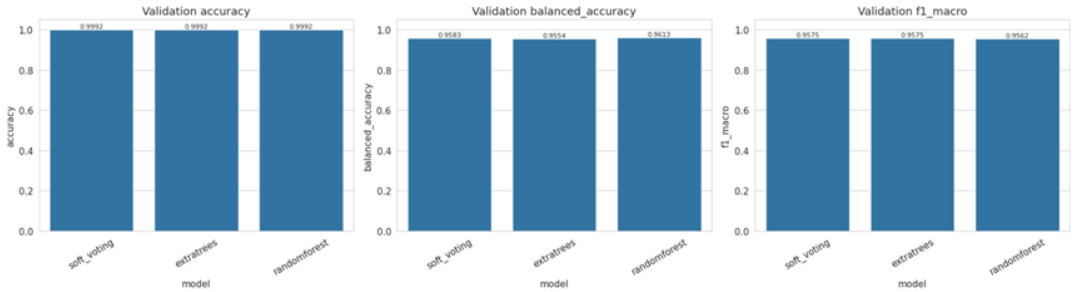

Figure 11. Validation Accuracy, Balanced-Accuracy, Validation f1_macroAccuracy, Validation f1_macro

Ablation Study:

The contribution of each component was assessed in a proposed framework through an ablation study. The results show that the performance of intrusion detection is improved in every phase repeatedly. The accuracy was enhanced from 99.41% to 99.92% and the macro F1-score was improved from 90.96% to 96.13% with the baseline model without applying any features. The feature standardization and random oversampling further improved the feature consistency and class balance, which resulted in significant improvements in precision, recall, and F1-scores. Further, clustering-based statistical feature engineering (SFE) was found to be useful to capture the hidden traffic patterns, while PCA helped to reduce the number of redundant features and improve classification efficiency. The soft voting ensemble achieved the best performance with 99.92% accuracy, macro precision of 96.04%, macro recall of 96.25% and macro F1-score of 96.13%. These results showed that the preprocessing, feature engineering, dimensionality reduction and ensemble learning process is very necessary to get the good and reliable performance for intrusion detection. The detailed performance metrics after each ablation step are reported in table 4, and show the incremental contribution of each component. Figure highlights the effectiveness of the entire pipeline in terms of the progressive improvement of the model performance in the various stages.

Table 4. Ablation Study of Multiple Models

Configuration	**Accuracy (%)**	**Macro Precision (%)**	**Macro Recall (%)**	**Macro F1-score (%)**
Baseline ML (Raw Features)	99.41	90.82	91.35	90.96
+ Standardization	99.53	92.14	92.66	92.33
+ Random Oversampling	99.68	93.87	94.51	94.13
+ Clustering-based SFE	99.79	95.01	95.43	95.18
+ PCA	99.86	95.62	95.94	95.77
+ Soft Voting Ensemble (Final)	99.92	96.04	96.25	96.13

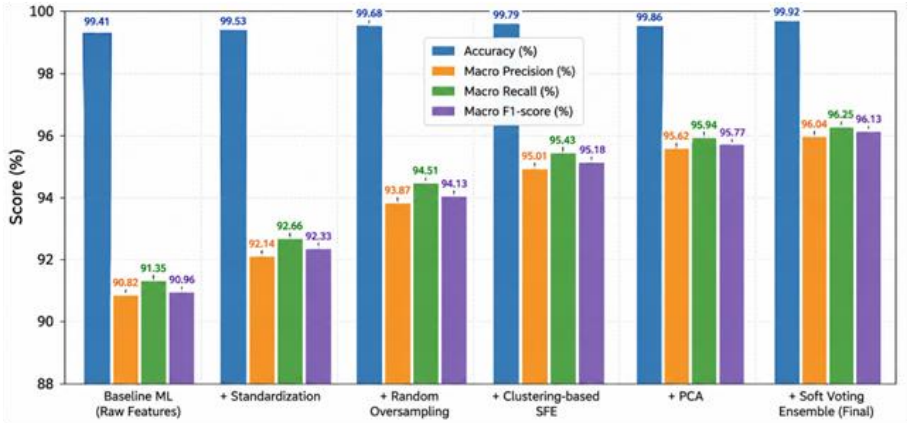

Figure 12. Ablation Study

Discussion:

The experimental findings reveal that the proposed machine learning-based intrusion detection framework is more effective than others in detecting majority and minority attack classes in complex network conditions. The overall accuracy of the baseline model is high, though its inferior macro precision, recall, and F1-score indicate that the model struggles with unbalanced data and locating less frequent types of attacks. This demonstrates the significance of employing advanced preprocessing and feature optimization techniques to make classification more than merely looking at accuracy. The gradual increase in the ablation study indicates that every section of the proposed pipeline functions as intended. Standardization of data enhances scaling of features and stability of model training.

Random oversampling, conversely, corrects imbalance in classes and in this way, minority attack classes are more likely to be found. In addition, clustering-based statistical feature engineering can be combined with other techniques to have a much stronger feature representation by uncovering latent patterns and relationships in the data. This assists the model in distinguishing between normal and bad traffic, particularly with large, complex data sets. Principal Component Analysis (PCA) takes the feature space a step further by eliminating irrelevant and redundant features. This simplifies the calculations and retains valuable information. This not only makes the model work better, but also makes it generalise better.

The final combination of soft voting ensemble model that incorporates the use of Random Forest, Extra Trees, and XGBoost performs the best on all the evaluation metrics. This demonstrates that ensemble learning is superior to single-model methods in that it reduces model bias and variance resulting to more consistent and precise predictions. One of the important contributions of the study is the development of a unified and integrated pipeline which is unlike most of the available intrusion detection systems. The past studies have mainly focused on discrete components such as classification algorithms or feature selection methods, but they tend to be disjointed and lack a unified framework that methodically incorporates preprocessing, feature engineering, dimensionality reduction, and ensemble learning. The proposed approach, however, integrates all these components into one pipeline, ensuring every step contributes to the overall performance. Such an end-to-end design significantly enhances the accuracy, robustness, and scalability of the system and is more suitable to be used in a real-world scenario such as cloud computing and IoT networks. Moreover, the fact that the results of the validation and test are very similar demonstrates that the proposed model is quite general and is not over fit.

The suggested system is efficient and trustworthy since it is capable of categorizing any type of attack, common or uncommon. However, despite these encouraging results, there are certain issues that should be addressed. The model might require additional real-time testing and with varying attack conditions to ensure that it can be adapted. In addition, the

explainable AI approach might simplify and streamline the decision-making process and make it more trustworthy. In general, the findings indicate that the combination of preprocessing, feature optimisation, and ensemble learning into one system is a highly efficient method of detecting modern intrusions. The proposed approach not only achieves highly precise results, but also ensures that performance is balanced, is computationally efficient, and strong, which is an issue with existing IDS models.

Conclusion:

To tackle the challenges of multiclass and imbalanced network traffic classification, this study proposes a complete multi-stage machine learning-based intrusion detection framework which consists of four components data standardization, random oversampling, clustering-based statistical feature engineering (SFE), and [4] soft voting ensemble. The proposed framework is evaluated through experiments on CICIDS2017, and the results show that the framework achieves 99.92% accuracy, 96.04% macro precision, 96.25% macro recall and 96.13% macro F1-score. The class-wise results validate good detection performance for the major attack categories, and the ablation study highlights the contribution of each component of the frameworks to the overall detection performance. Moreover, the soft voting ensemble has the advantage of enhancing the stability of prediction because Random Forest, Extra Trees, and XGBoost are able to complement each other. Additionally, the soft voting ensemble has the benefit of enhancing the stability of the prediction, as Random Forest, Extra Trees, and XGBoost are able to complement each other. Overall, the results showed that the proposed framework provides an effective and strong solution in handling multiclass intrusion detection in a complex and imbalanced network environment. Future research efforts will include the incorporation of Explainable AI (XAI) methods to increase the transparency of the model, expanding the framework to real-time IDS and testing the adaptability of the framework with a variety of and changing cybersecurity datasets to ensure resistance to new and zero-day threats.

References:

[1] M. Althunayyan, A. Javed, and O. Rana, "A survey of learning-based intrusion detection systems for in-vehicle networks," *Comput. Networks*, vol. 277, Mar. 2026, doi: 10.1016/j.comnet.2026.112031.

[2] Y. Ha and K. Kim, "Operationally Constrained Zero-Day Intrusion Detection with Target-FPR Calibration and Similarity Graph Construction," *Appl. Sci.*, vol. 16, no. 5, p. 2284, Mar. 2026, doi: 10.3390/APP16052284/S1.

[3] M. S. Harish, S. Lokesh, P. Sakthivel, and B. Akshaya, "Hybrid deep learning model for network intrusion detection using optimal feature fusion," *Ain Shams Eng. J.*, vol. 17, no. 1, p. 103904, Jan. 2026, doi: 10.1016/J.ASEJ.2025.103904.

[4] R. Ponnumani, N. Vasudeva, T. Elumalai, P. Kaliyaperumal, B. Balusamy, and F. Benedetto, "A multi-stage framework for scalable and context-aware intrusion detection in IoT-cloud systems using deep latent modeling and graph-based attack classification," *Comput. Electr. Eng.*, vol. 131, p. 110949, Mar. 2026, doi: 10.1016/J.COMPELECENG.2026.110949.

[5] Y. Zhu, Y. Wang, L. Zhou, and Y. Xia, "FC-Trans: Deep learning methods for network intrusion detection in big data environments," *Comput. Secur.*, vol. 154, p. 104392, Jul. 2025, doi: 10.1016/J.COSE.2025.104392.

[6] L. K. S. Kumar *et al.*, "Anomaly-based intrusion detection on benchmark datasets for network security: a comprehensive evaluation," *Sci. Reports 2026 161*, vol. 16, no. 1, pp. 8507-, Mar. 2026, doi: 10.1038/s41598-026-38317-w.

[7] Y. Sun and Z. Wang, "Intrusion detection in IoT and wireless networks using image-based neural network classification," *Appl. Soft Comput.*, vol. 177, p. 113236, Jun. 2025,

doi: 10.1016/J.ASOC.2025.113236.

[8] E. C. P. Neto, S. Iqbal, S. Buffett, M. Sultana, and A. Taylor, "Deep learning for intrusion detection in emerging technologies: a comprehensive survey and new perspectives," *Artif. Intell. Rev. 2025 5811*, vol. 58, no. 11, pp. 340-, Aug. 2025, doi: 10.1007/S10462-025-11346-Z.

[9] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion Detection Framework for Internet of Things with Rule Induction for Model Explanation," *Sensors 2025, Vol. 25, Page 1845*, vol. 25, no. 6, p. 1845, Mar. 2025, doi: 10.3390/S25061845.

[10] A. V. Turukmane and R. Devendiran, "M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning," *Comput. Secur.*, vol. 137, p. 103587, Feb. 2024, doi: 10.1016/J.COSE.2023.103587.

[11] V. Kantharaju, H. Suresh, M. Niranjanamurthy, S. I. Ansarullah, F. Amin, and A. Alabrah, "Machine learning based intrusion detection framework for detecting security attacks in internet of things," *Sci. Reports 2024 141*, vol. 14, no. 1, pp. 30275-, Dec. 2024, doi: 10.1038/s41598-024-81535-3.

[12] Y. Huang, J. Gou, Z. Fan, Y. Liao, and Y. Zhuang, "A multi-label network attack detection approach based on two-stage model fusion," *J. Inf. Secur. Appl.*, vol. 83, p. 103790, Jun. 2024, doi: 10.1016/J.JISA.2024.103790.

[13] J. Azimjonov and T. Kim, "Stochastic gradient descent classifier-based lightweight intrusion detection systems using the efficient feature subsets of datasets," *Expert Syst. Appl.*, vol. 237, p. 121493, Mar. 2024, doi: 10.1016/J.ESWA.2023.121493.

[14] A. S. Almuflih *et al.*, "Securing IoT devices with zero day intrusion detection system using binary snake optimization and attention based bidirectional gated recurrent classifier," *Sci. Reports 2024 141*, vol. 14, no. 1, pp. 29238-, Nov. 2024, doi: 10.1038/s41598-024-80255-y.

[15] P. Waghmode, M. Kanumuri, H. El-Ocla, and T. Boyle, "Intrusion detection system based on machine learning using least square support vector machine," *Sci. Reports 2025 151*, vol. 15, no. 1, pp. 12066-, Apr. 2025, doi: 10.1038/s41598-025-95621-7.

[16] S. Kaushik *et al.*, "Robust machine learning based Intrusion detection system using simple statistical techniques in feature selection," *Sci. Reports 2025 151*, vol. 15, no. 1, pp. 3970-, Feb. 2025, doi: 10.1038/s41598-025-88286-9.

[17] C. Christy, A. Nirmala, A. M. O. Teena, and A. I. Amali, "Machine learning based multi-stage intrusion detection system and feature selection ensemble security in cloud assisted vehicular ad hoc networks," *Sci. Reports 2025 151*, vol. 15, no. 1, pp. 27058-, Jul. 2025, doi: 10.1038/s41598-025-96303-0.

[18] J. Mao, X. Yang, B. Hu, Y. Lu, and G. Yin, "Intrusion Detection System Based on Multi-Level Feature Extraction and Inductive Network," *Electron. 2025, Vol. 14, Page 189*, vol. 14, no. 1, p. 189, Jan. 2025, doi: 10.3390/ELECTRONICS14010189.

[19] Y. Cai, J. Zuo, M. Fan, C. Zhao, and Y. Lu, "An Intrusion Detection System for the CAN Bus Based on Locality-Sensitive Hashing," *Electron. 2025, Vol. 14, Page 2572*, vol. 14, no. 13, p. 2572, Jun. 2025, doi: 10.3390/ELECTRONICS14132572.

[20] F. Al Tfaily, Z. Ghalmane, M. el A. Brahmia, H. Hazimeh, A. Jaber, and M. Zghal, "Graph-based federated learning approach for intrusion detection in IoT networks," *Sci. Reports 2025 151*, vol. 15, no. 1, pp. 41264-, Nov. 2025, doi: 10.1038/s41598-025-25175-1.

[21] N. U. Ain, M. Sardaraz, M. Tahir, M. W. Abo Elsoud, and A. Alourani, "Securing IoT Networks Against DDoS Attacks: A Hybrid Deep Learning Approach," *Sensors 2025, Vol. 25, Page 1346*, vol. 25, no. 5, p. 1346, Feb. 2025, doi: 10.3390/S25051346.

[22] Z. kadhim Alitbi, S. A. Hosseini Seno, A. Ghaemi Bafghi, and D. Zabihzadeh, "A

Generalized and Real-Time Network Intrusion Detection System Through Incremental Feature Encoding and Similarity Embedding Learning," *Sensors 2025, Vol. 25, Page 4961*, vol. 25, no. 16, p. 4961, Aug. 2025, doi: 10.3390/S25164961.

[23] A. A. Abu-Shareha, M. M. Abualhaj, A. Hussein, O. Almomani, A. Amer, and A. Achuthan, "Supervised machine learning intrusion detection review and multi-criteria evaluation," *Sci. Reports 2026 161*, vol. 16, no. 1, pp. 14525-, Mar. 2026, doi: 10.1038/s41598-026-44773-1.

[24] A. Kutlimuratov *et al.*, "A Lightweight Cascade-Based Farmework for Real-Time Zero-Day Attack Detection," *Comput. 2026, Vol. 15, Page 174*, vol. 15, no. 3, p. 174, Mar. 2026, doi: 10.3390/COMPUTERS15030174.

[25] R. Xie and D. Liu, "A novel hybrid graph neural network and transformer model for intrusion detection," *Peer-to-Peer Netw. Appl. 2026 192*, vol. 19, no. 2, pp. 59-, Mar. 2026, doi: 10.1007/S12083-025-02171-W.

[26] P. Appiahene *et al.*, "Network intrusion detection using a hybrid graph-based convolutional network and transformer architecture," *PLoS One*, vol. 21, no. 1, p. e0340997, Jan. 2026, doi: 10.1371/JOURNAL.PONE.0340997.

[27] R. Panigrahi and S. Borah, "A detailed analysis of CICIDS2017 dataset for designing Intrusion Detection Systems," *Int. J. Eng. Technol.*, vol. 7, no. 3.24, pp. 479–482, 2018, doi: 10.14419/IJET.V7I3.24.22797.



Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.