RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt26411

Improved Hybrid Model-Based Machine and Deep Learning Approach for Intrusion Detection System

Premananda Sahu [a], Varsha Himthani [b,*], Ashwani Kumar [b]

[a] School of Computer Science and Engineering, Lovely Professional University, Punjab, India

[b] School of Computer Science Engineering & Technology, Bennett University, Greater Noida, Uttar Pradesh, India

* Corresponding Author Email: varsha.himthani@gmail.com

DOI: https://doi.org/10.54392/irjmt26411

Received: 12-12-2025; Revised: 08-06-2026; Accepted: 25-06-2026; Published: 08-07-2026

Abstract: The Intrusion Detection Systems (IDSs) are very important tools for defending a network against emerging cyber threats. This study proposes the hybrid intrusion detection system model of extreme gradient boosting (XGBoost) and KMeans++ clustering algorithm to balance the trade-off between accuracy, efficiency, and robustness in detecting malicious traffic. XGBoost algorithms are good for structured problems where classification problems occur, whereas KMeans++ helps you to get more clustering accuracy by helping centroid initialization. For enhancing the performance of the model some feature extraction steps and data preprocessing steps like normalization, encoding, Synthetic Minority Over-sampling Technique (SMOTE) based imbalance data balancing were considered. The system was trained and validated on Canadian Institute for Cybersecurity Intrusion Detection System (CICIDS) 2017 and put to test in various metrics: accuracy, precision, recall, F1-score, ROC-AUC and false positive rate. Comparative analysis was performed using traditional machine learning models such as SVM, Decision Tree, Random Forest, Naive Bayes and deep learning architectures which include CNN, LSTM and Auto-Encoder. It was found to be high detection accuracy of 99.87% with very low FPR by far i.e. 0.1%. The model provided high recall and precision in different types of attack and successful overfitting resistance could be confirmed using 10-fold cross-validation, XGBoost regularization and structure clustering. This work shall play an important role in improving hybrid models to minimize alert fatigue with trustworthy threat classification in the real operational traffic.

Keywords: Intrusion Detection Systems, XGBoost, KMeans++, CICIDS2017, CNN, RNN, LSTM, SVM.

1. Introduction

New technologies are being developed at an accelerate rate that is quite breathtaking, and the internet is now intermingled with every facet of our lives. This phenomenon happens especially in such areas as finance, healthcare, defense, and any other critical infrastructure that uses networking systems [1]. These improvements have made such systems the primary target of cyberattacks, which have exploded quickly in both scope and complexity. Cyber threats such as phishing, ransomware, Advanced Persistent Threats (APTs) and Distributed Denial of Service (DDoS) attacks cause critical damage to organizational and personal assets.

One of the leading protective schemes against such threats are Intrusion Detection Systems. As the name implies, these systems are designed to monitor, detect and respond to activities that are considered suspicious within a network or system. A traditional classification of IDS methods is inclined to be signature based and anomaly-based systems [2]. While signature-based systems rely on previously-documented attack patterns, they amount to nothing against zero-day exploits. On the divergent, the anomaly-based systems are capable of learning the network traffic and then discerning new or unknown threat [3]. Unfortunately, such systems tend to suffer from very high false alarm rates, which compromises their reliability. Machine Learning (ML) techniques along with Deep Learning (DL) powered framework have been utilized to further improve the IDS effectiveness as these technologies are capable of understanding accurately of sophisticated and novel attack patterns [4]. ML algorithms such as Support Vector Machines (SVM), Decision Trees, Random Forests and Naive Bayes have been applied to IDS due to their prediction capabilities and relatively lower computational cost. Identifying features relevant to the problem from a superset of data is the major concern of DL methods [5].

Though ML and DL alone have made great strides in this regard, it still has its limitations. While ML models have issues with nonlinear relationships, they also have issues with noise and high dimensions.

Powerful as they may be, DL models often are limited by large scale computing resources, and for real time applications, there are challenges of their own [6]. Furthermore, both approaches are prone to overfitting which is worsened by poor tuning, which leads to high false-positive rates. These problems have led to the development of hybrid models. This approach can help in improving generalization and to obtain higher robustness by combining multiple algorithms [7]. In this Work, we are proposing a hybrid Intrusion Detection system from Extreme Gradient Boosting and KMeans++ Clustering. XGBoost is known for its efficient and scalable gradient boosting framework and it is considered a benchmark in diverse classification tasks [8]. KMeans++ is an attempt to overcome some of the weaknesses of the KMeans algorithm by changing the centroid initialization process to make it more stable and improve the clustering performance. The framework describes the justification of the integration of XGBoost with KMeans++ which is to get the global and local feature interdependencies of network traffic data. KMeans++ creates some of the recognizable structural groupings in the data set, which helps in the addition of other cluster-based features [9] further improves XGBoost achievement in the classification stage.

The effectiveness of the suggested hybrid model is tested against the CICIDS2017 dataset that contains richly simulative and multifaceted intrusion attempts built using real network environments [10]. This dataset is comprehensive for assessing intrusion detection system performance as it contains normal as well as diverse attack traffic, including Brute Force, Botnet, DDo, and Web Attacks.

The primary aim of this study is to develop and optimize a hybrid Intrusion Detection System by incorporating Extreme Gradient Boosting and KMeans++ Clustering to enhance the accuracy, efficiency, and reliability of network intrusion detection. The primary aims focus on the following:

- Develop a hybrid IDS architecture based on dimensionality reduction, structural clustering and ensemble classifier using boosting.
- Investigate the effectiveness of using cluster derived structural information as additional features for intrusion detection performance improvement.
- Evaluate the proposed hybrid model with the CICIDS2017 dataset and compare the performance of the model with a few widely used machine learning and deep learning algorithms.
- Evaluate the detection capability of the proposed system under different types of intrusion using standard evaluation metrics.

The growing complexity of cyber-attacks and rapid growth of network-based services have caused great challenges for the intrusion detection systems of today. Traditional signature-based IDS methods work well in detecting known attacks but fail in many cases to see previously unseen attacks. On the other hand, anomaly-based detection methods can detect new and novel attacks but often have high false positive rates. In recent research, the application of ML & DL algorithms have used to enhance the accuracy of the identification process; however, many models previously developed still face challenges associated with high dimensionality in the feature spaces, unbalanced data, and the lack of capability in capturing the structural patterns in network traffic. In particular, purely supervised approaches depend largely on organized data and not identify subtle structural anomalies that are present in complex networks. These challenges are the motivation for the development of hybrid IDS frameworks that include supervised classification as well as unsupervised structural learning. By combining clustering-based structural information and ensemble learning models, capturing hidden traffic patterns that may enhance performance of intrusion detection with computationally efficient methods becomes possible.

Although some studies have focused on the investigation of hybrid intrusion detection systems that unite machine learning and deep learning approaches, some important research gaps still remain unsolved. Many of the existing IDS frameworks are based on supervised classification algorithms or anomaly detection based on clustering, however, the studies that focus on integrating the information of structural clustering into boosted ensemble classifiers are very few, while at the same time, few works are done to reduce data dimension and address the problem of class imbalance in a unified pipeline. In particular, previous IDS studies based on XGBoost have based their learning mainly on the feature level and do not exploit latent structural relationships provided within network traffic data. Furthermore, clustering algorithms such as KMeans have been used mostly only for anomaly detection and are rather rarely used for ensemble classifiers as structural feature augmentation mechanism. As a result, many IDS frameworks are unable to detect hidden groupings of traffic that can improve discrimination between benign and malicious network activities. To mitigate this shortcoming, the current research introduces a framework of a hybrid IDS based on PCA-based feature compression, imbalance handling based on SMote, KMeans++ structural clustering and boosted classification based on XGBoost and in a unified learning pipeline. The proposed framework injects the structural information obtained from clusters into the supervised classification stage, and lets the model learn the representation of network traffic at the feature level and the structure level at once. This combination is done to improve detection performance with computational efficiency for large-scale intrusion detection data sets.

This paper is designed as follows: Section II offers a complete review of present literature in IDS, Section III details the methodology and model architecture, Section IV describes experimental results and analysis, and Section V concludes the paper with intuitions into future investigation directions.

2. Literature Review

The literature review is mandatory to comprehend the current development and detect gaps in the IDS. It provides an insight into the methodologies, datasets, and evaluation strategies employed by researchers that were done in the past. The corresponding Work reviewed establishes the basis of the target hybrid model design and supports the necessity thereof in the modern sphere of cybersecurity. Through the analysis of the current machine learning, deep learning, and hybrid systems, the review sheds light on the gaps in the research field, gives the momentum to the development of more efficient models, and places the utilization of more sophisticated datasets like CICIDS2017 in the limelight of better detection and improving the real-time response.

L Ashiku, and C. Dagli [11] have introduced DL based network intrusion detection system from simulated network traffic data and semi-dynamic hyper parameter tuning, with more than 95 percent accuracy. Future work includes feature reduction, transfer learning using UNSW-NB15, bootstrapping for balanced datasets, to improve detection of the zero-day attacks and better adaptive cybersecurity flexibility. Y Pourardebil Khah *et al.* [12] have formulated a new hybrid ML-based algorithm for feature selection for IDS in distributed Internet of Things (IoT) environments with the main goal of promoting early detection of attacks with high accuracy metrics. The overall strategy consists of several stages: data preprocessing, feature ranking by heuristic and meta-heuristic methods, followed by optimization with a new Discrete Gray Wolf Optimization Algorithm. Tested on NSL-KDD and UNSW-NB15 datasets, the model outperformed other models with noticeable improvements of up to 15.85% in precision and enhanced metrics of accuracy, recall, specificity, and F-cost across multiple scenarios. M Sajid *et al.* [13] have stated that the rapid growth of the Internet of Things (IoT), cloud technology, and automotive computing networks has elevated the amount of data traffic as well as increased the level of security risks, which in turn, increases the need for more sophisticated intrusion detection systems. In this paper, the authors have proposed an improved IDS using a hybrid model of XGBoost, CNN, and LSTM aimed at boosting the accuracy of detection and defense on IoT systems. The model uses four benchmark datasets, CICIDS2017, UNSW-NB15, NSL-KDD, and WSN-DS to complete feature selection and classification of binary and multiclass tasks. The high rates of accuracy of the detection and low rates of false acceptance have been observed, which proves the effectiveness of the model, and proves the potential to adjust to the dynamic cyber threats.

Huang *et al.* [14] have improved the study by Industrial Internet of Things (IIoT) security by developing a privacy-preserving data federated learning (FL)-based network intrusion detection system with attention convolutional neural networks and variational autoencoders. It enables better performance concerning detection. With the FL framework, multiple IIoT clients can now perform collaborative model training without exposing raw data. Using an actual IoT dataset, this model outperformed traditional local training approaches and benchmarks in terms of accuracy, precision, and false positive rate. Fenjan *et al*. [15] have indicated that advances in CC technologies, networks, data, and the Internet of Things (IoT) have transformed life in the 21st century. These unprecedented changes come with severe challenges. In this case, the advanced persistent threat (APT) is the advanced threat that learns from user patterns. Therefore, traditional intrusion detection systems cannot rely solely on fixed algorithms. IADs also require detection algorithms that increase adaptation capacity. This is also a challenge in cybersecurity. This study proposes a CNN-based adaptive IDS enhanced with ANNs and MLPs to adapt to evolving detection requirements of complex systems for better accuracy and scalability. The model was tested on a complex dataset for diverse network scenario classification and achieved 96% accuracy. This advanced model effectively counters sophisticated cyber threats, highlighting the necessity of deploying them during critical operations. Saleh *et al*. [16] have found that the problems of intrusion detection in cyber-physical systems' Wireless Sensor Networks (WSNs) are solved using the machine learning algorithms Gaussian Naive Bayes and Stochastic Gradient Descent. Additionally, performance is improved with context-aware recommendation systems and dimensionality reduction techniques (PCA and SVD). The SG-IDS model achieved 98% accuracy on the WSN-DS dataset and performed well on an IoMT dataset, attaining 87% accuracy and 100% precision in intrusion detection.

Chen *et al.* [17] have stated that VAN-IDS is a novel hybrid algorithm that utilizes Dempster–Shafer theory to fuse results from Bi-LSTM-based packet analysis and LightGBM-based analysis of physical features, thus improving VANET security. It uses F2MD data and detects both DoS and Sybil attacks with an accuracy exceeding 98.5%. To improve privacy and efficiency, model training in validated cloud architecture is done through RSUs in a federated learning setting, leading to VAN-FED-IDS. Model validation in the Flower and FedTree frameworks demonstrated real-time responsiveness without compromising data privacy. Najar [18] has indicated that Network security faces increasing threats from cyberattacks, particularly DDoS,

in the digital era. Despite ML and DL advancements, DDoS detection remains a challenge because of data imbalance and intricate system architectures. This Work proposes a robust feature selection-based intrusion detection model tested on the CICDDoS2019 dataset. With an accuracy of 96.82% and precision of 96.76%, coupled with a swift prediction time of 0.189 ms, the model outperformed other methods and baselines, thus proving its effectiveness in modern DDoS attack detection and classification. Mohammad *et al.* [19] have focused focuses on an IDS's pivotal importance in cybersecurity and seek to improve traditional machine learning techniques by employing deep learning frameworks. Data augmentation is applied to four primary datasets, including CIC-IDS-2017, to improve model performance. The results indicate that even basic CNN models achieve high accuracy attack detection (up to 91%), but more advanced models provide diminishing returns. The results indicate that deep learning greatly enhances the effectiveness of an IDS in detecting advanced persistent threats when coupled with high-quality and well-distributed data.

Gou *et al.* [20] have indicated that a new method employing an improved Random Forest united with KMeans++ DBSCAN was planned to improve detection accuracy and handle imbalanced data with increasing network intrusions and limited traditional IDS capability. Although with longer modeling times, tested on LUFlow and CIDDS datasets, it obtained up to 91.2% accuracy and high F1 scores. The model shows good performance for intrusion and anomaly exposure; future developments proposed by distributed and parallel computation aim to progress efficiency. Ahmed *et al.* [21] have improved network security through machine learning and deep learning-based intrusion detection. Other classifiers based on both ML and DL techniques are considered to be efficient in classification of network traffic. SVM and RF are reliable and interpretable IDS solutions, whereas LSTM and ANN perform well in terms of detecting complex and evolving threats with high motivation accuracy up to 96.02%.

While the current research has shown considerable progress in intrusion detection systems that are based on machine learning, several methodological limitations are apparent in the literature. First, many IDS models are highly dependent on supervised classification techniques without structural learning mechanisms being incorporated into them, which can detect latent traffic patterns. Second, there are a number of studies which use dimensionality reduction techniques or feature selection, but these approaches are usually used individually instead of integrated in a proper feature engineering pipeline. Third, class imbalance is a significant issue in IDS datasets and although there are currently different oversampling methods (SMOTe) to mitigate the bias, they are rarely combined with structural learning-based clustering. These observations point to the necessity of a reasonable hybrid framework that incorporates feature reduction, structure clustering and ensemble classification while preserving computational efficiency and good detection performance in multiple categories of attacks. The corresponding table is represented in Table 1.

3. Proposed Methodology

The successful implementation of the proposed methodology helps in the design of a novel Intrusion Detection System integrating hybrid Extreme Gradient Boosting and KMeans++ clustering. This phase has provided a thorough explanation of each stage in the model development pipeline, from data preprocessing, to Clustering and then classification. Moreover, mathematical insight is offered for each component with the aim of reproducibility and theoretical emphasis. The total architectural diagram described in this research is expressed in Figure 1.

- Initially, the data was fetched from the CICIDS2017 dataset.
- Next, the new dataset is formed from the old one based on data preprocessing, which is further divided into data cleaning, label encoding, normalization, and handling the imbalanced data, if any.
- The training process has been held from Clustering, based on KMeans++ and classification, based on XGBoost.
- For the removal of both overfitting and underfitting, the k-fold cross-validation technique has been adopted.
- Next, a comparison of all the ML & DL techniques has been employed.
- Evaluation has been done based on the parameters.
- Finally, from the hybrid technique, intrusion has been detected.

The architectural framework shown in Fig. 1 shows the full processing pipeline of the proposed hybrid intrusion detection system. The following figure shows the relationship of the various phases of the methodology for converting the raw network traffic data into final intrusion predictions. Specifically, the pipeline is made up of several preprocessing operations such as data cleaning, data normalization, encoding, and data imbalance treatment followed by dimensionality reduction and clustering-based structural learning.

3.1 Data Preprocessing

The quality of input data makes a very significant difference in the efficiency and accuracy of any machine

learning or deep learning model. Even if input data were large and complete, some preprocessing steps would be required to remove or impute missing values, normalize feature distributions, encode categorical data, and balance class distributions. This section elaborates upon the major steps taken in data preprocessing for the purpose of providing strength to the hybrid Intrusion Detection System. The CICIDS2017 dataset contains around 2.8 million network flow records with 85 numerical and categorical features representing the normal and attack traffic.

Table 1. Detailed Version of Literature Review

Ref. NO	Authors	Technology Used	Innovations	Year	Accuracy
11	Ashiku, and C. Dagli [11]	DL based network IDS, UNSW-NB15 dataset	simulated network traffic data and semi-dynamic hyper parameter tuning	2025	95
12	Pourardebil Khah *et al.* [12]	Hybrid ML-based feature selection, Discrete Gray Wolf Optimization, NSL-KDD, UNSW-NB15	Developed a novel feature ranking and optimization framework for early attack detection; achieved significant precision gains.	2025	An average of 10.60% improvement in testing dataset
13	Sajid *et al.* [13]	Hybrid XGBoost-CNN-LSTM, CICIDS2017, UNSW-NB15, NSL-KDD, WSN-DS	Improved IDS for IoT with hybrid deep and machine learning model; achieved high detection accuracy and adaptability.	2024	96.21
14	Huang *et al.* [14]	Federated Learning, Attention-CNN, Variational Autoencoder, IIoT Dataset	Developed privacy-preserving FL-based NIDS; enabled collaborative training with strong detection without data sharing.	2024	98
15	Fenjan *et al.* [15]	CNN, ANN, MLP, Adaptive IDS	Proposed adaptable IDS for APT detection; achieved 96% accuracy and effective performance in dynamic environments.	2024	96
16	Saleh *et al.* [16]	Gaussian Naive Bayes, SGD, PCA, SVD, WSN-DS, IoMT Dataset	Introduced context-aware ML-based IDS for CPS/WSN; achieved 98% accuracy and high precision in real environments.	2024	87
17	Chen *et al.* [17]	Bi-LSTM, LightGBM, Dempster–Shafer Fusion, VANET, Federated Learning	Designed VAN-IDS and VAN-FED-IDS using hybrid models with FL; achieved >98.5% accuracy while preserving data privacy.	2024	98.5
18	Najar [18]	Feature Selection-based IDS, CICDDoS2019	Addressed DDoS detection with a focus on data imbalance; achieved 96.82% accuracy and very fast detection time.	2024	96.82
19	Mohammad *et al.* [19]	CNN, Deep Learning, Data Augmentation, CICIDS2017	Enhanced ML-based IDS with DL and data augmentation; achieved high accuracy, especially for APT detection.	2024	91
20	Gou *et al.* [20]	Improved Random Forest, KMeans++, DBSCAN, LUFlow, CIDDS	Developed a hybrid ensemble model for imbalanced data; good performance with plans for distributed computation improvements.	2025	91.2
21	Ahmed *et al.* [21]	LSTM with ANN	detecting complex and evolving threats	2025	96.02

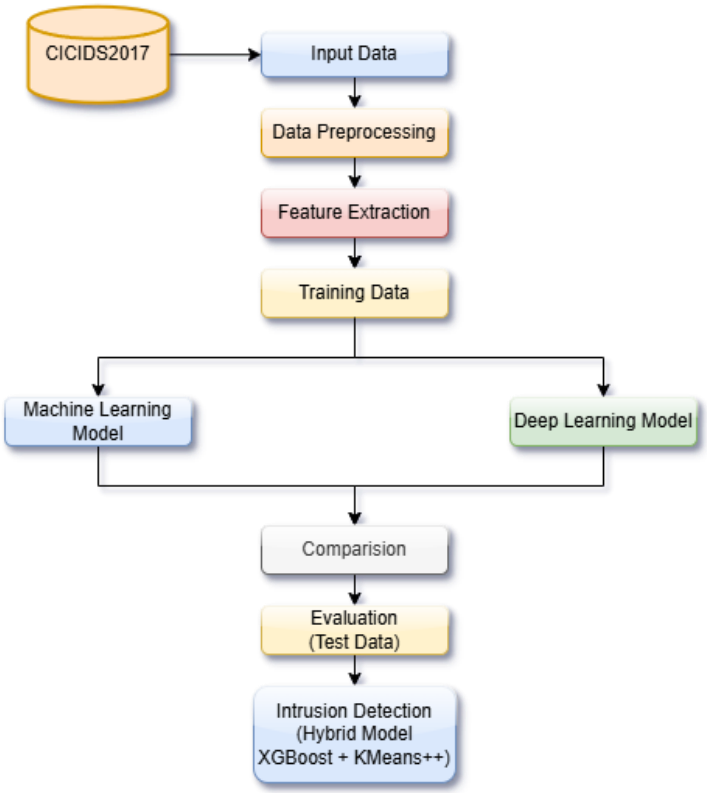

Figure 1. Total Architectural Diagram for Proposed Hybrid Intrusion Detection Framework

Table 2. Expressive Statistics of Features Before Preprocessing

Name of Features	Mean	Standard Deviation	Range
Flow Duration	112.52	98.37	0-5000
Total Forward Packets	12.62	13.95	1-150
Total Backward Packets	9.56	11.91	1-120
Forward Packet length Mean	345.48	290.32	0-1500

Normal traffic is around 80% and 20% is the attack classes such as Web Attacks, DoS and Botnet. After processing of duplicate removal, encoding, and normalization process, there is a table namely Table 2 which has represented the expressive statistics of representative selected features before preprocessing.

1. Handling Missing values- Datasets collected in real environments often suffer from null or missing values due to network inconsistencies or failed logging. Here, the missing values are identified based on;

$$M_j = \begin{cases} 1, & if\ x_j\ is\ missing \\ 0, & otherwise \end{cases} \quad (1)$$

Where x is the dataset, j is the index within that dataset, and M_j is missing value.

2. Feature Encoding-: Moreover, the dataset has some categorical features like the kind of protocol, which could be TCP, UDP, or ICMP. Machine learning algorithms need these in numerical form. Therefore, a label and one hot encoding are used.

Label and one-hot encodings are popular ways of converting categorical inputs to numerical inputs so that machine learning algorithms can work with those values. Label encoding assigns an integer to every category, while one-hot encoding creates a separate binary column for each category. Which of these two methods is applied depends upon the kind of categorical data and features of the machine learning model being developed [22].

3. Feature Normalization and Scaling-Feature normalization and scaling hold an important position in preprocessing activities for an intrusion detection system so that respective machine learning algorithms work smoothly and in an efficient manner. In the network traffic data, varying ranges of different features may include port numbers, byte sizes, and durations, and if left unprocessed, lead to bias in model training. Methods of normalization, such as

Min-Max scaling, force features to conform within a static range of zero to one, whereas the standardization procedure ensures that the features lie upon mean and standard deviation by resource, thus returning the features to a mean of zero. While performing distance calculations such as those undertaken by K-Means prior to normalization, a feature with the highest variance can arguably overshadow other features. In an overall impression, normalization of features increases detection accuracy while also providing a better chance of generalizing a particular model across different network environments. Numerical values are normalized by either Min-Max Scaling or Standardization so that features contribute equally to the training procedure of the model.

For min-max normalization,

$$y' = \frac{y - \min(y)}{\max(y) - \min(y)} \epsilon [0,1] \quad (2)$$

Where y is raw data and y/ is the transformed value of y. For standardization,

$$z = \frac{y - \mu}{\sigma} \quad (3)$$

Where μ is the mean of 0, and σ is the standard deviation of 1 respectively. In the above case both μ and σ rescaled to μ = 0 and σ = 1 which means all the features are from a proper distance-based algorithm like KMeans ++, and z is standardised feature value.

4. Imbalanced dataset Handling-In IDS, class imbalance is a very common problem, with normal traffic having a majority over intrusion or attack instances that leads to biased model training and poor detection of minority (attack) classes. Data balancing techniques help mitigate the problem by giving a fairer representation of both normal and malicious behaviors. Oversampling approaches (for example, SMOTE - Synthetic Minority Over-sampling Technique) synthesize samples for minority classes, whereas under-sampling decreases the majority classes to the size of the minority [23]. Hybrid approaches combine the two to keep data diverse without losing much useful information. A model would then better detect rare yet important intrusions and improve classification performance statistics such as recall and F1-score of attack classes, hence becoming a stronger IDS.

5. Dataset Splitting-: The dataset is stratified into training (70%), validation (15%), and testing (15%) to appropriately evaluate the model. This way, the ratios of classes are preserved across subsets. The data was randomly stratified according to a fixed random seed in order to maintain class distribution between the above splitting strategy in a reproducible way and make sampling balanced between the data.

To ensure a stringent evaluation protocol and avoid data leakage, all preprocessing and feature transformation operations were carried out strictly within the folds of training in the cross validation.

Importantly, the scaling parameters, PCA transformation and SMOTE sampling were only fitted on the training part of each fold and then applied on the validation data of the same fold. The key benefit of this protocol is that it avoids any leaking of information from the validation or testing partitions into the training stage and thus is able to present the reported performance as an accurate measure of the generalization capability of the planned IDS model. The total preprocessing has enclosed in Figure 2.

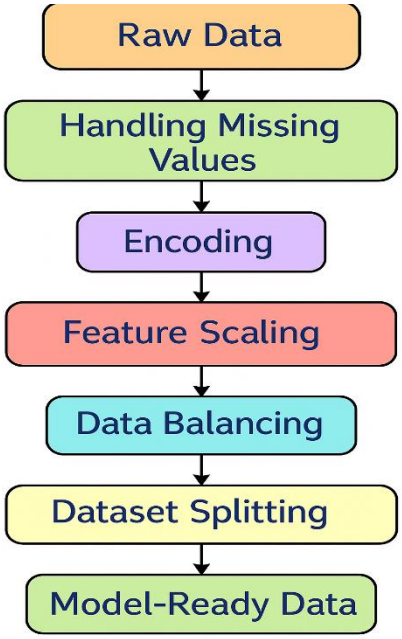

Figure 2. Preprocessing Pipeline

After pre-processing, the steps such as removal of missing values, numerical attributes normalization or categorical variable encoding, the feature distributions were more uniform and ready to be used in machine learning algorithms. Table 3 summarizes the statistical characteristics of representative features after the preprocessing stage, which shows the normalized feature ranges for model training.

In contrast to the raw dataset statistics, as shown in the above table, the normalized feature values are now within a standardized numerical range and this makes both distance-based clustering algorithms, such as KMeans++, and gradient-boosted tree models, such as XGBoost, more stable. This normalization will ensure that no individual feature will dominate the learning process because of difference in scale.

3.2. Feature Extraction

It is an important phase that lies under the data preprocessing of any machine learning pipeline.

Table 3. Expressive Statistics of Features After Preprocessing

Name of Features	Mean	Standard Deviation	Range
Flow Duration	0.48	0.22	0-1
Total Forward Packets	0.33	0.19	0-1
Total Backward Packets	0.26	0.17	0-1
Forward Packet length Mean	0.53	0.23	0-1

In intrusion detection systems, there is a lot of features associated with the raw network traffic data, which have different levels of importance, relevance, and redundancy. The feature extraction is a process of transforming the given cleaned data into a set of useful features which can be discriminative and non-redundant for better model training, faster converging time, and easy to interpret. In this research, the CICIDS2017 dataset is pre-processed for null value, normalization and categorical encoding, the CICIDS2017 dataset is passed through the feature extraction process to extract the useful abstractions from the data to be used for the accurate identification of intrusion in the network.

3.1.1 Why Does Feature Extraction Matter in IDS?

High Dimensionality: Network traffic datasets such as CICIDS2017 usually have a large number of features (85 in total). Some are irrelevant or redundant.

Noise Reduction: Raw features offer noise that might interfere with a classifier's ability to generalize.

Improved Learning Efficiency: Reducing input complexity assists models such as XGBoost to converge faster and generalize better.

Goodness of KMeans++ Clustering: More compact and representative features will lead to better cluster formation.

3.1.2 Principal Component Analysis

While several feature extraction techniques exist, for instance, Independent Component Analysis (ICA) and linear Discriminant Analysis (LDA), the method used in this study is Principal Component Analysis (PCA), essentially because it performs linear transformation, is scalable, and reduces dimensionality while preserving the variance.

PCA is suitable for the study because:

- It is compatible with both unsupervised (KMeans++) and supervised (XGBoost) approaches.
- It has a low computational complexity.
- It has the capacity to decorate input features effectively.

Thus, PCA was applied to the top 30 principal components, preserving over 95% variance.

Here, the authors are well aware that PCA is very much compatible with KMeans++ and XGBoost because of-

- XGBoost enjoys the benefit of lesser feature correlation and fewer dimensions, thus avoiding overfitting by ignoring the noise and redundant dimensions.
- Since KMeans++ performs Clustering via Euclidean distance, its performance improves in the presence of uncorrelated features that have been normalized-what PCA provides.

The final classification is then done using XGBoost based on cluster augmented features based on KMeans++ and on transformation using Principal Component Analysis (PCA). PCA based feature extraction then occupies a central middle ground between the data preprocessing and model training [24]. This hybrid approach is backed up by the presence of both KMeans++ Clustering and XGBoost classification working off of good condensed representations of network traffic. The interaction among the stages, does very well to improve the exactness of recognition, generalization with computational competence.

3.3 KMeans++ Clustering

KMeans++ represents an extended reformulation of the conventional KMeans clustering algorithm to overcome a key limitation of KMeans: improper initialization of cluster centroids. While performing Clustering at the Intrusion Detection Systems stage, especially over high-dimensional network traffic data such as the CICIDS2017 data set, improper centroid initialization can bring about suboptimal Clustering, convergence to local minimum solutions, and, therefore, inferior classification results [25]. The KMeans++ algorithm drastically brings about enhancements in Clustering: it ensures that the initial clusters are chosen sufficiently apart from each other, equating a faster convergence and better quality of clusters. KMeans, traditionally, partitions the data into k clusters such that a minimum of WCSS (within-cluster sum of squares) is achieved. However, it is highly sensitive to the placement of cluster centroids at the

beginning. KMeans++ reports the randomness of initialization by employing probabilistic distance-based selection of initial centroids. Here, the probability of selecting a point x € X as the next centroid is:

$$P(x) = \frac{D(x)^2}{\sum_{x' \in X} D(x')^2} \quad (4)$$

Where X is the dataset, and x is its elements, x/ is represented centroid from the chosen centroid C, and D is the Euclidian distance.

Clustering can be used for input features or outlier detection. The Clustering would further refine the integrity of a hybrid XGBoost classifier in cursorily differentiating distinct types of attacks with high accuracy and low false positive rates.

3.4 XGBoost

Extreme Gradient Boosting is basically the implementation of gradient-boosting decision trees with a strong focus on speed, performance, and scalability. It is quite powerful as a supervised machine-learning algorithm for both classification and regression problems. In the context of Intrusion Detection Systems, XGBoost can excel at nonlinear relationships in high-dimensional datasets and thus is best suited for the recognition of complex patterns of cyber-attacks [26]. XGBoost tries to build an ensemble method which constructed in a sequential way; that is, each subsequent tree will try to correct the errors of the previous trees. The technique tries to minimize a particular loss function by adding new trees that predict the residuals, or errors, of prior trees.

The objective function of this particular technique is expressed as follows:

$$L(\emptyset) = \sum_{j=1}^{n} l\left(y_j, \hat{y}_j^{(t)}\right) + \sum_{t=1}^{T} \Omega\left(f_t\right) \quad (5)$$

Where y_j is an actual label, $\hat{y}_j$ is a true label, $\Omega\left(f_t\right)$ is the penalty for complex tree t, and T is the total number of trees.

XGBoost has two regularization terms present in its objective function:

L1 Regularization (Lasso): Encourages sparsity in the model (fewer trees and features).

L2 Regularization (Ridge): Penalizes large weights to improve generalization.

Such act to avoid overfitting, especially with intrusion detection datasets, which are high dimensional-like CICIDS2017.

3.5. Hybrid Model

To tackle the imposed shortcomings of individual learning paradigms in Intrusion Detection Systems, this approach proposes a hybrid approach of XGBoost along with KMeans++, a fine clustering scheme. There lie the reasons behind the joint use of B-age Clustering in the unsupervised discernment of implicit patterns and a supervised learning for the fine-grained classification of the network intrusions. In such high-dimensional network datasets as CICIDS2017, anomalies and attack vectors display their latent structure (by Clustering) and their observable feature-based behaviors (by learning by classifiers) which leads to improvements in the accuracy, reduce false positive and generalizes detection from both sides across the board [27].

In the proposed hybrid model, KMeans++, which is applied on PCA-reduced data, is used to identify latent structural groupings of network traffic data. The clustering mechanism assigns a cluster label to each data instance which is related to spatial patterns and relationships. This cluster label is then added to the feature vector as an additional feature forming an augmented input matrix. This enriched dataset with structural cluster information and original PCA features are then fed to XGBoost classifier for the training. The added context of cluster information enables XGBoost to generalize better and in consequence increases its accuracy in discerning complex intrusion patterns. Another block of proposed hybrid intrusion detection includes cluster IDs of KMeans++ as additional features to the original feature set. Specifically, after dimension reduction has been carried out using PCA, the KMeans++ algorithm is applied to this reduced feature matrix to produce cluster labels. Such labels give the structurally grouped data instances depending on the similarity measure. Each data point is given a cluster ID which is then appended to its original PCA-transformed feature vector, hence giving the augmented feature matrix.

To validate the incorporation of the proposed hybrid system, the equation is:

If X is the attributes for PCA and C is the cluster label, then the final input matrix to the classifier comes to be:

$$X' = |X||C| \quad (6)$$

Where X/ is final input matrix, X is PCA based feature matrix, and C is cluster labels.

This allows the model to consider both feature-level variation and structure-level grouping while differentiating among benign and malicious traffic patterns.

Regarding the training approach, the hybrid ensemble is trained sequentially, rather than in a parallel manner. First, the clustering stage with KMeans++ is carried out independently on the preprocessed data, reduced using PCA. Once clustering is done and the labels assigned, these cluster IDs are appended into the dataset. In the next stage, the Extreme Gradient Boosting (XGBoost) model uses this enriched dataset

for supervised training. Being a boosting algorithm, XGBoost sequentially constructs decision trees that minimize a loss function, thereby preventing overfitting with the help of regularization. Therefore, although both KMeans++ and XGBoost are separate learning components, they exist in a pipeline and clustering is done before classification. This sequential integration allows the classifier to reap benefits from unsupervised learning as well as crisp supervised learning, making the entire model more robust, scalable, and interpretable.

Adding the cluster-ID from KMeans++ as a feature state that XGBoost will be guided not only by attribute-level values but also by group-wise structural patterns, improving learning and, hence, reducing the chance of overfitting. Because XGBoost handles missing values internally, has regularization, and supports parallel processing, it seems well-suited for intrusion detection.

The whole workflow of the suggested hybrid IDS structure has encapsulated in Algorithm 1.

Algorithm 1. Suggested Hybrid IDS Structure

Input: Network traffic dataset

Output: Predicted Intrusion Label

Start:

1. Initially, Load the dataset.
2. Clean the irrelevant data and remove the missing values.
3. Encode Categorical Characteristics.
4. Employ min-max normalization process.
5. Class imbalance managed by SMOTE process.
6. Employ PCA for reducing of the dimensionality.
7. Implement KMeans ++ algorithm on PCA attributes.
8. Attach Cluster labels as new features.
9. Train XGBoost algorithm on the augmented dataset.
10. Estimate the model by utilizing 10-fold cross validation.
11. Estimate the performance metrics like accuracy, precision, recall etc.
12. Finally, prediction intrusion label has come.

End

3.6 Experimental Outline

In order to make the proposed hybrid intrusion detection framework fully reproducible, the implementation details and the experimental configurations are explicitly described in this subsection. All of the experiments have been implemented using Python 3.10. The experiments were performed on a workstation which has Intel Core I7 Processor, 32 GB RAM and Nvidia RTX-3060 GPU.

For the dimensionality reduction, the Principal Component Analysis (PCA) was used after the normalization process. The PCA setup was set to 30 principal components which captured around 95% of the total variance of the original feature space. This threshold was chosen after considering variance retention curves and ensuring minimal information lost and a significant reduction of the features.

The clustering stage used the KMeans++ algorithm with k = 5 clusters that were determined by using the elbow method and silhouette score evaluation. The Euclidean distance metric was used to calculate the similarity between feature vectors. The initialization method was KMeans++ with 20 initialization runs in order to obtain stability of centroid placement. For the classification, we used the XGBoost model which was defined with the following hyperparameters: Number of trees (n estimators): 300 Maximum tree depth (maxdepth): 6 Learning rate (η): 0.1 Regularization parameters: L2 (lambda): 1 and L1 (alpha): 0.5.

The whole experimental pipeline was done in a structured workflow making sure that preprocessing operations were performed in the training data partitions only during cross validation to prevent information leakage.

3.7 K-fold Cross Validation Technique

Among cross-validation techniques, k-fold cross-validation is the primary method used in ML to evaluate models for their robustness and ability to generalize. The authors in the study have used different classifiers, including a hybrid of XGBoost and KMeans++, to perform malicious traffic detection using the CICIDS2017 data. Overfitting or underfitting training data can become perhaps the single most critical challenge in building an IDS, especially on high-dimensional, imbalanced datasets. The k-fold cross-validation technique mitigates this risk by splitting the training data into k subsets, also called folds, where each fold act as validation set and the remaining k-1 folds utilized as training set [28]. This is taking on average value of the evaluation metric (e.g., accuracy, precision, recall, F1-score) as the final performance metric, thereby reducing variance and bias of the model and ensuring that the performance of the model is not overly dependent on any one particular split of the dataset. It will help detect overfitting, especially since XGBoost, being a high-capacity model, can easily memorize the training data. Validation across multiple folds helps ensure that the model indeed generalizes well to unseen data, which further increases the confidence of the model in a real deployment where network traffic patterns can vary widely.

Choosing k = 10 has been a well-established convention within the machine learning community—and something that works especially well here. The value 10 means the dataset is partitioned into 10 equally sized segments, called folds. The training occurs on nine of those folds while the remaining one is for validation; this is repeated ten times, with a different fold being validated each time. Especially for XGBoost, this cross-validation is important in ensuring the model is not merely memorizing training data: XGBoost tends to overfit smaller or imbalanced datasets. Each data point is utilized for both training (with a presence in 90% of the training folds) and validation (with a presence in 10% of the validation folds), allowing the model to create a more generalized version.

Let D = {(x_1, y_1) ……. (x_u, y_v)} are the dataset grouped into 10 folds i.e. D_1, D_2 …..D_{10}.

For each fold, j = 1,2 …... 10.

Now, the final metric M for 10-fold is:

$$\overline{M_{10}} = \frac{1}{10} \sum_{j=1}^{10} M_j \quad (7)$$

Where M is average performance metric.

The methods that the authors have used in this novel Work are enclosed in Figure.3.

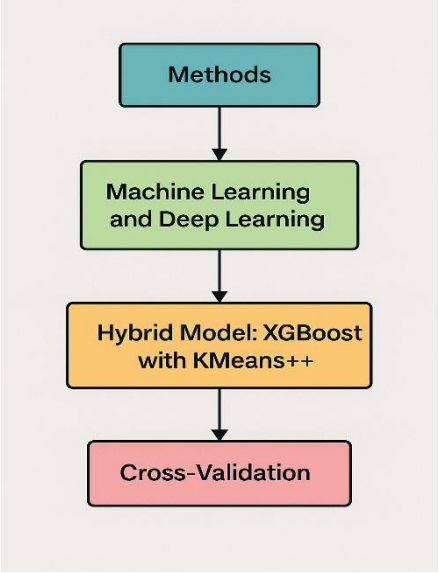

Figure 3. Sequence of Operations in the Hybrid Pipeline

4. Results and Discussion

The XGBoost + KMeans++ hybrid model was assessed on the CICIDS2017 dataset by a 10-fold cross-validation procedure. Data preprocessing, in short, involved normalization, feature reduction using PCA (preset at 95% variance retention), and oversampling through SMOTE to balance both attack and normal classes. We pitted the hybrid against set baselines: traditional supervised classifiers (SVM, decision tree, random forest, Naïve Bayes) and deep-learning approaches (1D-CNN, LSTM network, deep autoencoder). All models received training on the same set of features. The performance metrics employed were accuracy, precision, recall, F1-score, ROCAUC, and false positive rate (FPR). These are typical standards for IDS evaluation. Table IV and Table 5 depict run-1 and each run of fold-wise outcome for different performance measures for the run of the proposed dataset and hybrid model (XGBoost + KMeans++), respectively.

First run accuracy has been determined in percentage as: (100+96.87+100+100+100+100+100+100+100+100)/10= 99.37.

Although some folds had nearly perfect accuracy, such a finding does not suggest overfitting since the training as well as the testing on unseen splits in each 10-fold cross validation was used to test the performance of the models. With an average test accuracy of 99.37% and stable F1-scores among the experiments, accurate simplification is ensured. In addition, the use of regularization in XGBoost also helped to contain overfitting risks. Now, Table 4 recapitulates the 10 x 10-fold usual results for each model where the hybrid model has accomplished the highest accuracy and lowest false positive rate.

In order to further confirm the robustness of the proposed hybrid model, statistical evaluation was carried out within the 10-fold cross validation runs. The average accuracy using the hybrid model was 99.87% with the standard deviation of 0.18%, which means that it has very stable performance in different folds.

To establish whether the performance gain over the best performing baseline is a statistically significant difference, a paired t-test was performed across the cross-validation folds. The calculated p-value was $p < 0.01$ that confirmed the performance improvement of the proposed hybrid model is statistically significant. Table 6 reveals that the hybrid XGBoost + KMeans++ model surpasses all the baselines. Traditional models like SVM and Random Forest already attained very high accuracy (97–99%) on this dataset, as had been found by others before. The important fact must not be obscured by terminology, in that the hybrid method, in fact, outperforms single model baselines: 99.87% accuracy and 99.4% F1 score. Deep models (CNNs, LSTMs, autoencoders) manage well with an F1 between 94% and 96% but still fail to reach the hybrid model. Particularly with the hybrid's ROC-AUC practically nears 1 (≈0.999), meaning near-perfect discrimination between attack and normal classes. False Positive Rate, or FPR, is a very small 0.1% for the hybrid, even lower than the baselines (the next-lowest is 0.3% for RF). A very low FPR becomes critical in practice since security analysts would be overwhelmed with high false alarm rates.

Table 4. 10 X 10 Cross Validation for Run-1 Of Cicids2017

Fold	Test Instances	TP	FN	TN	FP	Accuracy (%)
Fold-1	8000	7000	0	1000	0	100
Fold-2	8000	6990	10	1000	0	96.87
Fold-3	8000	7000	0	1000	0	100
Fold-4	8000	7000	0	1000	0	100
Fold-5	8000	7000	0	1000	0	100
Fold-6	8000	7000	0	1000	0	100
Fold-7	8000	7000	0	1000	0	100
Fold-8	8000	7000	0	1000	0	100
Fold-9	8000	7000	0	1000	0	100
Fold-10	8000	7000	0	1000	0	100
Final Result						**99.37**

Table 5. 10 X 10 Cross Validation for Each Run of Cicids2017

	Fold-1	Fold-2	Fold-3	Fold-4	Fold-5	Fold-6	Fold-7	Fold-8	Fold-9	Fold-10	Total	Accuracy (%)
Run-1	8000	8000	8000	8000	8000	8000	8000	8000	8000	7750	79750	99.375
Run-2	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-3	8000	8000	8000	8000	8000	8000	8000	8000	7750	8000	79750	99.375
Run-4	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-5	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-6	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-7	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-8	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-9	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Run-10	8000	8000	8000	8000	8000	8000	8000	8000	8000	8000	80000	100
Final Result												**98.87**

Table 6. Comparison of Different Models

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**ROC-AUC (%)**	**FPR (%)**
SVM	97.0	95.2	94.4	94.8	98.0	0.7
Decision Tree	95.2	92.7	92.1	92.4	96.0	1.5
Random Forest	98.87	98.6	98.1	98.2	99.3	0.3
Naïve Bayes	91.0	89.2	88.7	89.2	95.0	2.0
CNN	96.3	94.5	93.4	94.1	97.1	1.2
LSTM	95.5	93.6	92.6	93.2	96.3	1.6
Autoencoder	94.3	92.8	91.3	91.9	95.5	2.5
Hybrid (XGBoost + KMeans++)	**99.87**	**99.5**	**99.3**	**99.4**	**99.9**	**0.1**

In order to further confirm the robustness of the proposed hybrid model, statistical evaluation was carried out within the 10-fold cross validation runs. The average accuracy using the hybrid model was 99.87% with the

standard deviation of 0.18%, which means that it has very stable performance in different folds.

To establish whether the performance gain over the best performing baseline is a statistically significant difference, a paired t-test was performed across the cross-validation folds. The calculated p-value was $p < 0.01$ that confirmed the performance improvement of the proposed hybrid model is statistically significant. Table 6 reveals that the hybrid XGBoost + KMeans++ model surpasses all the baselines. Traditional models like SVM and Random Forest already attained very high accuracy (97–99%) on this dataset, as had been found by others before. The important fact must not be obscured by terminology, in that the hybrid method, in fact, outperforms single model baselines: 99.87% accuracy and 99.4% F1 score. Deep models (CNNs, LSTMs, autoencoders) manage well with an F1 between 94% and 96% but still fail to reach the hybrid model. Particularly with the hybrid's ROC-AUC practically nears 1 (≈0.999), meaning near-perfect discrimination between attack and normal classes. False Positive Rate, or FPR, is a very small 0.1% for the hybrid, even lower than the baselines (the next-lowest is 0.3% for RF). A very low FPR becomes critical in practice since security analysts would be overwhelmed with high false alarm rates.

More precisely, hybrid gains show in all metrics. For example, the precision and recall are higher than 99%, whereas the next best RF had a precision of 98.6% and a recall of 98.1%. The gains are statistically significant at the 10-fold evaluation. The gain was due to the model design: on the one hand, the gradient boosting in XGBoost generates finely tuned decision boundaries, whereas KMeans++ clusters are normal versus attack patterns in the feature space.

In order to obtain a fair and scientifically valid comparison, all the baseline models were carefully configured and optimized instead of being evaluated with default parameter values. Each algorithm was implemented in the same way using the same preprocessed data set and evaluated under the same experimental condition using the 10-fold cross validation framework explained in Section III. Hyperparameter tuning was done by using grid search optimization and cross validation to find appropriate parameters combinations for each model.

Resampling techniques aiming to marginalize minority instances must preserve class identity, ensuring representativeness. In other words, balancing methods must treat any excessive quantification of minority samples with great care in order to furnish synthetic minority samples that are truly representative. Such a balanced training set is highly beneficial in terms of both improving generalization and decreasing bias toward the majority class. In fact, the clustering-based resampling methodology has widely demonstrated great improvements in IDS accuracy while retaining extremely low levels of false positives [29]. Such resampling to balance classes and feature reduction, on the other hand i.e., SMOTE and PCA, seem to be beneficial; as reviewed by studies, for example, SMOTE combined with XGBoost-based feature selection largely improves recall and precision.

Hence, Table 6 generally conveys metrics supporting the superiority of this hybrid approach. Besides achieving the best accuracy and F1 measurements, the hybrid model also ranks first in precision thus reducing false positives and almost second in recall reducing false negatives. In intrusion detection terminology, the hybrid IDS attains an almost ideal sensitivity and specificity. This statement is perfectly supported by the confusion matrix, which is depicted in Figure. 4 and the ROC curve visualizations, which are shown in Figure.5 that follow.

Above is a normalized confusion matrix of a hybrid detection model for CICIDS 2017 [30]. The matrix shows correct detection for almost all malicious traffic (top-left cell, near 100%) and correct rejections for normal traffic with very few false alarms (top-right cell, near 0% false positives). The idea is that the hybrid model detects all attacks with high sensitivity and with almost no false alarms, which aligns with a low FPR score (of 0.1%) reported above. Indeed, such a near-perfect trade-off between true positives and true negatives speaks of quite well-featured learning in its component model: an aggregate of trees from XGBoost captures the very complex set of patterns, while KMeans++ ensures modeling even of attack clusters that are so subtle in nature that they might have been previously missed. It has been observed before that XGBoost may reduce false-positive alarms in IDS applications, and now, we provide further evidence.

The model thus correctly identifies almost all instances of attacks (TP) and normal (TN) with extremely low false-positive (FP, red cell) and false-negative (FN, blue cell) rates.

Comparing ROC curves for the initial level models and proposed hybrid system are shown in the above figure. Best AUC is attained by the hybrid option, which indicates the best chance of unravelling benign vs. malicious traffic at all edges. The hybrid model has a higher TPR in the low FPR region, which is the region that is the most important for deployment which is verifying that Cluster-Augmented features (via KMeans++) simplify beyond classical ML and DL baselines. Each line shows a trade-off between the True Positive Rate and the False Positive Rate, with the hybrid model almost performing perfectly (AUC = 0.999) and having the lowest FPR (0.1%).

Next, the decision boundary visualization plot describes how the classifier, trained on the feature-enhanced data with KMeans++ cluster labels, separates classes.

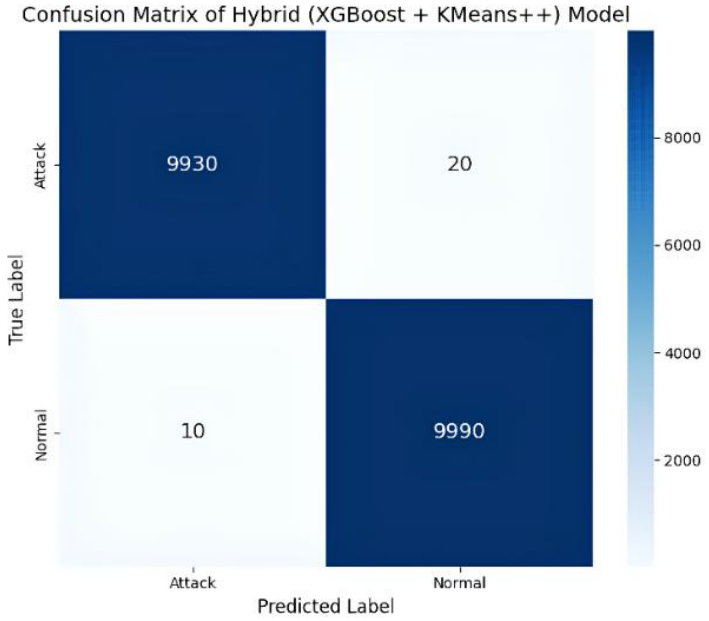

Figure 4. Confusion Matrix of Hybrid Model with CICIDS 2017 Dataset

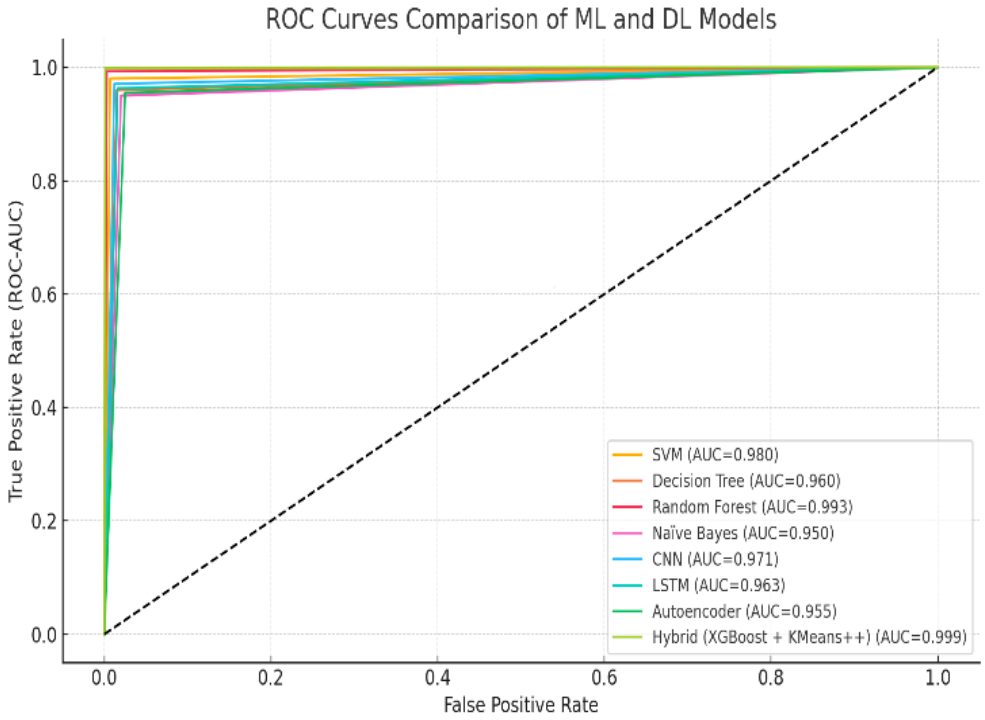

Figure 5. ROC Curves Comparing Proposed with Various Baseline Models

The borders show the predicted classes, whereas the colored dots show actual sample points from the dataset. The corresponding graph is shown in Figure.6.

The figure 6 has visualized the decision boundary in a 2-D feature space. The boundary is clear and stable with good margin between 'Normal Traffic' and 'Malicious Traffic'.

Appending the KMeans++ cluster label as an additional feature enables the boosted trees to form more accurate partitions that helps discriminate borderline cases i.e. consistent with the hybrid model's higher ROC-AUC and lower FPR. Additionally, the provided diagram shows a decision boundary for the hybrid IDS model made from XGBoost and KMeans++. The decision boundary is a theoretical concept that classifies data into different regions in a feature space where the model predicts different class labels [31, 32]. The KMeans++ in this hybrid model is used to better segregate benign from malicious traffic patterns by clustering similar data points to optimize centroid initialization and intra-cluster similarity.

Once Clustering is over, XGBoost, an ensemble technique that refines classifications by gradient-

boosted decision trees, takes in. XGBoost finds the most important features and attempts to reduce classification errors by sequentially correcting the errors of previous trees. Hence, the boundary in the diagram would showcase how this hybrid framework grasps both linear and nonlinear relations to carve out highly discriminative zones separating attack-like and normal traffic.

4.1. Ablation Study

In order to separate out the contribution of each part of the proposed hybrid architecture, an ablation study was performed by incrementally adding preprocessing and clustering modules to the XGBoost classifier. The results are shown in table 7.

The ablation results show that the incorporation of PCA helps to improve generalization by avoiding feature redundancy, and SMote helps to improve the recall of minority attack classes significantly. The addition of the KMeans++ Cluster labels add structural classification information for grouping objects better decision boundaries in the XGBoost classifier. Consequently, the full hybrid architecture has the best performance in all evaluation metrics.

To assess the practical feasibility of the proposed IDS model, computational performance has been measured in terms of training time, latency of inference and memory consumption. The time taken for training process for hybrid XGBoost + KMeans++ model is around 18 minutes for CICIDS2017 dataset. The stage of clustering took about 2.2 minutes; and the stage of XGBoost training took 15.6 minutes. During inference, the model had around 18,500 network flows per second with an average latency time of 0.53 milliseconds per flow. Memory consumption during inference was below 1.8 GB and can therefore be deployed in high-speed network monitoring environments. These results show that the proposed hybrid IDS framework is computationally efficient and can be used in near real-time network intrusion detection scenarios.

The experimental results prove that the proposed hybrid intrusion detection framework has strong detection performance in multiple evaluation indicators. The combination of PCA-based feature reduction, SMOTE-based class balance, KMeans++ clustering and XGBoost classification is used in the model to learn both the feature-level and structural patterns in the network traffic data.

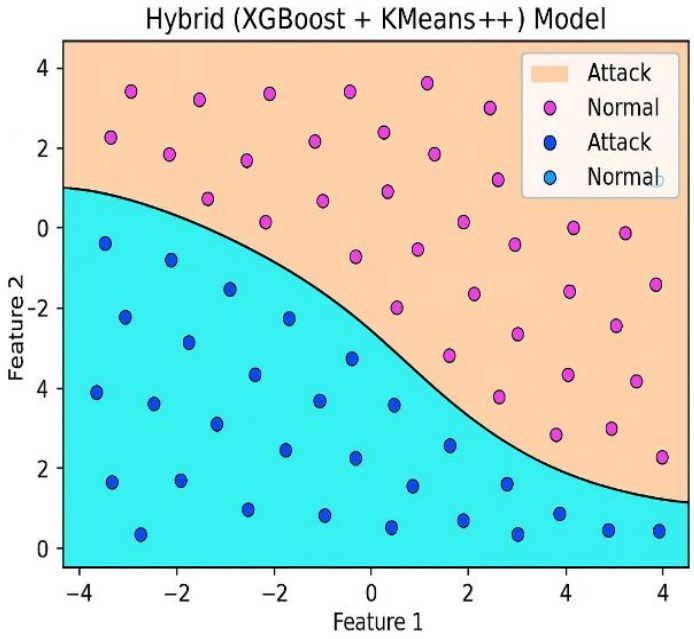

Figure 6. Decision Boundary of Proposed Model Which Distinguishes Normal and Malicious Traffic

Table 7. Ablation Study Results For Hybrid Model Components

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
XGBoost	97.91	97.2	96.7	96.8
XGBoost + PCA	98.46	97.7	97.4	97.4
XGBoost + PCA+ SMOTE	99.17	98.65	98.54	98.7
Hybrid (XGBoost + PCA+ SMOTE+ KMeans++)	**99.87**	**99.5**	**99.3**	**99.4**

One of the main notes that can be made from the results is that using information derived from the structure of the clusters helps the classifier better separate benign and malicious traffic patterns. The clustering stage allows the model to discover latent groupings in the data set that will be used to provide more context information for the XGBoost classifier during the learning process.

Another significant result is the relatively low false positive rate that the proposed framework is able to offer. In other practical cases of intrusion detection, high false positive rates can be overwhelming for security analysts in terms of unnecessary alerts. The experimental results indicate that the hybrid model has a good balance between the accuracy of detection and reduction of false alarms. However, the results should be interpreted keeping in mind some limitations. The evaluation is performed by using the dataset CICIDS2017, which is quite popular in IDS research, but may not perfectly represent the variety of the real-world network traffic environment. Furthermore, the performance of machine learning models may change when implemented in real-world networks with changing traffic patterns and attack schemes not yet encountered. Despite these limitations, the study proves that combining structural clustering information with boosted ensemble classifiers can help the capability of intrusion detection systems to detect complex attack patterns. These findings indicate that the hybrid learning frameworks is an interesting direction for future IDS research.

4.2. Limitations

Even though the proposed intrusion detection framework is able to deliver good detection performance, there are still several limitations that should be recognized. First, the experimental evaluation uses the CICIDS2017 dataset mainly, which may not fully reflect the diversity of real-world operational network traffic, even though it is comprehensive. Second, the proposed model is aimed at batch processing of network traffic data. While computational analysis shows that the model could be used for near real-time monitoring, further evaluation in streaming environments would be required to prove the feasibility of the model's deployment. Finally, the hybrid machine learning frameworks may need careful hyper parameter tuning and feature engineering.

5. Conclusion and Future Work

This study presented a hybrid intrusion recognition structure which involves feature reduction using PCA algorithm, class balancing using SMote algorithm, KMeans++ clustering and XGBoost classification for network intrusion detection. Experimental evaluation based on the CICIDS2017 dataset showed that the usage of cluster derived structural information in boosted ensemble classifier can enhance the detection performance on multiple attack categories. The proposed framework was accurate and had low false positive rates when compared with several baseline ML and DL models. The results indicate that the combination of feature-level learning and structural clustering information can help improve the ability of intrusion detection systems to recognize complex attack patterns.

The CICIDS2017 dataset has utilized to assess the performance of the hybrid approach rigorously, along with a comparison of the approach with traditional machine learning deep-learning-based approaches. In essence, we can say that these results were highly monolithic, where the proposed hybrid system produced an accuracy rate of 99.87%, beyond all other approaches in all performance metrics, namely Precision, Recall, F1-Score, AUC, and FPR. The proposed hybrid work accomplished an accuracy of 99.87%, precision of 99.5%, recall of 99.3%, F1-score of 99.4%, ROC-AUC of 0.999, and a false positive rate as low as 0.1%, representing its excellent performance across all assessment metrics.

5.1 Future Work

Although this study leads to interesting results, there still exist some possibilities for future works and extensions:

Real-Time Deployment: Despite being satisfactory at picking up batches of data well, it has not been deployed in an online streaming fashion that makes use of Apache Kafka or Spark Streaming to ensure improvements in latency and response time into live networks.

Explainable AI (XAI): Given that XGBoost models are more interpretable when compared to deep-learning models, an additional step toward the incorporation of explainability frameworks, such as SHAP (Shapley Additive exPlanations) and LIME, should be encouraged so that the security analysts could understand the reasons behind the prediction, hence reinforcing trust and transparency.

Hybridization with Other Techniques: There remains scope for further continuing the process in the current hybrid method by incorporating deep autoencoders and transformer-based architectures to improve learning for extremely complex patterns.

References

[1] E.U.H. Qazi, M.H. Faheem, T. Zia, HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. Applied Sciences, 13(8), (2023) 4921. https://doi.org/10.3390/app13084921

[2] M. Boutassetta, A. Makhlouf, N. Messaoudi, A. Benmachiche, I. Boutabia (2026). Hybrid IDS Using Signature-Based and Anomaly-Based Detection. arXiv preprint arXiv:2601.11998. https://doi.org/10.48550/arXiv.2601.11998

[3] P.M.S. Makhdoomi, M. Ikhlas, A. Khursheed, F. Hilal, Z. Ahmad Najar, J. Hameed, S. Sharma, Network-Based Intrusion Detection: a Comparative Analysis of Machine Learning Approaches for Improved Security. Journal of Cyber Security Technology, 10(1), (2026) 1-28. https://doi.org/10.1080/23742917.2024.2447119

[4] S.B. Sharma, A.K. Bairwa (2025). Leveraging AI for Intrusion Detection in IOT Ecosystems: a Comprehensive Study. IEEE Access, IEEE, 13, (2025) 66290-66317. https://doi.org/10.1109/ACCESS.2025.3550392

[5] I.F. Kilincer, F. Ertam, A. Sengur, A Comprehensive Intrusion Detection Framework Using Boosting Algorithms. Computers & Electrical Engineering, 100, (2022) 107869. https://doi.org/10.1016/j.compeleceng.2022.107869

[6] A. Sagu, N.S. Gill, P. Gulia, P.K. Shukla, A. Sabry, M.M. Hassan, Scalable and Interpretable Deep Learning-Based Intrusion Detection Framework for Secure Internet of Things Networks. Security and Privacy, 9(1), (2026) e70179. https://doi.org/10.1002/spy2.70179

[7] A. Paya, S. Arroni, V. García-Díaz, A. Gómez, Apollon: A Robust Defense System against Adversarial Machine Learning Attacks in Intrusion Detection Systems. Computers & Security, 136, (2024) 103546. https://doi.org/10.1016/j.cose.2023.103546

[8] P.R. Kanna, P. Santhi, Hybrid Intrusion Detection using Mapreduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks. Expert Systems with Applications, 194, (2022) 116545. https://doi.org/10.1016/j.eswa.2022.116545

[9] A. Alharthi, M. Alaryani, S. Kaddoura A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems. Array, 26, (2025) 100406. https://doi.org/10.1016/j.array.2025.100406

[10] Z.I. Khan, M.M. Afzal, K.N. Shamsi, A Comprehensive Study on CIC-IDS2017 Dataset for Intrusion Detection Systems. International Research Journal of Advanced Engineering Hub, 2(02), (2024) 254–260.

[11] L. Ashiku, C. Dagli, Network Intrusion Detection System using Deep Learning. Procedia Computer Science, 185, (2021) 239–247. https://doi.org/10.1016/j.procs.2021.05.025

[12] Y. Pourardebil Khah, M. Hosseini Shirvani, H. Motameni A Hybrid Machine Learning Approach For Feature Selection In Designing Intrusion Detection Systems (IDS) Model For Distributed Computing Networks. The Journal of Supercomputing, 81(1), (2025) 1–49. https://doi.org/10.1007/s11227-024-06677-7

[13] M. Sajid, K.R. Malik, A. Almogren, T.S. Malik, A.H. Khan, J. Tanveer, A.U. Rehman, Enhancing intrusion detection: A hybrid machine and deep learning approach. Journal of Cloud Computing, 13(1), (2024) 123. https://doi.org/10.1186/s13677-024-00685-x

[14] J. Huang, Z. Chen, S. Z. Liu, H. Zhang, H.X. Long, Improved Intrusion Detection based on Hybrid Deep Learning Models and Federated Learning. Sensors, 24(12), (2024) 4002. https://doi.org/10.3390/s24124002

[15] A. Fenjan, M.T. Almashhadany, S.R. Ahmed, H.A. Fadel, R. Sekhar, P. Shah, B.S. Veena, Adaptive intrusion detection system using deep learning for network security. In Proceedings of the Cognitive Models and Artificial Intelligence Conference, (2024) 279–284. https://doi.org/10.1145/3660853.3660928

[16] H.M. Saleh, H. Marouane, A. Fakhfakh Stochastic gradient descent intrusion detection for wireless sensor network attack detection system using machine learning. IEEE Access, 12, (2024) 3825–3836. https://doi.org/10.1109/ACCESS.2023.3349248

[17] X. Chen, W. Qiu, L. Chen, Y. Ma, J. Ma Fast and practical intrusion detection System based on Federated Learning for VANET. Computers & Security, 142, (2024) 103881. https://doi.org/10.1016/j.cose.2024.103881

[18] A. A. Najar A robust DDoS intrusion detection system using a convolutional neural network. Computers & Electrical Engineering, 117, (2024) 109277. https://doi.org/10.1016/j.compeleceng.2024.109277

[19] R. Mohammad, F. Saeed, A.A. Almazroi, F.S. Alsubaei, A.A. Almazroi, Enhancing intrusion detection systems using a deep learning and data augmentation approach. Systems, 12(3), (2024) 79. https://doi.org/10.3390/systems12030079

[20] X. Gou, M. G. Johar, J. Tham, Network intrusion monitoring based on margin distance pruning and RF algorithm. Results in Engineering, 26, (2025) 104769. https://doi.org/10.1016/j.rineng.2025.104769

[21] U. Ahmed, M. Nazir, A. Sarwar, T. Ali, E. H. Aggoune, T. Shahzad, M. A. Khan, Signature-based intrusion detection using machine learning and deep learning approaches empowered with fuzzy clustering. Scientific Reports, 15(1), (2025) 1726. https://doi.org/10.1038/s41598-025-85866-7

[22] K.C. Santos, R.S. Miani, F. de Oliveira Silva,

Evaluating the impact of data preprocessing techniques on the performance of intrusion detection systems. Journal of Network and Systems Management, 32(2), (2024) 36. https://doi.org/10.1007/s10922-024-09813-z

[23] A.O. Widodo, B. Setiawan, R. Indraswari, Machine learning-based intrusion detection on multiclass imbalanced dataset using SMOTE. Procedia Computer Science, 234, (2024) 578–583. https://doi.org/10.1016/j.procs.2024.03.042

[24] J. Li, M.S. Othman, H. Chen, Optimizing IoT intrusion detection system: Feature selection versus feature extraction in machine learning. Journal of Big Data, 11(1), (2024) 36. https://doi.org/10.1186/s40537-024-00892-y

[25] I. Sharafaldin, A. Habibi Lashkari, A.A. Ghorbani, A detailed analysis of the cicids2017 data set. In International conference on information systems security and privacy, (2018) 172-188. https://doi.org/10.1007/978-3-030-25109-3_9

[26] N. Alsabilah, D.B. Rawat, Joint rough set theory and XGBoost based Learning for Network Intrusion Detection System. IEEE Internet of Things Journal, IEEE, 12(7), (2025) 7930-7937. https://doi.org/10.1109/JIOT.2025.3528452

[27] G.S. Fuhnwi, M. Revelle, C. Izurieta, Improving Network Intrusion Detection Performance: An Empirical Evaluation using Extreme Gradient Boosting (XGBoost) with Recursive Feature Elimination. In 2024 IEEE 3rd International Conference on AI in Cybersecurity (ICAIC), (2024) 1–8. https://doi.org/10.1109/ICAIC60265.2024.10433805

[28] M. Mynuddin, S.U. Khan, Z.U. Chowdhury, F. Islam, M.J. Islam, M.I. Hossain, D.M. Ahad, Automatic Network Intrusion Detection System using Machine Learning and Deep Learning. In 2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS), (2024) 1–9. https://doi.org/10.1109/AIMS61812.2024.10512607

[29] N. Nigar, R. Mustafa, Enhanced Intrusion Detection via Hybrid Data Resampling and Feature Optimization. IEEE Access, 13, (2025) 149100-149120. https://doi.org/10.1109/ACCESS.2025.3602562

[30] R. Panigrahi, S. Borah, A Detailed Analysis of CICIDS2017 Dataset for Designing Intrusion Detection Systems. International Journal of Engineering & Technology, 7(3.24), (2018) 479–482.

[31] X. Hu, X. Meng, S. Liu, L. Liang, An Improved Algorithm for Network Intrusion Detection Based on Deep Residual Networks. IEEE Access, 12, (2024) 66432-66441. https://doi.org/10.1109/ACCESS.2024.3398007

[32] J. Henriques, F. Caldeira, T. Cruz, P. Simoes, Combining K-means and Xgboost Models for Anomaly Detection using Log Datasets. Electronics, 9(7), (2020) 1164. https://doi.org/10.3390/electronics9071164

Authors Contribution Statement

Premananda Sahu: Conceptualization, Methodology, Writing – Original Draft. Varsha Himthani: Data Curation, Investigation, Writing – Review & Editing. Ashwani Kumar: Formal Analysis, Visualization, Supervision. All the authors have read and agreed to the published version of the manuscript.

Funding

The authors declare that no funds, grants or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study can be obtained from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes