A Machine Learning Framework with Optimizations Towards an Efficient Intrusion Detection System

Pasham John Babu[1], Amjan Shaik[2]

[1]Research Scholar,Department of CSE, Bharatiya Engineering Science and Technology Innovation University (BESTIU)
[2]Professor & HOD Department of CSE, St.Peter's Engineering College, Hyderabad, Telangana, India

Received: 13.04.2024 Revised : 13.05.2024 Accepted: 19.05.2024

ABSTRACT

With the escalating number of attacks on information systems and networks, the need for robust cybersecurity measures has never been more pressing. Leveraging the power of Artificial Intelligence (AI) and Machine Learning (ML), we can develop intrusion detection systems that are more adaptive and efficient. While existing machine learning models can be used for intrusion detection, their performance can be significantly improved through hyperparameter tuning and feature engineering. This paper introduces a machine learning-based framework that is specifically designed to create an efficient intrusion detection system. The framework incorporates a hybrid feature engineering methodology and hyperparameter tuning using Bayesian Optimization for machine learning models. We propose an ensemble of machine-learning models that have been fine-tuned for superior intrusion detection performance. Our proposed algorithm, known as Ensemble Learning-Based Intrusion Detection (ELBID), harnesses the power of machine learning models with optimizations for intelligent detection of intrusions. The ensemble model we have developed surpasses many existing ML models in intrusion detection, achieving an impressive accuracy of 94.59%. As a result, our optimized ensemble model can be seamlessly integrated into real-world applications, significantly enhancing cyber security.

Keywords: Intelligent Intrusion Detection, Artificial Intelligence, Machine Learning, Hyper Parameter Tuning, Feature Engineering, Cyber Security

1. INTRODUCTION

An Intrusion Detection System (IDS) is a security solution designed to monitor network or system activities for malicious activities or policy violations. It works by analyzing network traffic, system logs, or behavior patterns to identify potential threats. IDS can be classified into two main types: Network-based Intrusion Detection Systems (NIDS) that monitor network traffic in real-time, and Host-based Intrusion Detection Systems (HIDS) that analyze activities on individual devices. IDS helps organizations detect and respond to cyber threats, enhancing overall security posture. Machine learning for intrusion detection involves using machine learning algorithms to detect and prevent unauthorized access to computer systems or networks. By analyzing patterns in network traffic data, machine learning models can learn to identify abnormal behavior that may indicate a potential security breach or cyber attack. These models can be trained on labeled datasets containing examples of normal and malicious network activity, enabling them to classify new data and flag any suspicious or anomalous behavior in real-time.

The use of machine learning for intrusion detection has become increasingly important due to the growing complexity and volume of cyber threats, allowing organizations to enhance their security measures and protect sensitive information from unauthorized access. There are many existing intrusion detection systems developed based on machine learning as found in the literature in Section 2. From the literature, it was observed that machine learning models need to be optimized towards leveraging intrusion detection performance. Our contributions in this paper are as follows:

1. We present a machine learning-based framework focused on creating an efficient intrusion detection system.The framework includes a hybrid feature engineering methodology and hyperparameter tuning using Bayesian Optimization for machine learning models.
2. We propose an ensemble of machine-learning models optimized for superior intrusion detection performance.
3. Our proposed algorithm, Ensemble Learning-Based Intrusion Detection (ELBID), utilizes machine learning models with optimizations to effectively and intelligently detect intrusions. The ensemble

model we developed outperforms many existing machine learning models in intrusion detection, achieving remarkable accuracy.

The remainder of the paper is structured as follows: Section 2 reviews the literature on existing machine learning models used for intrusion detection systems. Section 3 presents the proposed intrusion detection system, which is a signature-based approach with optimizations toward leveraging performance in intrusion detection. Section 4 presents our experimental results, and the results are compared with many existing models. Section 5 discusses the research we carried out in this paper besides providing the study's limitations. Section 6 concludes our research work, besides giving directions for the future scope of the research.

2. RELATED WORK

This review covers literature about existing methods used for intrusion detection. Mishra et al. [1] discuss the challenges machine learning techniques face in detecting various attacks, highlighting technique-specific relevance, challenges, and future directions.Sridevi et al. [2] focuses on how intrusion detection systems (IDS) with machine learning algorithms enhance security by identifying threats in various environments. The study compares the efficiency of algorithms across different applications.Ertam et al. [3] emphasize the increasing internet usage and the subsequent rise in security threats, which necessitate the development of IDS with machine learning-based systems. The study notes promising results using various datasets.Cavusoglu [4] proposes an IDS that combines machine learning and feature selection to efficiently detect attacks. Tests show high accuracy and low false positives.Lisboa et al. [5] highlights the escalating concerns about computer network security as technology advances, particularly focusing on the challenges of intrusion detection in IoT. The study emphasizes the importance of intelligent techniques that balance accuracy and efficiency.

Sultana et al. [6] discusses how software-defined networking (SDN) with machine learning/deep learning-based network intrusion detection systems (NIDS) effectively safeguards networks, despite facing challenges such as dynamic detection, feature selection, and scalability. Sai Kiran et al. [7] uses an IoT test bed to simulate attacks and capture data for machine learning classification, noting challenges such as realistic dataset generation and diverse data handling. Othman et al. [8] emphasizes the significance of Big Data in reshaping IDS and proposes a Spark-Chi-SVM model with feature selection for high-performance intrusion detection. Hagar et al. [9] examines machine learning and deep learning algorithms on a specific dataset, favoring random forests for intrusion detection. Gao et al. [10] addresses the challenges faced by traditional IDS in timely detection of advanced threats and proposes an ensemble model to enhance accuracy. Otoom et al. [11] observes high performance in IDS using supervised learning and proper feature selection and data balancing.

Moubayed et al. [12] introduces an optimized machine learning-based NIDS framework that enhances detection accuracy and reduces computational complexity. Maseer et al. [13] evaluates anomaly-based IDS using 10 popular machine learning algorithms and proposes a benchmarking approach to improve future research. Resender et al. [14] outlines the usage of Random Forest models in Intrusion Detection Systems, focusing on challenges and future prospects. Khan et al. [15] proposes a user-friendly biometric system for public transport in the UAE and emphasizes machine learning's success in Vehicular Ad-Hoc Networks (VANETs). Dilip et al. [16] proposes a machine learning-based IDS utilizing the NSL KDD dataset and evaluating ANN, SVM, and ID3 algorithms, with SVM exhibiting higher accuracy. Verma et al. [17] focuses on using machine learning classifiers to secure IoT against DoS attacks and plans to design an IDS to defend against routing attacks in IoT networks. Maxwell et al. [18] examines the vulnerability of NIDS to adversarial attacks and aims to investigate the internal mechanisms of machine learning models.

Zong et al. [19] introduces a 3D visualization method for NIDS data to aid understanding of machine learning detection outcomes and misclassifications. Santos et al. [20] discusses the drawbacks of current intrusion detection systems and introduces a reinforcement learning model to efficiently tackle evolving network behavior, with proposals for further enhancements. Wang et al. [21] addresses the challenges of detecting imbalanced network traffic and proposes a novel DSSTE algorithm for improved classification accuracy. Yang et al. [22] focuses on enhancing interpretability in intrusion detection models using a SHAP-based framework and suggests utilizing more data and real-time interpretation. Raheem et al. [23] evaluates intrusion detection methods like SVM, RF, and ELM for network security, concluding that ELM performs best with large datasets. Caminero et al. [24] proposes an intrusion detection algorithm that combines supervised and reinforcement learning to effectively address complex network security challenges.

In a recent study, Sethi et al. [25] presented a context-adaptive IDS that utilizes deep reinforcement learning agents for accurate and robust intrusion detection. The system's ability to withstand adversarial attacks has been confirmed, and there are plans to improve its deployment architecture for heterogeneous cloud applications in the future. Rahman et al. [26] proposed two IoT intrusion detection system architectures to address latency issues in resource-constrained networks. Their effectiveness was validated through experimentation on the AWID dataset. The study also emphasized the importance of comprehensive data validation and exploration of feature removal. Chen et al. [27] introduced AE-IDS, an intrusion detection system based on auto-encoders, which addresses data imbalance and low accuracy issues. The study recommends incorporating additional information such as system logs and security device alarms for better defense against complex intrusions.

Park et al. [28] proposed a machine learning-based system that improves detection and process achievement, providing valuable insights for future implementations. The study suggests further data collection for comprehensive analysis using advanced techniques. Lu et al. [29] highlighted the critical importance of safeguarding network security, with the proposed SVM and Naive Bayes framework demonstrating strong performance. Future work includes handling diverse attacks and network traffic. Adeel et al. [30] emphasized that ensuring cyber security is a critical challenge in the digital age. Their proposed IDS, which combines statistical analysis and auto-encoders, outperformed traditional and deep learning methods. Future work includes exploring real-time data analysis. Overall, it was observed from the literature that machine learning models need to be optimized to enhance intrusion detection performance.

3. PROPOSED FRAMEWORK

This section will outline the methodology used to develop an effective intrusion detection system based on machine learning models with optimizations. We will discuss the proposed framework, including the underlying mechanisms, algorithms, dataset details, and evaluation methodology.

3.1 Problem Definition

Cyberattacks are malicious actions carried out by individuals or groups targeting computer systems, networks, or digital devices. These attacks can involve activities such as stealing sensitive information, disrupting normal operations, or causing damage to data or systems. Cyberattacks can take various forms, including malware infections, phishing scams, denial of service attacks, ransomware, etc. It is essential for organizations and individuals to implement robust cybersecurity measures to protect against cyber attacks and safeguard their digital assets. The focus of this research is on developing a machine learning-based framework with optimizations for efficient detection of intrusions.

3.2 Proposed Framework

We have developed a machine learning-based framework with optimizations for efficient endogen detection. After reviewing the literature, we have found that machine learning models need to be optimized through hyperparameter tuning and feature engineering to improve detection performance. It is essential to develop a machine learning-based framework with optimizations to ensure that the underlying machine learning models achieve optimal performance, especially in the case of supervised learning models. To achieve this, we emphasize the importance of feature engineering and hyperparameter tuning in our proposed framework. Through feature engineering, the underlying machine learning models receive quality training, and with hyperparameter tuning, the models are optimized with suitable values for their hyperparameters. The proposed intrusion detection system is a signature-based system that relies on training samples.

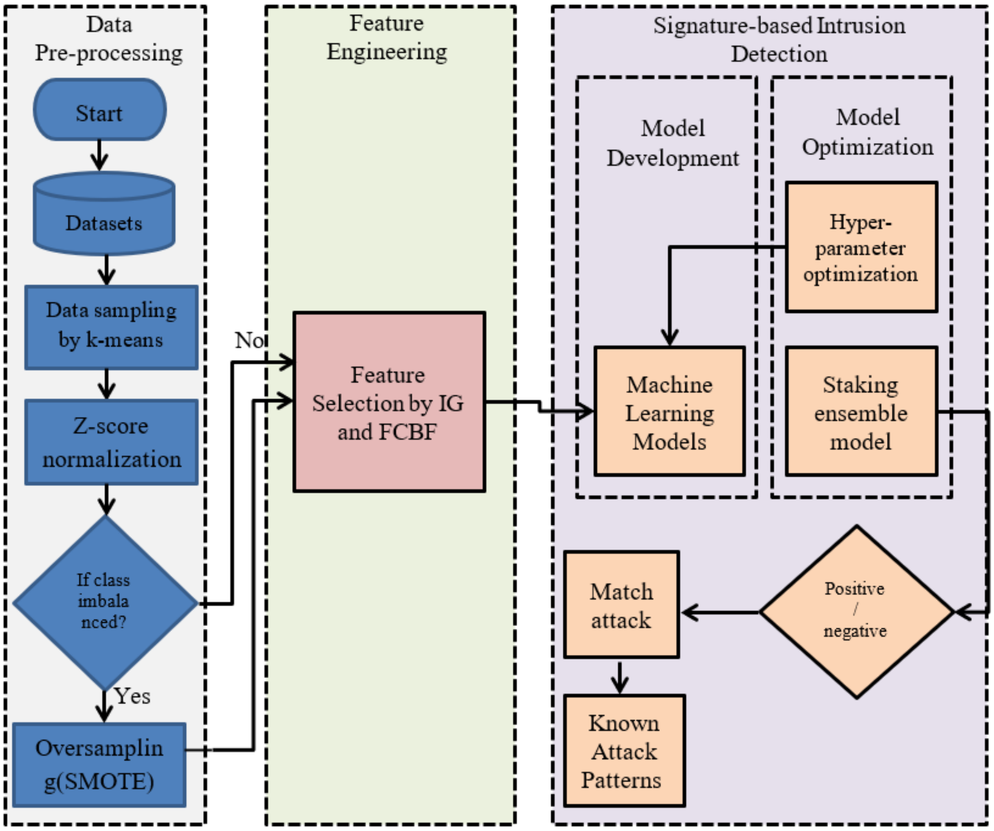


Figure 1: Outline of the proposed machine learning-based framework for realizing an efficient intrusion detection system

Figure 1 illustrates the proposed framework for an intrusion detection system (IDS) utilizing machine learning. The framework comprises three stages: data pre-processing, feature engineering, and signature-based intrusion detection. In the data pre-processing stage, raw data is gathered and prepared for further analysis. This involves data sampling using k-means clustering to reduce the data set's size and normalization using z-score normalization to ensure that all features are on the same scale. Feature engineering includes selecting the most relevant features from the data set. The framework utilizes information gain (IG) and Fast Correlation Based Filter (FCBF) for feature selection. The final stage is signature-based intrusion detection, where a machine-learning model is trained to identify malicious traffic patterns. The framework deploys a stacked ensemble model, which is a combination of different machine-learning models. The model is optimized using Bayesian optimization (BO). The framework also addresses class imbalance, a situation where there is a significant difference between the number of normal data points and the number of attack data points. If the data set is imbalanced, the framework uses SMOTE (Synthetic Minority Oversampling Technique) to generate synthetic data points for the minority class.

3.3 Feature Engineering

Feature engineering involves modifying, selecting, or creating new features from existing data to improve machine learning model performance. This includes techniques such as imputation, scaling, encoding categorical variables, creating interaction terms, and transforming variables to make them more suitable for the model. Careful feature engineering can improve a model's predictive power by capturing underlying data patterns.

Feature selection on the CICIDS2017 dataset focuses on choosing a relevant subset of features or variables to enhance model performance. This is crucial for reducing overfitting, decreasing training time, and improving model interpretability. Before feature selection, it's important to understand the dataset, conduct exploratory data analysis, handle missing values, encode categorical variables, and normalize or scale the data if necessary. Evaluating the selected features using cross-validation and performance metrics is also essential to ensure the effectiveness of the feature selection process.

A hybrid feature selection approach based on IG and FCBF methods is proposed in this paper. This method combines the strengths of IG and FCBF to enhance feature selection effectiveness. Initially, FCBF is used to filter out features based on their correlation with the target and mutual correlation with other features, reducing the feature space and removing redundant or less informative features. Information Gain is then applied to the remaining subset of features to comprehensively assess the predictive power of each feature. Features that survive both FCBF and Information Gain filtering steps are selected as the final subset for training machine learning models. This integrated approach leverages the strengths of both methods, with FCBF handling correlations and redundancy and Information Gain providing information content assessment. The hybrid approach effectively balances the trade-off between feature relevance and redundancy, leading to improved model performance, reduced overfitting, and faster training times.

3.4 Hyperparameter Optimization

Hyperparameter optimization is a critical step in training machine learning models. It involves adjusting the hyperparameters of a model to enhance its performance. Hyperparameters are configuration settings that are defined before the training process begins and are not changed during training. They include parameters such as learning rate, batch size, number of hidden layers in a neural network, and regularization strength. Hyperparameter optimization can be done using techniques like grid search, random search, Bayesian optimization, as well as more advanced methods such as genetic algorithms and reinforcement learning. The aim of hyperparameter optimization is to identify the best set of hyperparameters that result in improved model performance on a validation dataset.

This paper utilized a hyperparameter optimization technique called Bayesian optimization with a tree-based Parzen estimator (BO-TPE). BO-TPE combines Bayesian optimization, which is a sequential model-based optimization technique, with the Tree-structured Parzen Estimator (TPE) algorithm. In BO-TPE, the TPE algorithm represents the objective function as a probability distribution and uses this model to guide the search for optimal hyperparameters. This method strikes a balance between exploring new hyperparameters and exploiting the best ones found so far to effectively search the hyperparameter space and identify the optimal configuration. Overall, BO-TPE is a powerful approach for hyperparameter optimization in machine learning. It is capable of efficiently navigating complex search spaces and identifying high-performing hyperparameter configurations with relatively few objective function evaluations.

3.5 Proposed Algorithm

We have developed a new algorithm called Ensemble Learning-Based Intrusion Detection (ELBID) to enhance the accuracy and reliability of network intrusion detection. This algorithm leverages the strengths of multiple machine learning (ML) models to address the challenge of detecting intrusions in network traffic. By using an ensemble learning approach, ELBID combines the predictions of several individual ML models to make more precise and dependable decisions. To achieve this, the algorithm goes through a structured process of data preprocessing, feature engineering, model training, and performance evaluation, using the CICIDS2017 dataset as the basis for testing and validation.

```
Algorithm: Ensemble Learning-Based Intrusion Detection (ELBID)
Input:CICIDS2017 dataset D, ML models M (GB, XGB, RF, ET)
Output: Intrusion detection results R, performance statistics P
    Begin
    D'←DataPreprocess(D)
    F←HybridFeatureEngineering(D')
    (T1, T2, T3)←DataSplit(D', F) //trian, test, val
    For each model m in M
    m.params←HyperparameterTuning(parameterSpace, D', F)
    End For
    For each model m in M
    m'←TrainModel(m, T1)
    R←DetectIntrusions(T2, m')
    P←EvaluatePerformance(T3, R)
      Display R
      Display P
    End For
    ensembleModel←StackingEnsemble(M)
    ensembleModel.params←HyperparameterTuning(parameterSpace, D', F)
      R←DetectIntrusions(T2, ensembleModel)
      P←EvaluatePerformance(T3, R)
      Display R
      Display P
    End
```

Algorithm 1

Ensemble Learning-Based Intrusion Detection (ELBID)

The ELBID algorithm, Algorithm 1, uses a combination of machine learning (ML) models to detect intrusions in the CICIDS2017 dataset. The algorithm involves several steps including data preprocessing, hybrid feature engineering, data splitting, hyperparameter tuning, model training, intrusion detection, and performance evaluation for individual ML models as well as a stacking ensemble model. The ELBID algorithm addresses the challenge of detecting intrusions in network traffic by employing an ensemble learning approach, which combines the strengths of multiple ML models to improve the accuracy and robustness of intrusion detection. The CICIDS2017 dataset contains a comprehensive collection of benign and malicious network activities, forming the foundation for this study. The dataset undergoes initial data preprocessing to handle missing values, normalize features, and make it ready for feature engineering. A hybrid approach to feature engineering techniques is then applied to enhance the dataset's features. The preprocessed dataset is split into training, testing, and validation sets to ensure robust model evaluation.

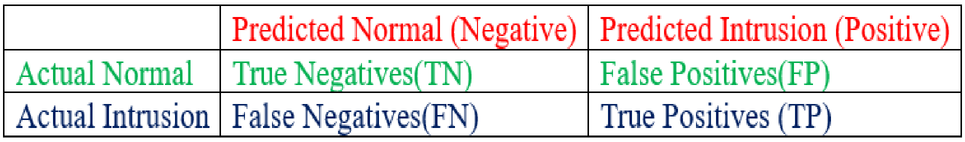
Each ML model undergoes hyperparameter tuning to optimize its performance. Following this, each model is trained on the training set and used to detect intrusions in the testing set. The performance of each model is evaluated on the validation set, and results are displayed. Finally, a stacking ensemble model is created by combining the individual models, and its performance is also evaluated and displayed. The ELBID algorithm demonstrates the effectiveness of ensemble learning in intrusion detection. By combining multiple ML models, the algorithm achieves improved performance over individual models. The detailed process of data preprocessing, feature engineering, and model tuning ensures that the models are well-prepared to handle the complex task of intrusion detection. Using the CICIDS2017 dataset provides a challenging and realistic environment for testing the algorithm's capabilities. The ELBID algorithm presents a robust approach to network intrusion detection. The algorithm achieves high performance by leveraging the strengths of multiple ML models and employing a rigorous process of data preparation and model tuning. This study highlights the potential of ensemble learning in enhancing the accuracy and reliability of intrusion detection systems.

3.6 Dataset Details

The dataset identified as CICIDS2017 [31] is utilized in empirical studies to build and analyze intrusion detection systems. This dataset is widely recognized and utilized in research for intrusion detection. It comprises network traffic data that is valuable for training and assessing intrusion detection systems. The dataset contains various types of network traffic, including normal and malicious traffic, making it an essential resource for developing and evaluating intrusion detection algorithms. Researchers frequently use this dataset to gauge the effectiveness of various intrusion detection techniques and algorithms. CICIDS2017 provides a comprehensive range of features for each network flow or connection, encompassing statistical, payload, and flow features. This enables researchers to apply diverse machine learning and statistical techniques to identify and categorize network intrusions. Professionals and experts in cybersecurity, machine learning, and network security harness the CICIDS2017 dataset for training and testing intrusion detection models, comparing different algorithms, and enhancing the accuracy of their detection systems.

3.7 Performance Evaluation

A confusion matrix, which can be seen in Figure 4, is a tool used to evaluate the performance of intrusion detection models (IDS). It summarizes the model's predictions compared to the actual ground truth labels.



	Predicted Normal (Negative)	Predicted Intrusion (Positive)
Actual Normal	True Negatives(TN)	False Positives(FP)
Actual Intrusion	False Negatives(FN)	True Positives (TP)

Figure 4. Confusion matrix

True Positives (TP): Instances where the model correctly predicts an intrusion (attack) when the actual label is also an intrusion. True Negatives (TN): Instances where the model correctly predicts normal behavior (no attack) when the actual label is also normal. False Positives (FP): Instances where the model incorrectly predicts an intrusion when the exact label is normal (Type I error). False Negatives (FN): Instances where the model incorrectly predicts normal behavior when the actual label is an intrusion (Type II error).

Precision (p) = $\frac{TP}{TP+FP}$ (1)

Recall (r) = $\frac{TP}{TP+FN}$ (2)

F1-score = $2 * \frac{(p * r)}{(p+r)}$ (3)

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$ (4)

The confusion matrix provides several metrics for evaluating the performance of an Intrusion Detection System (IDS). Accuracy measures the overall correctness of the model's predictions. Precision (or Positive Predictive Value) measures the proportion of predicted intrusions that are actually intrusions. Recall (or Sensitivity or True Positive Rate) measures the proportion of actual intrusions that are correctly predicted by the model. F1 Score calculates the harmonic mean of precision and recall, providing a single metric that balances both measures.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results related to intrusion detection using an optimized machine-learning framework and various algorithms. The experiments were conducted on a personal computer running Windows 11 operating system with a 13th Gen Intel(R) Core(TM) i7-1355U processor, running at 1700 Mhz with 10 cores, 12 logical processors, and 16 GB of RAM. The machine learning models were implemented using the Scikit-learn library. The models used in the empirical study consist of XGBoost, Random Forest, LightGBM, and Gradient Boosting.

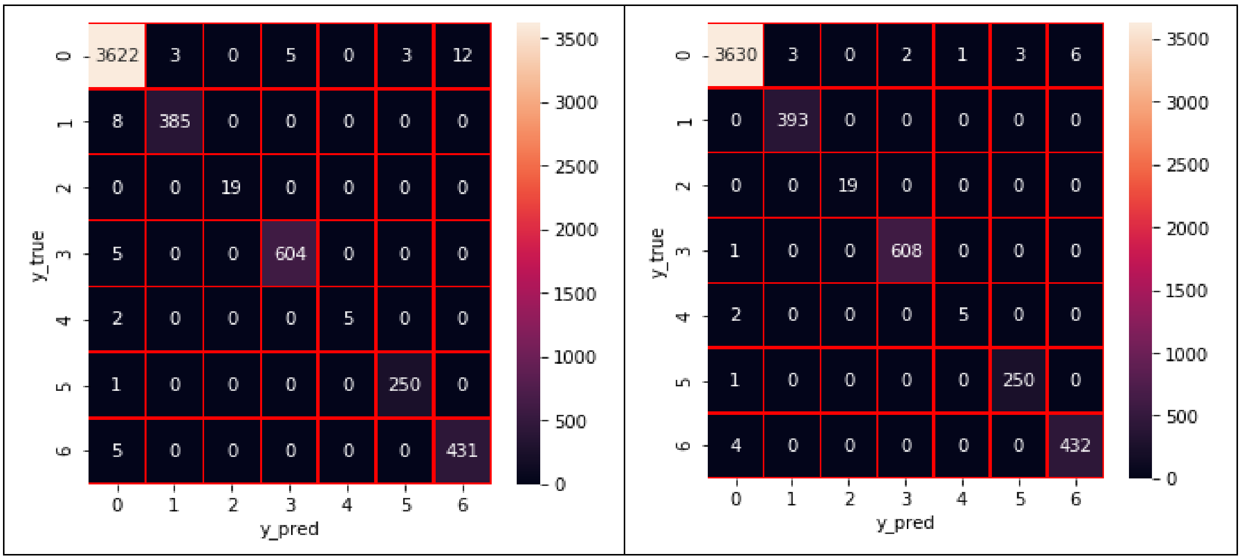


Figure 5. Confusion matrix of XGB (left) and XGB with HO (right)

Figure 5 shows two confusion matrices placed side by side. They demonstrate the performance of two different models: XGB (on the left) and XGB with HO (on the right). These matrices compare the predicted labels (y_pred) to the true labels (y_true) for seven classes (0 to 6). The color-coding on each matrix forms a gradient from light to dark, indicating the frequency of occurrences, with lighter colors representing higher frequencies. The left confusion matrix displays 3,622 correct predictions for class 0, while the right matrix for XGB with HO shows 3,630 correct predictions for class 0. Both models exhibit similar performance across other classes, with minor differences in the number of correct and incorrect predictions. Red lines have been used to distinguish the boundaries between different classes for clarity. Overall, the XGB with HO model seems to have slightly improved performance compared to the standard XGB model, as seen in the higher number of correct predictions for certain classes.

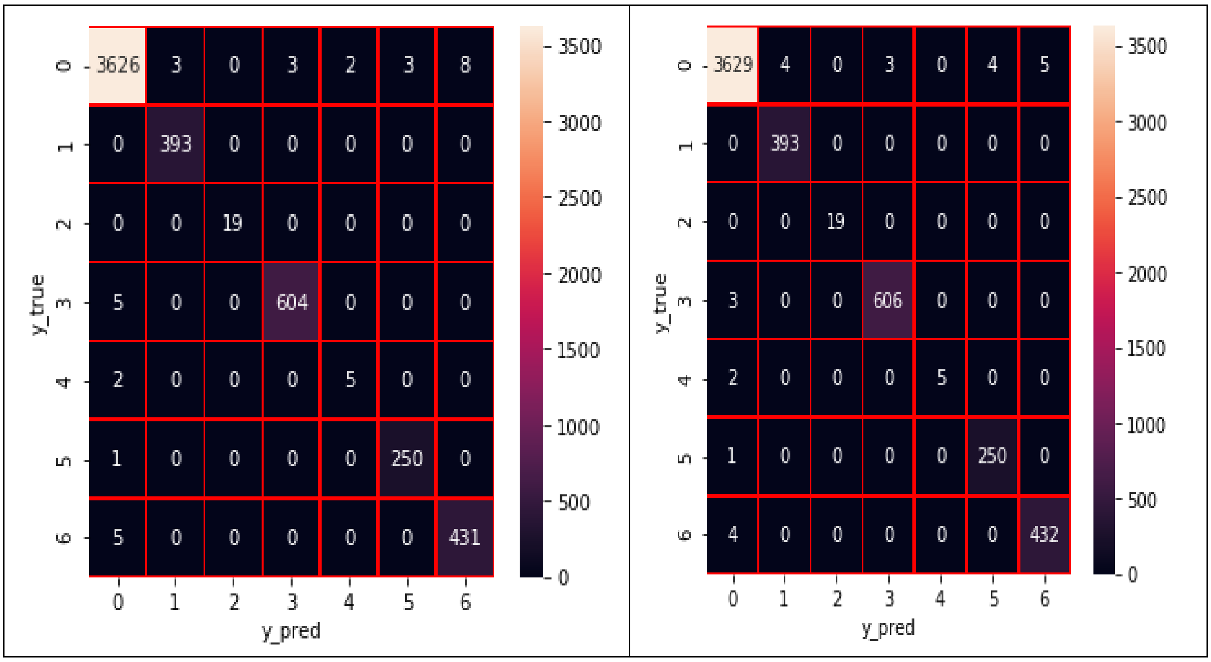


Figure 6. Confusion matrix of RF (left) and RF with HO (right)

Figure 6 presents two confusion matrices placed side by side. These matrices illustrate the performance of two models: RF (Random Forest) on the left and RF with HO (Hyperparameter Optimization) on the right. Each confusion matrix compares the predicted labels (y_pred) against the true labels (y_true) for seven classes (0 to 6). The color gradient from light to dark represents the frequency of occurrences, with lighter shades indicating higher counts. In the left matrix, the RF model correctly predicts 3,626 instances of class 0, whereas the right matrix for the RF with HO model correctly predicts 3,629 instances of class 0. Both models demonstrate similar accuracy across other classes, with minor variations in correct and incorrect predictions. Red lines are used to delineate the class boundaries for better visibility. Overall, the RF with HO model shows a slight improvement in performance over the standard RF model, as indicated by a marginal increase in correct predictions for certain classes.

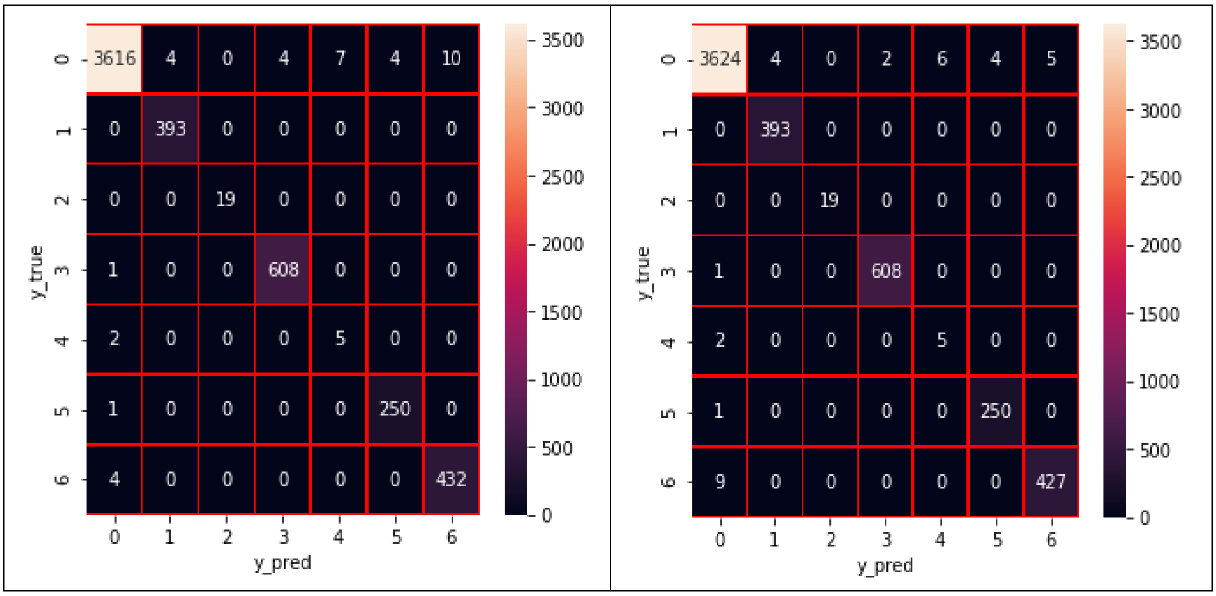


Figure 7. Confusion matrix of LGBM (left) and LGBM with HO (right)

Figure 7 displays two confusion matrices side by side, showing the performance of two models: LGBM (LightGBM) on the left and LGBM with HO (Hyperparameter Optimization) on the right. These matrices compare the predicted labels (y_pred) with the true labels (y_true) across seven classes (0 to 6). The color gradient ranging from light to dark indicates the frequency of occurrences, with lighter shades representing higher frequencies. In the left matrix, the LGBM model correctly predicts 3,616 instances of class 0, while the right matrix for the LGBM with HO model correctly predicts 3,624 cases of class 0. Both models exhibit similar accuracy across other courses, with slight differences in the number of correct and incorrect predictions. The red lines help to separate the class boundaries clearly. Overall, the LGBM with HO model demonstrates a minor improvement in performance over the standard LGBM model, as seen by the slight increase in correct predictions for certain classes.

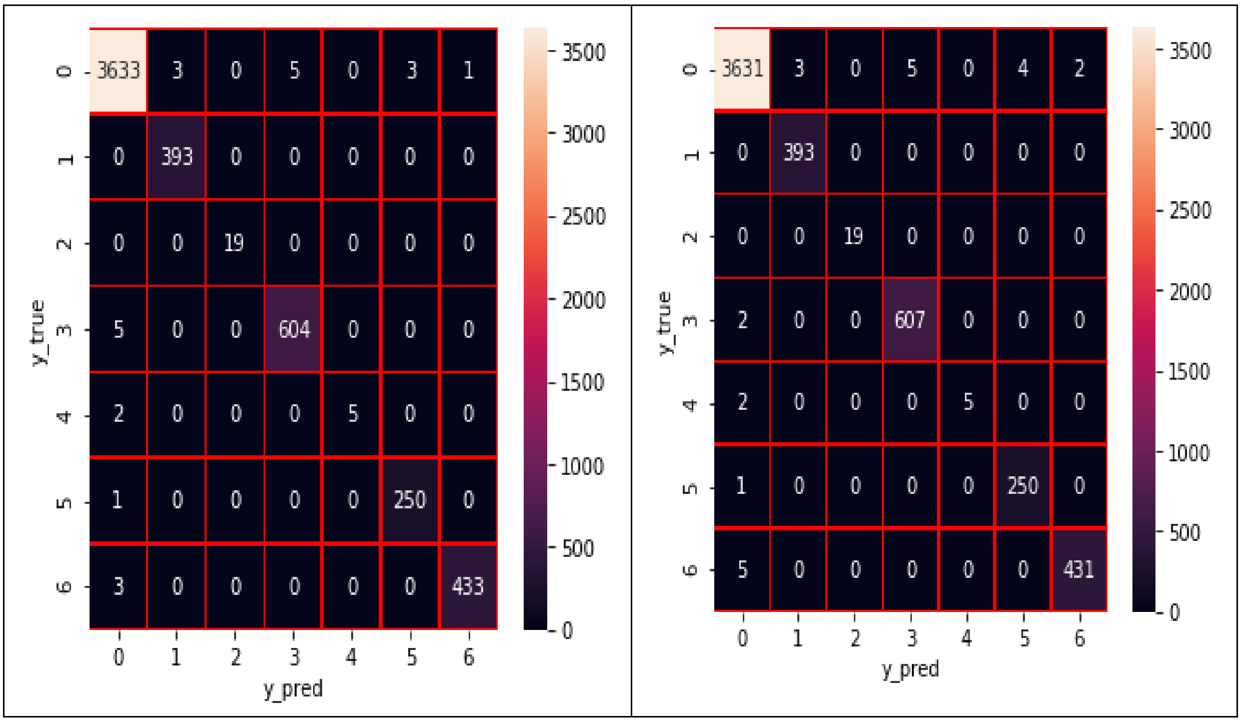


Figure 8. Confusion matrix of Gradient Boosting (left) and Gradient Boosting with HO (right)

Figure 8 displays two confusion matrices comparing the performance of two gradient-boosting models in a classification task. The left matrix illustrates the performance of a standard Gradient Boosting model, while the right matrix corresponds to a Gradient Boosting model with Hyperparameter Optimization (HO). The matrices show the true labels (y_true) along the vertical axis and the predicted labels (y_pred) along the horizontal axis, with labels ranging from 0 to 6. Each cell in the matrices represents the count of instances for each true-predicted label pair, with darker cells indicating higher counts. Both models exhibit high accuracy for label 0, with over 3600 correctly predicted instances, and moderate performance for labels 3, 5, and 6. Labels 1, 2, and 4 have significantly fewer instances, suggesting areas for potential model improvement.

Table 1. Performance of different models for intrusion detection

IDS Model	Precision	Recall	F1-Score	Accuracy
LGBM	0.914112	0.91356	0.913744	0.91356
LGBM (with HO)	0.91448	0.914112	0.914204	0.914112
GB	0.916044	0.916044	0.916044	0.916044
GB (with HO)	0.91586	0.91586	0.91586	0.91586
RF	0.91448	0.91448	0.91448	0.91448
RF (with HO)	0.915492	0.915492	0.915492	0.915492
XGB	0.912456	0.912364	0.912364	0.912364
XGB (with HO)	0.915952	0.916044	0.915952	0.916044
Ensemble	0.93577	0.93577	0.93577	0.93577
Ensemble (with HO)	0.94582	0.945915	0.94582	0.945915

As presented in Table 1, various machine learning models and the ensemble model were used to evaluate intrusion detection performance with and without hyperparameter optimization.

Table 2. Performance comparison

IDS Model	Precision	Recall	F1-Score	Accuracy
KNN [32]	0.896241	0.895962	0.896055	0.889917
RF [32]	0.917259	0.916236	0.916701	0.916329
DBN [33]	0.918468	0.915585	0.91698	0.918747
Ensemble with HO(Proposed)	0.94582	0.945915	0.94582	0.945915

As presented in Table 2, the performance of the proposed model used for intrusion detection is compared against many state-of-the-art models.

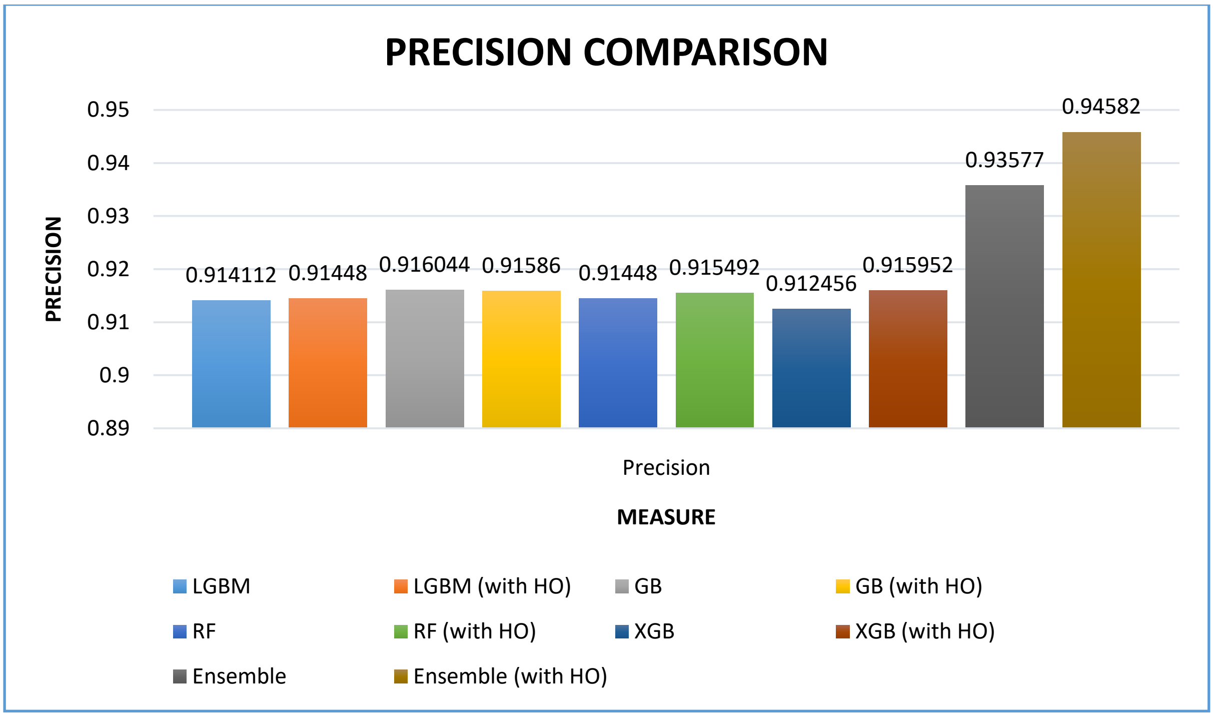


Figure 9. Precision comparison among intrusion detection models

In Figure 9, a precision comparison of various intrusion detection models is illustrated. This includes both standard and hyperparameter-optimized (HO) versions of models such as LightGBM (LGBM), Random Forest (RF), Gradient Boosting (GB), and XGBoost (XGB), as well as their ensemble versions. The precision of each model is shown on the y-axis, with values ranging from 0.89 to 0.95. On the x-axis, the different models are listed and color-coded for clarity. The standard versions of LGBM, RF, and GB exhibit similar precision values, around 0.914 to 0.916. However, the XGB model with hyperparameter optimization achieves the highest precision at 0.94582, followed by the ensemble model also with hyperparameter optimization at 0.93577. This indicates that hyperparameter optimization significantly enhances the precision of XGB and ensemble models in detecting intrusions.

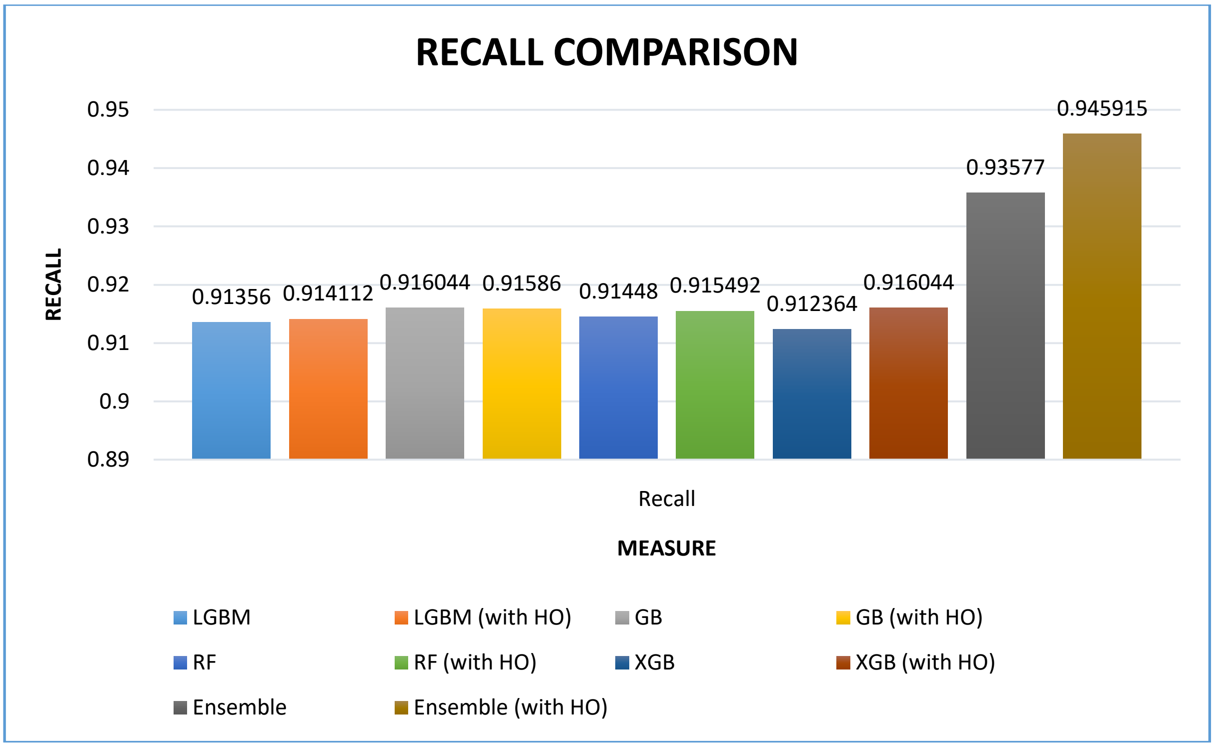


Figure 10. Recall comparison among intrusion detection models

In Figure 10, we have a comparison of recall scores for different intrusion detection models, including both their standard and hyperparameter-optimized (HO) versions. The models analyzed are LightGBM (LGBM), Random Forest (RF), Gradient Boosting (GB), and XGBoost (XGB), as well as ensemble versions of these models. The recall values, which range from 0.89 to 0.95, are displayed on the y-axis, while the x-axis shows the models, each represented by different colors. The standard versions of LGBM, RF, and GB show similar recall values, approximately 0.913 to 0.916. Notably, the XGB model with HO achieves the highest recall at 0.945915, followed by the ensemble model with HO at 0.93577. These results emphasize the substantial improvement in recall achieved through hyperparameter optimization, particularly for the XGB and ensemble models in intrusion detection tasks.

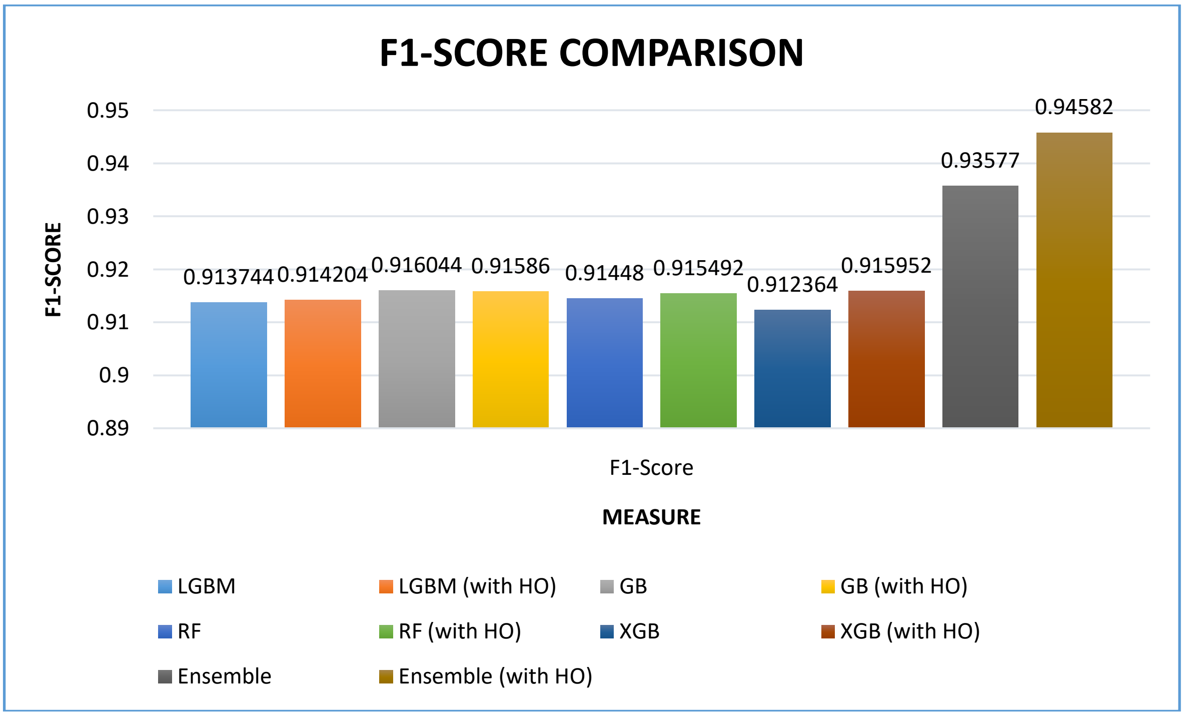


Figure 11. F1 score comparison among intrusion detection models

Figure 11 compares F1 scores for various intrusion detection models, including both standard and hyperparameter-optimized (HO) versions. The evaluated models are LGBM, LGBM (with HO), RF, RF (with HO), GB, XGB, XGB (with HO), Ensemble, and Ensemble (with HO). The F1 scores for each model are as follows: LGBM (0.913744), LGBM (with HO) (0.914204), RF (0.916044), RF (with HO) (0.91586), GB (0.91448), XGB (0.915492), XGB (with HO) (0.912364), Ensemble (0.93577), and Ensemble (with HO) (0.94582). The chart indicates that the "Ensemble (with HO)" model achieved the highest F1 score of 0.94582, demonstrating superior performance in intrusion detection compared to the other models. The next best performing model is the "Ensemble" without HO, with an F1 score of 0.93577. Other models have similar F1 scores, all around the 0.91-0.92 range. This comparison highlights the effectiveness of ensemble methods, particularly when hyperparameter optimization (HO) is applied.

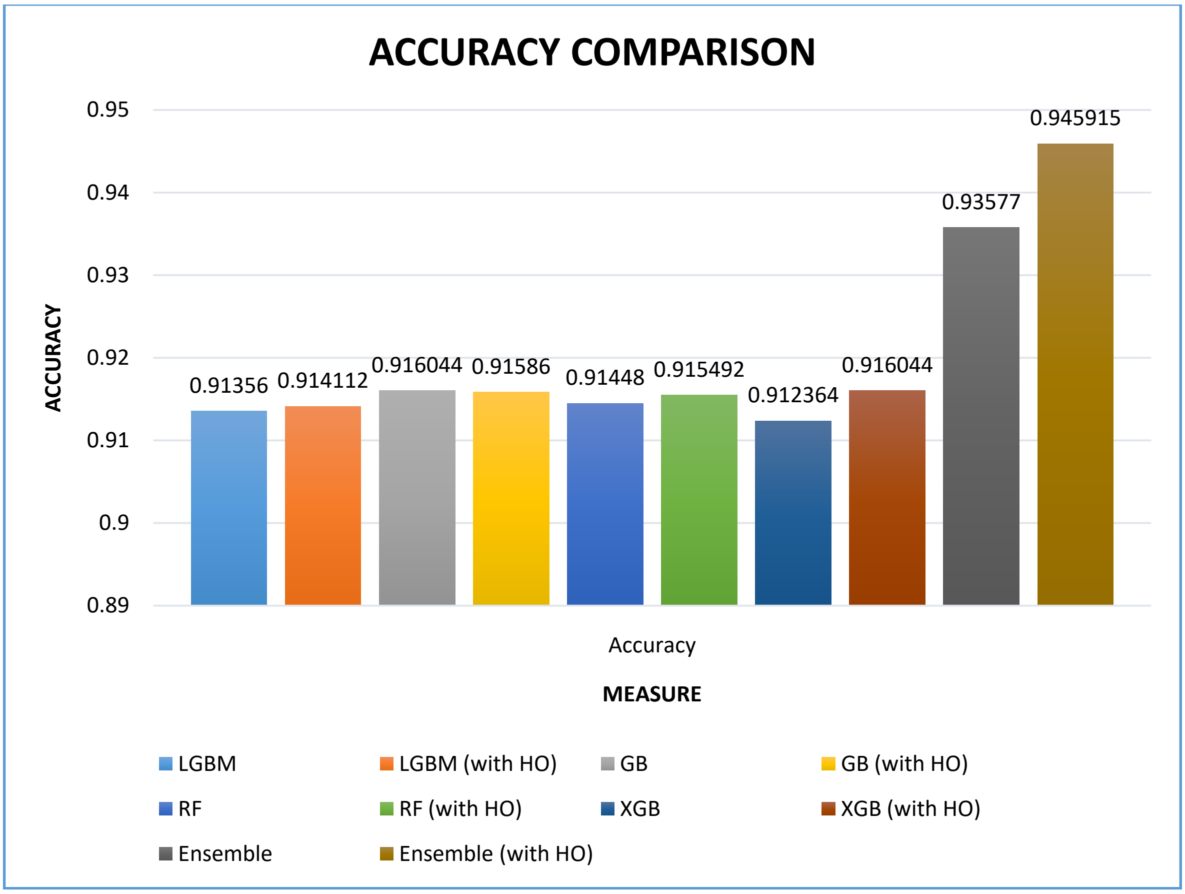


Figure 12. Accuracy comparison among intrusion detection models

In Figure 12, we compare the accuracy scores of different intrusion detection models, including both standard and hyperparameter-optimized (HO) versions. The evaluated models are LGBM, LGBM (with HO), RF, RF (with HO), GB, GB (with HO), XGB, XGB (with HO), Ensemble, and Ensemble (with HO). The accuracy scores for each model are as follows: LGBM (0.91356), LGBM (with HO) (0.914112), RF (0.916044), RF (with HO) (0.91586), GB (0.91448), GB (with HO) (0.916044), XGB (0.915492), XGB (with HO) (0.912364), Ensemble (0.93577), and Ensemble (with HO) (0.945915). According to the chart, the "Ensemble (with HO)" model achieved the highest accuracy score of 0.945915, demonstrating superior performance in intrusion detection compared to the other models. The next best performing model is the "Ensemble" without HO, with an accuracy score of 0.93577. Other models have relatively similar accuracy scores, all hovering around the 0.91-0.92 range. This comparison emphasizes the effectiveness of ensemble methods, particularly when hyperparameter optimization (HO) is applied.

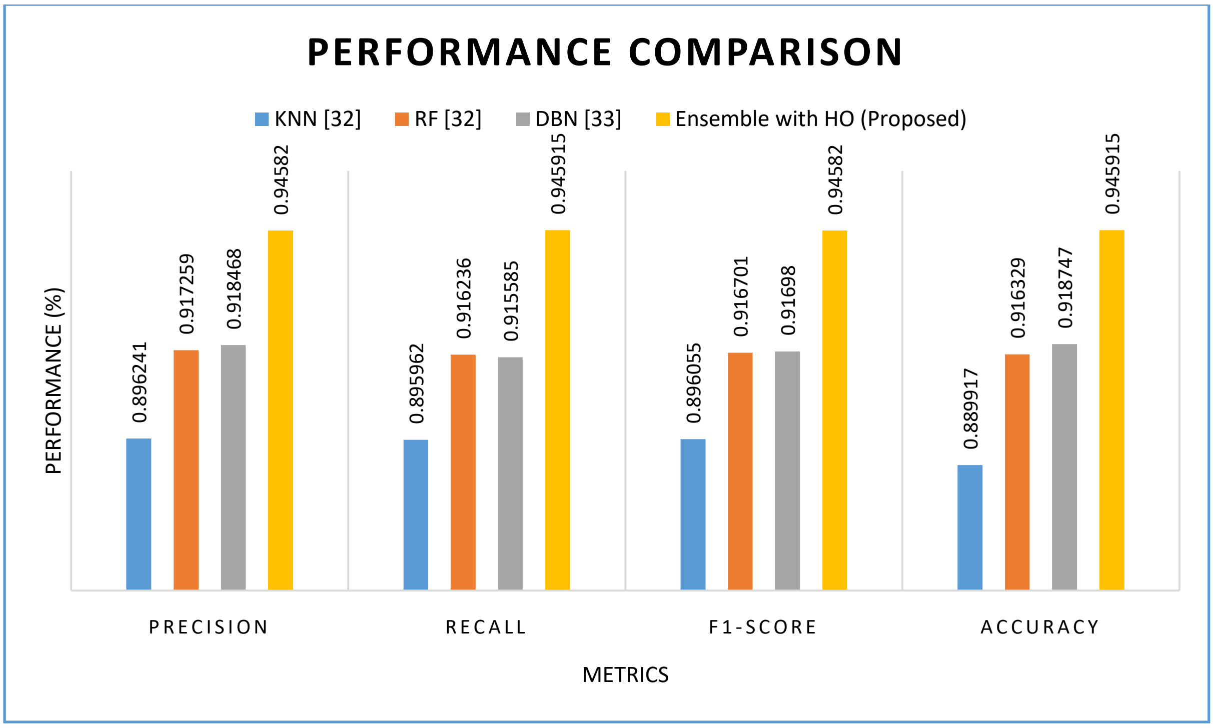


Figure 13. Performance comparison of the proposed and existing intrusion detection models

Figure 13 compares the performance of various intrusion detection models based on four metrics: Precision, Recall, F1-Score, and Accuracy. The models compared are K-Nearest Neighbors (KNN), Random Forest (RF), Deep Belief Network (DBN), and the proposed Ensemble with HO. The results indicate that the proposed Ensemble with HO model outperforms the other models across all metrics, scoring 0.94582 for Precision, 0.94515 for Recall, 0.94582 for F1-Score, and 0.94515 for Accuracy. This suggests that the proposed model is more effective in detecting intrusions than the other tested models

5. DISCUSSION

In today's world, there has been a significant increase in security issues related to networks and information systems. Cyberattacks on information systems and networks have been on the rise, necessitating continuous research to address this threat by developing various security measures to protect the cyberspace. While traditional security measures have been effective in safeguarding data and information systems, the emergence of artificial intelligence has made it essential to enhance cybersecurity through learning-based approaches suitable for solving complex problems. The utilization of machine learning models for intrusion detection systems is crucial in the current era. However, the effectiveness of machine learning models in intrusion detection depends on the quality of the training data. Therefore, it is imperative to optimize machine learning models through hyperparameter tuning and feature engineering to enhance cyber attack detection performance. The framework proposed in this paper includes such optimizations and demonstrates that the proposed models could outperform existing ones. However, it is important to note that the proposed system in this paper has certain limitations, as discussed in section 5.1.

5.1 Limitations

The intrusion detection system described in this paper has some limitations. The system was evaluated using a specific data set, making it challenging to generalize the findings without incorporating a diverse range of data. Additionally, the system does not utilize advanced neural network models, such as deep learning techniques. Furthermore, it does not consider the hybridization of machine learning models, which is another significant limitation.

6. CONCLUSION AND FUTURE WORK

This paper presents a machine learning-based framework specifically designed to create an efficient intrusion detection system. The framework includes a hybrid feature engineering methodology and hyperparameter tuning using Bayesian Optimization for machine learning models. We propose an ensemble of fine-tuned machine-learning models for superior intrusion detection performance. Our proposed algorithm, known as Ensemble Learning-Based Intrusion Detection (ELBID), utilizes machine learning models to intelligently detect intrusions. The ensemble model we have developed surpasses many existing ML models in intrusion detection, achieving an impressive accuracy of 94.59%. As a result, our optimized ensemble model can be seamlessly integrated into real-world applications, significantly enhancing cybersecurity. In the future, we aim to enhance our framework with deep learning models and diverse data sets.

REFERENCES

[1] Mishra, Preeti; Varadharajan, Vijay; Tupakula, Uday and Pilli, Emmanuel S. (2018). A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection. IEEE Communications Surveys & Tutorials, 1–1. http://doi:10.1109/COMST.2018.2847722

[2] Saranya, T.; Sridevi, S.; Deisy, C.; Chung, Tran Duc and Khan, M.K.A.Ahamed (2020). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. Procedia Computer Science, 171, 1251–1260. http://doi:10.1016/j.procs.2020.04.133

[3] Ilhan FiratKilincer; Fatih Ertam and Abdulkadir Sengur; (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study . Computer Networks. http://doi:10.1016/j.comnet.2021.107840

[4] Çavuşoğlu, Ünal (2019). A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence. http://doi:10.1007/s10489-018-01408-x

[5] da Costa, Kelton A.P.; Papa, João P.; Lisboa, Celso O.; Munoz, Roberto and de Albuquerque, Victor Hugo C. (2019). Internet of Things: A Survey on Machine Learning-based Intrusion Detection Approaches. Computer Networks, S1389128618308739–. http://doi:10.1016/j.comnet.2019.01.023

[6] Sultana, Nasrin; Chilamkurti, Naveen; Peng, Wei and Alhadad, Rabei (2018). Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications. http://doi:10.1007/s12083-017-0630-0

[7] Sai Kiran, K.V.V.N.L.; Devisetty, R.N. Kamakshi; Kalyan, N. Pavan; Mukundini, K. and Karthi, R. (2020). Building a Intrusion Detection System for IoT Environment using Machine Learning Techniques. Procedia Computer Science, 171, 2372–2379. http://doi:10.1016/j.procs.2020.04.257

[8] Othman, Suad Mohammed; Ba-Alwi, Fadl Mutaher; Alsohybe, Nabeel T. and Al-Hashida, Amal Y. (2018). Intrusion detection model using machine learning algorithm on Big Data environment. Journal of Big Data, 5(1), 34–. http://doi:10.1186/s40537-018-0145-4

[9] Abdulnaser A. Hagar and Bharti W. Gawali. (2023). Implementation of Machine and Deep Learning Algorithms for Intrusion Detection System. *Springer*, pp.1-21. https://doi.org/10.1007/978-981-19-1844-5_1

[10] Gao, Xianwei; Shan, Chun; Hu, Changzhen; Niu, Zequn and Liu, Zhen (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. IEEE Access, 1–1. http://doi:10.1109/ACCESS.2019.2923640

[11] Emad E. Abdallah, Wafa' Eleisah and Ahmed Fawzi Otoom. (2022). Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey. *Elsevier*. 201, pp.1-8. https://doi.org/10.1016/j.procs.2022.03.029

[12] Injadat, Mohammad Noor; Moubayed, Abdallah; Nassif, Ali Bou and Shami, Abdallah (2020). Multi-Stage Optimized Machine Learning Framework for Network Intrusion Detection. IEEE Transactions on Network and Service Management, 1–1. http://doi:10.1109/TNSM.2020.3014929

[13] Ziadoon Kamil Maseer; Robiah Yusof; Nazrulazhar Bahaman; Salama A. Mostafa and Cik Feresa Mohd Foozy; (2021). Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset . IEEE Access. http://doi:10.1109/access.2021.3056614

[14] Resende, Paulo Angelo Alves and Drummond, André Costa (2018). A Survey of Random Forest Based Methods for Intrusion Detection Systems. ACM Computing Surveys, 51(3), 1–36. http://doi:10.1145/3178582

[15] Puneet Himthani and Ghanshyam Prasad Dubey. (2022). Application of Machine Learning Techniques in Intrusion Detection Systems: A Systematic Review. *Springer*, p.97–105. https://doi.org/10.1007/978-981-16-4538-9_10

[16] R. Dilip, N. Samanvita, R. Pramodhini,S. G. Vidhya,Bhagirathi S. Telkar. (2022). Performance Analysis of Machine Learning Algorithms in Intrusion Detection and Classification. *Springer*, p.283–289. https://doi.org/10.1007/978-3-031-07012-9_25

[17] Abhishek Verma and Virender Ranga. (2019). Machine Learning Based Intrusion Detection Systems for IoT Applications. *Springer*, pp.1-24. https://doi.org/10.1007/s11277-019-06986-8

[18] Alhajjar, E., Maxwell, P., & Bastian, N. (2021). Adversarial machine learning in Network Intrusion Detection Systems. Expert Systems with Applications, 186, 115782. http://doi:10.1016/j.eswa.2021.115782

[19] Zong, Wei; Chow, Yang-Wai and Susilo, Willy (2019). Interactive three-dimensional visualization of network intrusion detection data for machine learning. Future Generation Computer Systems, S0167739X18331091–. http://doi:10.1016/j.future.2019.07.045

[20] Roger R. dos Santos, Eduardo K. Viegas, Altair O. Santin and Vinicius V. Cogo. (2023). Reinforcement Learning for Intrusion Detection: More Model Longness and Fewer Updates. *IEEE*. 20(2), pp.2040 - 2055. http://DOI:10.1109/TNSM.2022.3207094

[21] Lan Liu; Pengcheng Wang; Jun Lin and Langzhou Liu; (2021). Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning . IEEE Access. http://doi:10.1109/access.2020.3048198

[22] Wang, Maonan; Zheng, Kangfeng; Yang, Yanqing and Wang, Xiujuan (2020). An Explainable Machine Learning Framework for Intrusion Detection Systems. IEEE Access, 8, 73127–73141. http://doi:10.1109/ACCESS.2020.2988359

[23] Ahmad, I.; Basheri, M.; Iqbal, M. J. and Raheem, A. (2018). Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access, 1–1.http://doi:10.1109/ACCESS.2018.2841987

[24] Caminero, Guillermo; Lopez-Martin, Manuel and Carro, Belen (2019). Adversarial environment reinforcement learning algorithm for intrusion detection. Computer Networks, S1389128618311216–. http://doi:10.1016/j.comnet.2019.05.013

[25] Sethi, Kamalakanta; Sai Rupesh, E.; Kumar, Rahul; Bera, Padmalochan and Venu Madhav, Y. (2019). A context-aware robust intrusion detection system: a reinforcement learning-based approach. International Journal of Information Security. http://doi:10.1007/s10207-019-00482-7

[26] Rahman, Md Arafatur; Asyharia, A. Taufiq; Leong, L.S.; Satrya, G.B.; Tao, M. Hai and Zolkipli, M.F. (2020). Scalable Machine Learning-Based Intrusion Detection System for IoT-Enabled Smart Cities. Sustainable Cities and Society, 102324–. http://doi:10.1016/j.scs.2020.102324

[27] Li, XuKui; Chen, Wei; Zhang, Qianru and Wu, Lifa (2020). Building Auto-Encoder Intrusion Detection System Based on Random Forest Feature Selection. Computers & Security, 101851–. http://doi:10.1016/j.cose.2020.101851

[28] Park, Seong-Taek; Li, Guozhong and Hong, Jae-Chang (2018). A study on smart factory-based ambient intelligence context-aware intrusion detection system using machine learning. Journal of Ambient Intelligence and Humanized Computing. http://doi:10.1007/s12652-018-0998-6

[29] ‘Gu, Jie and Lu, Shan (2021). An effective intrusion detection approach using SVM with naÃ¯ve Bayes feature embedding. Computers & Security, 103, 102158–. http://doi:10.1016/j.cose.2020.102158

[30] Ieracitano, Cosimo; Adeel, Ahsan; Morabito, Francesco Carlo and Hussain, Amir (2019). A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach. Neurocomputing, S0925231219315759–. http://doi:10.1016/j.neucom.2019.11.016

[31] Intrusion detection evaluation dataset (CIC-IDS2017). Available at https://www.unb.ca/cic/datasets/ids-2017.html.

[32] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detectiondataset and intrusion traffic characterization,”in Proc. Int. Conf. Inf. Syst. Secur. Privacy, 2018, pp.108–116.

[33] W. Elmasry, A. Akbulut, and A. H. Zaim, “Evolving deep learning architectures for networkintrusion detection using a double PSO meta- heuristic,” Comput. Networks, vol. 168, 2020.