International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Machine Learning Framework with Feature Selection and Hyperparameter Tuning Optimizations for Intrusion Detection

Sayyada Mubeen*[1], Dr. Harikrishna Kamatham [2]

Submitted:10/03/2024 **Revised**: 25/04/2024 **Accepted**: 02/05/2024

Abstract: The recent increase in cyber-attacks has made cyber security upgrading an on-going task. Heuristics were the foundation of traditional security systems; they were designed to identify intrusions depending on how they were detected. But as artificial intelligence (AI) techniques like machine learning (ML) have become more popular, learning-based models have shown to be effective because of their capacity to constantly learn from tagged data. The research indicates that when training samples are not of the intended amount and quality, supervised learning-based machine learning models perform worse at detecting intrusions. Utilizing the performance of ML models with certain tweaks is crucial. The paper, which intends to create an intrusion detection system based on machine learning and feature engineering, is motivated by this. We proposed two algorithms named Hybrid Feature Selection (HFS) and Learning based Intrusion Detection (LbID). We evaluate the system with the CICIDS2017 dataset. Binomial and multi-class classification is applied in the implementation of intrusion detection systems. With 94.22% accuracy, the RF model has the best binomial classification accuracy. Decision Tree has the greatest accuracy (91.67%) when it comes to multi-class classification without optimizations. RF exhibits the maximum accuracy of 93.46% in the case of multi-class classification with optimizations.

Keywords: *Artificial Intelligence, Machine Learning, Intrusion Detection*

1. Introduction

The modern period has made cyber security extremely difficult (Razan et al., 2018). Machine learning is crucial since traditional security is unable to fend off advanced attacks (Kamran et al., 2020). Through the application of machine learning, intrusion detection systems can be able to recognize both known and unknown attacks with incremental information based on learning. It is essential to carry out research on improving cyber security in light of the rise in cyber-attacks. Security requires ongoing system and intrusion detection mechanism enhancement since it is not a one-time task. That is the motivation for this work, which is notable for what it is expected to reveal about suitable machine learning (ML) models for training datasets, optimization techniques, and intrusion detection. This project aims to provide a machine learning framework with optimizations for efficient intrusion detection. The following are the study questions. Question 1: Is it feasible to use machine learning models to create an intrusion detection system? RQ2: Do improvements like feature selection and hyper parameter tuning have an impact on how successful machine learning models are at detecting intrusions? These study questions are predicated on the conclusions of the literature. While the second question looks at two ML model tweaks for improving intrusion detection efficiency, the first question explores the potential of machine learning models for detecting intrusion. Empirical research is used in this project.

These are this study's principal contributions. First, a framework in order to identify intrusions using machine learning is created and put into place. Second, several machine learning models are assessed for multi-class classification, feature engineering, and hyperparameter tuning as well as binomial classification for intrusion detection. We proposed two algorithms named Hybrid Feature Selection (HFS) and Learning based Intrusion Detection (LbID). This is how the remainder of the document is arranged. The literature on the different intrusion detection techniques now in use is reviewed in Section 2. We outline our study technique in Section 3. The design specification is presented in Section 4. Implementation and assessment information are given in Sections 5 and 6. A further assessment of the research's relevance is given in Section 6. The section presents the findings and the research's future directions.

2. Related Work

Lee et al. [1] without requiring specialized expertise, machine learning increases accuracy in network intrusion detection systems. The suggested C-ELM technique adds hidden neurons gradually in order to achieve quick learning and high attack detection rates. Abdulhammed et al. [2] Threats are growing, so network security is

[1] *Research Scholar, Dept of CSE, Malla Reddy University, Hyderabad And Assistant Professor, Muffakham Jah College of Engineering and Technology, Hyderabad, India, sayyada.mubeen@mjcollege.ac.in*
ORCID ID: 0009-0007-8704-7897
[2] *Associate Dean, School of Engineering, Malla Reddy University, Hyderabad, India,kamathamhk@gmail.com*
ORCID ID: 0009-0004-0221-4894
** Corresponding Author Email: sayyada.mubeen@mjcollege.ac.in*

essential. By reducing features using Auto-Encoder and PCA, machine learning helps intrusion detection systems (IDS) by increasing DR, F-Measure, FAR, and accuracy. CombinedMc, which has a higher accuracy rate on CICIDS2017, is the measure recommended by the research for performance comparison. Alhajjar et al. [3] examined adversarial instances in Network Intrusion Detection Systems (NIDS) and using deep learning and evolutionary computing to evade detection. High misclassification rates in the results point to NIDS's susceptibility to hostile disruption. Bertoli et al. [4] tackled the issue of network intrusion detection using out-of-date datasets. The revised model deployment process consists of five phases that provide optimal performance while minimizing resource use. Upcoming initiatives include testing on light-weight operating systems, refining dataset production, and expanding categorization. Ahmad et al. [5] examined ML and DL-based NIDS techniques, emphasizing their benefits and drawbacks. It highlights current developments, highlights dataset constraints, and suggests future research avenues for lighter and more effective DL-based NIDS.

Zaman et al. [6] depended on the identification of anomalies in network traffic. The shortcomings of signature-based IDS have given rise to ML-based methods. RBF performs best among the seven machine learning techniques evaluated in this study using data from Kyoto 2006+. Further study employing an Ensemble technique appears to be promising as well. Sultana et al. [7] for better safety, SDN-based NIDS incorporates ML and DL techniques. The current state of ML/DL work in SDN-based NIDS is examined in this paper, along with issues and potential solutions. Li et al. [8] defend against network attacks, machine learning and data mining are essential. Using the KDD 99 dataset, fuzzy logic and artificial neural networks are studied and assessed. Future research directions and challenges in AI-based cyber-attack defence are emphasized. Parashar et al. [9] required an intrusion detection system (IDS), and this research proposes a network intrusion detection solution that employs machine learning stacking ensembles. ID3, XGBoost, and Random Forest all performed well. Taher et al. [10] supervised machine learning system finds that an ANN with wrapper feature selection beats SVM in classifying network traffic as dangerous or benign.

Carrion et al. [11] purpose of this work is to improve NIDS evaluation using the UGR'16 dataset and a structured approach. In today's linked world, Network Intrusion Detection Systems (NIDSs) are indispensable, yet there are no established techniques for assessing them. Halimaa et al. [12] for network security, intrusion detection systems (IDS) are essential since they look for unusual activities. SVM and Naïve Bayes are two machine learning approaches that increase the accuracy of IDS. Phadke et al. [13] demanded for dynamic intrusion detection systems utilizing machine learning has arisen due to the growing risks posed by the internet. Network Intrusion Detection accuracy is improved by a number of approaches. Dini et al. [14] with superior feature selection for precise intrusion detection, KNN marginally beats ANN in this dataset. An increasingly common use of machine learning is anomaly detection, especially for network infiltration. LAN traffic analysis is done for protection using K-nearest neighbours (KNN) and artificial neural network (ANN) techniques. Mishra et al. [15] examined the limits of algorithms using machine learning to identify intrusions, with a focus on particular strategies for each kind of assault, these techniques are examined.

Costa et al. [16] invested in cutting edge intrusion detection systems are motivated by worries about global security. IoT machine learning presents difficulties, focusing on accuracy and efficiency gains. Seraphim et al. [17] generated due to technological improvements, which emphasize the need for intrusion detection systems (IDS). Network security vulnerabilities are efficiently detected by applying machine learning techniques. A variety of methods are surveyed, including Random Forest, SVM, and k-means. A suggested two-level strategy that makes use of both simple and complex learning algorithms, such as Artificial Neural Networks (ANN), seeks to improve the efficiency of IDS. Devi et al. [18] expanded of the digital age underscores the necessity of automated security. Security for networks is provided by Network Intrusion Detection Systems (NIDS), although threat identification is the primary function of Intrusion Detection Systems (IDS). DARPA 1999, KDD 99, and NSL-KDD cup 99 are examples of out dated datasets that are insufficient for evaluation since they do not contain up-to-date attack data. Examining the CIDDS-001 dataset, this study compares approaches and selects Verma et al.'s latest machine learning method as the preferred implementation. Almseidin et al. [19] Implemented IDS counterattacks, but issues arise from changing tactics. Research using the KDD dataset focuses on false positives and negatives. Attack distribution was highlighted by the decision table and random forest's strong performance. Tests with 60,000 data emphasized the importance of precise intrusion detection. Liu et al. [20] concealed due to uneven network load, which makes it challenging for NIDS to detect them. Through the creation of fresh samples and enhanced classification capabilities, the DSSTE algorithm corrects imbalance. Experiments with SVM, XGBoost, LSTM, AlexNet, Mini-VGGNet, and RF on the NSL-KDD and CSE-CIC-IDS2018 datasets show that DSSTE outperforms other methods.

Wu et al. [21] suggested approach emphasizes efficiency and accuracy in network intrusion detection using machine learning. To lower false alerts, it uses random forest. Megantara and Ahmad [22]provided for advancements in many industries, yet cyber threats still exist. Intrusion Detection Systems (IDS) use anomaly and signature detection to find intrusions. This work proposes a hybrid machine learning approach that includes choosing features and reducing data to boost the accuracy of detecting R2L assaults. More research is necessary since there are still issues with improving IDS for outliers and unbalanced data. Abubakar and Pranggono [24] revolutionized by Software-Defined Networks (SDN), yet security issues develop. It is imperative that Intrusion Detection Systems (IDS) be integrated with SDN. Attacks are detected by an IDS testbed using Snort, demonstrating the potential for improved machine learning and flow-based IDS. Khan and Gumaei [25] focused on accuracy and efficiency while evaluating artificial intelligence classifiers for detecting network intrusions. When tested in 10-fold cross validation and given test modes, Decision Trees (DT), Random Forests (RF), Hierarchical Trees (HT), and K-Nearest Neighbours (KNN) perform well on the KDD99 and UNSW-NB15 datasets.

Rincy and Gupta [26] presented NID-Shield, a combined intrusion detection system utilizing machine learning methods and CAPPER for feature selection. Evaluations using the NSL-KDD and UNSW-NB15 datasets demonstrate low FPR and promising accuracy. Li et al. [27] improved via machine learning. In terms of sensitivity, less feature reduction, and detection, feature extraction outperforms. Selection allows for quicker training and greater accuracy gains. Jaradat et al. [28] used three classifiers and feature selection, to identify intrusions, machine learning searches networks for irregularities. The accuracy rate according to the results. Fan and You [29] founded by network monitoring, with XGBoost, Random Forest, and Decision Tree performing best. Manage data using integrated models to prevent over fitting. Catboost and Logistic Regression function rather well and are feature-sensitive. Plain Bayesian models and support vector machines do badly. Research on security benefits from findings. Zhang et al. [30] achieved excellent recall and precision by merging random forest and decision branches to handle network intrusion detection difficulties.

Talukder et al. [31] with ML-based analysis, network intrusion detection is essential to cyber security. The new model, which uses PCA, RO, and clustering, outperforms in accuracy. Improves security posture and lowers false alert rates. Dhaliwal et al. [32] through the reduction of harmful network data, XGBoost on the NSL-KDD dataset improves data integrity. As technology progresses, network security becomes increasingly important. Zhang et al. [33] with decision boundary entropy, the IDTSRF model enhances feature selection, recall, and accuracy. Data volume and attribute relevance provide obstacles for network intrusion detection. Chimphlee [34] described a two-phase approach that prioritizes the identification of anomalies in network data by feature selection and imbalance management.

3. Proposed Framework

This project aims to provide a machine learning framework with optimizations for efficient intrusion detection. Figure 1 illustrates the research approach that was used to accomplish the study's goal. Several datasets are initially located and examined. After a thorough analysis of several datasets, it is discovered that CICIDS2017 has superior benchmarking and is relatively new, with a variety of incursion types. As a result, the dataset used for the empirical research is CICIDS2017. It was discovered that a number of machine learning models, such as XGBoost, Random Forest, Decision Tree, and Extra Trees, were suitable for this investigation. These are all supervised learning models based on trees. When training with high-quality data, these models function effectively. If not, their performance deteriorates. Feature selection and hyperparameter adjustment might be looked at for leveraging the performance of ML models as a solution to this issue, according to the literature study. The suggested intrusion detection system's design is depicted in Figure 1 and is based on the methodology's findings.

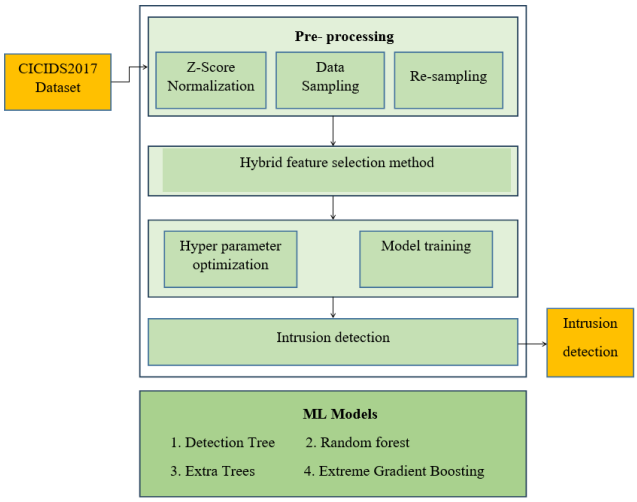


Fig 1: Architecture of the proposed intrusion detection system

The suggested system's functioning is explained here. Pre-processing of the provided dataset involves resampling the data to eliminate over fitting, Z-score normalization, zeroing out blank values, and splitting the information into sets for testing and training. The technique employed to address the issue of class

imbalance is called SMOTE. Following pre-processing, feature engineering is applied to the data to identify the highest performing features. Every machine learning model that was employed in the empirical investigation is then tuned using hyperparameters. This procedure helps to enhance a certain machine learning model's performance by determining the best values for its hyperparameters. After that, the models are trained using training data. The trained model is saved for later use when training is finished. The test data is examined for intrusions using the learnt model. Results for intrusion detection are produced by it.

3.1 Feature Engineering

Calculating each feature's value in class label prediction and choosing every feature is the process of feature selection with the highest significance to train machine learning classifiers. A general filter-based feature selection method is shown in Figure 2.

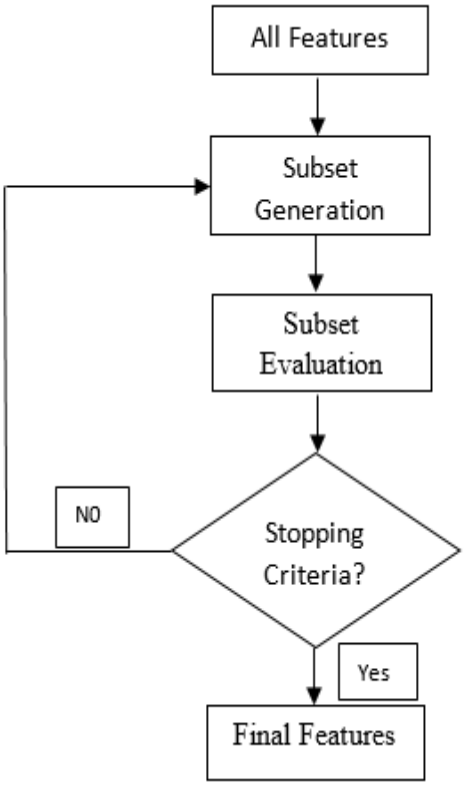


Fig 2: Illustrates feature selection process

This method referred to as "filter-based approach," uses a measure to calculate feature importance and elects the best contributing features depending on the feature importance threshold. Three measures are utilized in this study to choose features. It is a combination of the t-test, entropy, and Fisher criteria filter techniques.

The mean value of μ_1 (i) and σ_i (i) is described. The number of patterns in the null and unitary class, n_1 and n_0. KL-range Distance Kullback Liebler PDistribution of probabilities QProbability distribution of the target

The notations used in the suggested feature engineering approach are displayed in Table 1. For feature selection, the Fisher index computation method described in [36] is frequently employed. It's calculated using Equation 1.

$$F(i) = |\frac{\mu_1(i)-\mu_0(i)}{\sigma_1^2(i)-\sigma_0^2(i)}| \quad (1)$$

Every characteristic or variable linked to the underlying dataset is given weight by the figure index. As covered in [37], the t-test is another popular filter technique. It computes according to Equation 2 and is utilized to assess the relative relevance of every attribute.

$$t(i) = |\frac{\mu_1(i)-\mu_0(i)}{\sqrt{\frac{\sigma_1^2(i)}{n_1}+\frac{\sigma_0^2(i)}{n_0}}}| \quad (2)$$

Relative entropy is only one more popular filter technique. Another name for it is Kullback-Leibler divergence, which is covered in [38]. It represents the distance between two probability distributions.

$$KL(p,q)=\sum_i p_i log_2(\frac{p_i}{q_i}) \quad (3)$$

The suggested hybrid technique makes use of these three measurements in order to select the most advantageous characteristics for brain stroke detection study.

Algorithm 1: Hybrid Feature Selection (HFS)

Input: CICIDS2017 dataset D, threshold th

Output: Features *F*

1. F←getAllFeatures(D)
   Hybrid Approach
2. For each *f* in *F*
3. Compute fisher score using Eq. 1
4. Compute tscore using Eq. 2
5. Compute relative entropy using Eq. 3
6. Compute mean of the above
7. Save mean score for each feature to a map M
8. End For
9. F←SelectFeatues(th, F)
10. Return F
11. End

Algorithm 1: Hybrid Feature Selection

As shown in Algorithm 1, it receives as inputs the threshold th and the dataset D, and outputs the features that have been chosen. There are many phases of the algorithm's execution. These processes are referred to as identifying every attribute, considering every feature from every attribute to generate the whole feature space, using a hybrid filter-based strategy to choose features, and ultimately making the final feature selection based

on threshold. With 0.32 as threshold, our algorithm could provide best results in intrusion classification.

3.2 Hyperparameter Optimization

Appropriate parameter adjustment is facilitated by hyperparameter optimization of machine learning models. Bayesian optimization is used to do this. In Table 2, many parameters optimal for machine learning models are shown. The ML model's enhanced XGBoost parameters include learning rate and maximum depth. The criteria for the number of RF estimators, the count of estimators, the maximum features, the maximum depth, the minimum samples leaf, and the minimum samples split. Criteria, Decision Tree, Maximum Features, Maximum Depth, Minimum Samples Split, Minimum Samples Leaf, and Additional Trees The models of the Hyperparameter to optimization are Criteria, Minimum Samples Leaf, Minimum Samples Split, Maximum Features, Maximum Depth, and Count of Estimators.

3.3 Machine Learning Techniques

Four machine learning models are covered by the suggested system's intrusion detection procedure, as seen in Figure 3. Every model stands alone from the others.

3.3.1 Decision Tree

The decision tree is one of the best supervised learning methods available for regression and classification applications. It creates a tree structure that looks like a flowchart, where each leaf node has the class name, each internal node has an attribute test, and each branch represents a test result. The training data is split recursively into subsets based on attribute values until a stopping condition is satisfied, such as the maximum depth of the tree or the minimum number of samples required to split a node. The Decision Tree technique determines which attribute to split the data into during training by calculating a metric such as entropy or Gini impurity, which measures the degree of impurity or unpredictability in the subsets. The goal is to identify the feature that maximizes the reduction of contaminants or the information gained following the split.

3.3.2 Random Forest

Random Forest algorithm is one of the most powerful machine learning techniques for tree learning. A lot of Decision Trees are constructed by it throughout the training process. At each division, a random subset of characteristics is measured using a piece of the data set at random during the tree-building process. Overall prediction accuracy is improved and the likelihood of over fitting is decreased as a result of the unpredictability that provides variance to the individual trees. The algorithm averages, or votes, across the results from each tree to provide predictions. The results of this cooperative decision-making process are precise and reliable because of the assistance of several trees and their insights. Given its reputation for handling complex data, reducing over fitting, and generating precise predictions, Regression and classification issues are frequently addressed with random forests.

3.3.3 Extra Trees

Many decision trees are produced via the extra trees approach in a manner akin to the random forest strategy, but it does so in a random manner without replacing each tree. As a consequence, each tree in the dataset has a unique sample. A specific number of randomly selected features from the whole feature set are also included in each tree. It’s most important and unique feature is that it allows extra trees to randomly select a splitting value for a feature. Rather of using entropy or Gini to get a locally optimum value, the approach splits the data and selects a split value at random. This leads to the trees becoming diversified and uncorrelated.

3.3.4 XGBoost

Machine learning models may be trained using XGBoost, a scalable and effective distributed gradient boosting toolbox. By merging the predictions of several weak models using an ensemble learning approach, a stronger prediction is produced. "Extreme Gradient Boosting," or XGBoost, has become one of the most well-known and widely used machine learning algorithms due to its ability to handle large datasets and generate cutting-edge outcomes in a variety of machine learning tasks, such as regression and classification. Because of its proficiency in managing missing values, XGBoost is a valuable tool for handling missing values in real-world data. This function eliminates the need for extensive pre-processing. Furthermore, XGBoost's inherent parallel processing capabilities enable quick model training on big datasets.

3.4 Intrusion Detection Approaches

Two methods for intrusion detection are used in the design of the experiments. Binomial categorization of test samples into BENIGN and INTRUSION is the name of the first method. We refer to the second strategy as multi-class categorization. As Table 3 illustrates, different groups represent various types of infiltration. BENIGN 0 Bot 1, BruteForce 2, DoS 3, Infiltration 4, PortScan 5, WebAttack 6 is the class index for Intrusion Class. These classes make up the multiclass classification. This study employs machine learning models for multi-class as well as binomial classification. Implementation details are presented in Section 5. An intrusion detection system based on Python is implemented. SMOTE tools are used in the implementation phase to rectify class imbalances.

Minority classes are kept while the bulk of classes have their data sampled. K-Means clustering is the method used to group data samples. Confusion matrices for every model and testing strategy, including binomial and multi-class, are among the outputs generated during implementation.

Algorithm 2: Learning based Intrusion Detection (LbID)

Input: CICIDS2017 dataset D, ML models M

Output: Intrusion detection results R, performance statistics P

1. D'←PreProcess(D)
2. (T1, T2)←DataPreparation(D')
3. F1←runHFSAlgorithm(T1)
4. F2←runHFSAlgorithm(T2)
5. For each model in M
6. Train m using F1
7. Save m
8. End For
9. For each model in M
10. R←DetectIntrusions(F2)
11. P←PerformanceEvaluation(R, groundTruth)
12. Print R
13. Print P
14. End For

Algorithm 2: Learning based Intrusion Detection (LbID)

As presented in Algorithm 2, it takes the given dataset and a threshold value as inputs. The algorithm has provision for pre-processing and data preparation in such a way that it generates training data and test data denoted as T1 and T2 respectively. Then the algorithm has an iterative process where each model is trained with corresponding features. The feature selection algorithm proposed in this paper is reused by this algorithm to generate best performing features in both the training data and test data. In the process of training the extracted feature are used to train the models. In the process of testing the features from test data are used in order to predict all possible intrusions. In the process of evaluation, the algorithm predictions are compared against ground truth values in order to find the performance of each machine learning model.

3.5 Evaluation Methodology

The efficacy of the suggested intrusion detection system is assessed based on many standards that are often used in the literature. Recall, accuracy, precision, and F1-score are the names of these measurements. They are computed using the confusion matrix displayed in Figure 3.

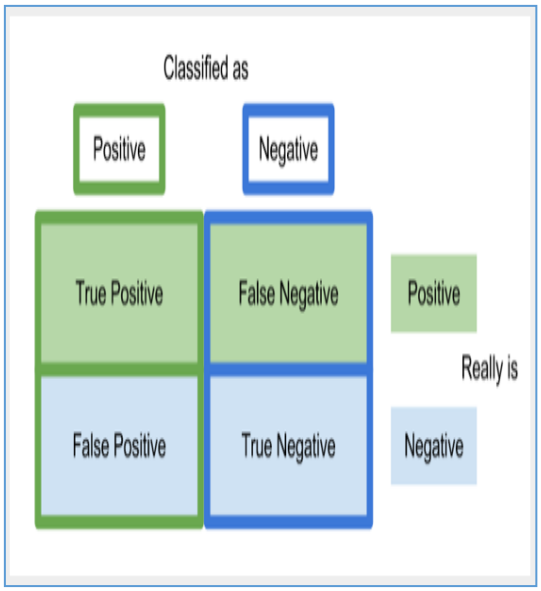


Fig 3: Confusion matrix

A situation is referred to be True Positive (TP) if the algorithm predicts that the test sample provided also exhibits intrusion. A situation is referred to as True Negative (TN) if both the algorithm prediction and the test sample that was provided are benign. False Positive (FP) occurs when the algorithm predicts INTRUSION when the test sample provided is BENIGN. False Negative (FN) is the term used to describe the situation when the algorithm predicts BENIGN but the test sample provided has INTRUSION.

$$\text{Precision (p)} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{Recall (r)} = \frac{TP}{TP+FN} \quad (5)$$

$$\text{F1-score} = 2 * \frac{(p * r)}{(p+r)} \quad (6)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

Accuracy, F1-score, precision, and recall are calculated using Eqs. 4, 5, 6, and 7, based on the four examples displayed in the confusion matrix.

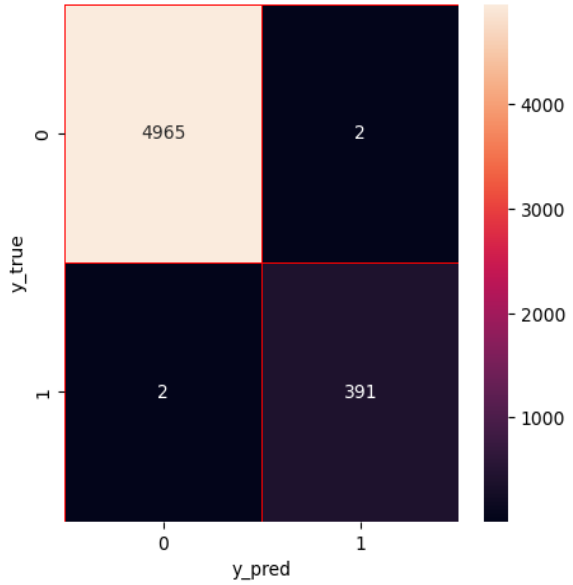
4. Experimental Results

The suggested intrusion detection system is assessed in order to determine how well various ML models perform both with and without optimizations. Analysis of the outcomes using multi-class and binomial classification is also included in the evaluation. The study's findings are provided here. Three categories— classification using binomials, multiclass classification with optimization, and multiclass classification without optimization—are used to show the findings. The CICIDS2017 dataset [35] is employed in our study.

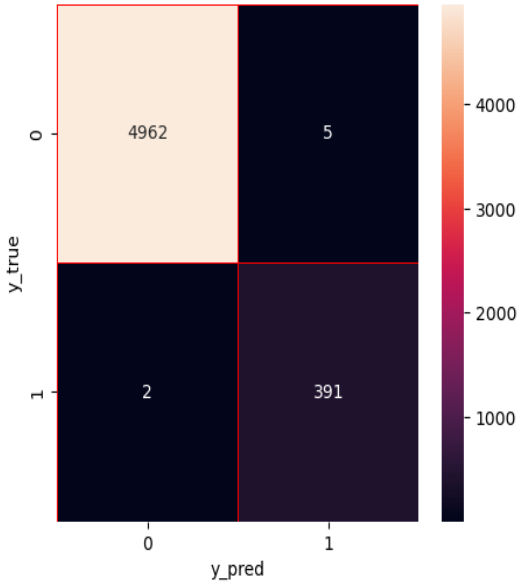
4.1 Results of Binomial Classification

Confusion matrix-based statistics and performance statistics are used to present the binomial classification results.

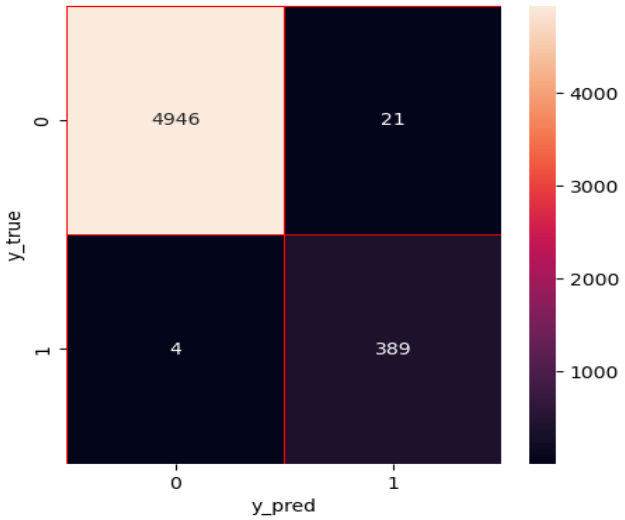
XGBoost Confusion matrix



Random Forest Confusion matrix



Extra Trees Confusion matrix



Decision Tree Confusion matrix

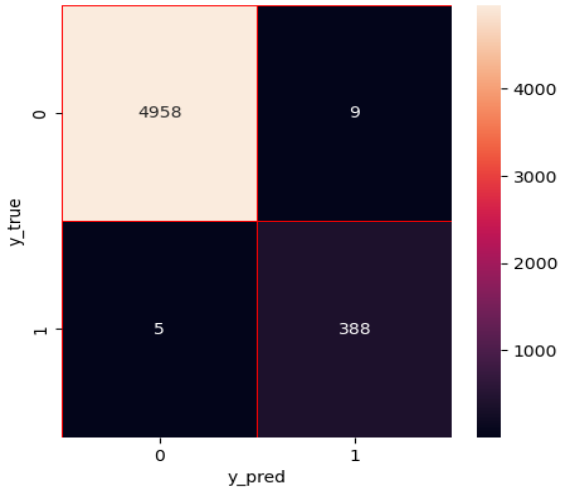


Fig 4: Results of binary classification in terms of confusion matrix

For each of the four machine learning models, Comparison is made between the ground truth labels and the anticipated labels as shown in Figure 4. Table 4 displays the performance data that were calculated using the confusion matrix.

Table 4: Intrusion detection performance of models with binomial classification

Bionomial Classification				
Intrusion Detection Model	Precision	Recall	F1-score	Accuracy
XGBoost	0.8337	0.8789	0.8557	0.8927
ExtraTrees	0.8989	0.9354	0.9113	0.9354
DecisionTree	0.9134	0.9393	0.9118	0.9393
Random Forest	0.9278	0.9422	0.9214	0.9422

Every detection model's performance was displayed in Table 4 along with additional metrics, including accuracy. All of the models were shown to be able to classify binomial data with greater than 99% accuracy.

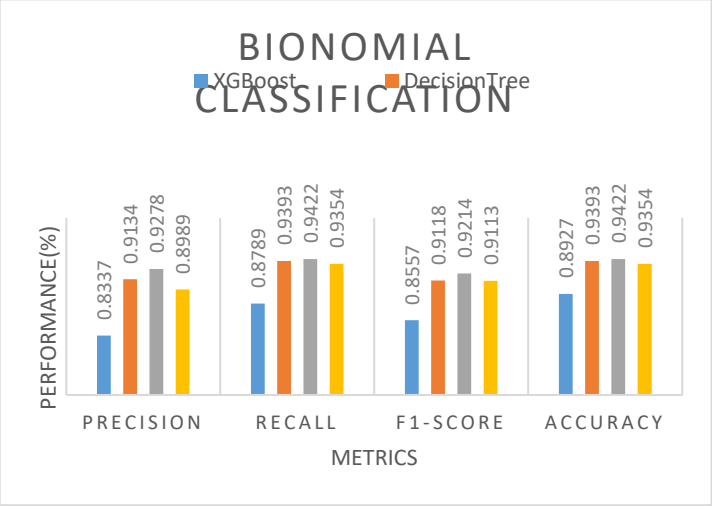


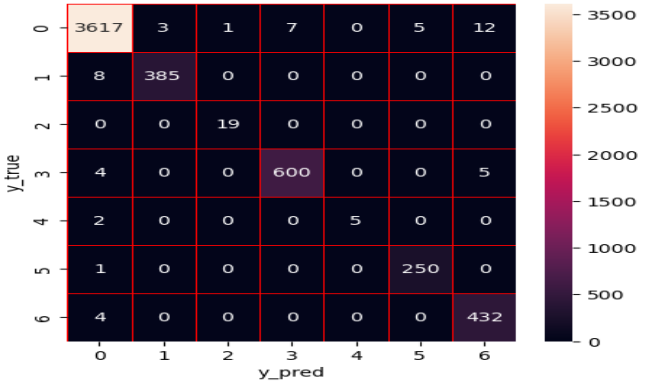
Fig 5: Results of bionomial classification

Every model in the experiment examining the intrusion detection performance of binomial classification models is shown to operate more efficiently. Precision is 83.37%, recall is 87.89%, F1-score is 85.57%, and accuracy is 89.27% for XGBoost performance. Precision, recall, accuracy, and F1-score for the DecisionTree performance are 91.34%, 93.93%, and 91.18%, respectively. Precision is 92.78%, recall is 94.22%, F1-score is 92.14%, and accuracy is 94.22% for Random Forest performance. Precision is 89.89%, recall is 93.54%, F1-score is 91.13%, and accuracy is 93.54% for ExtraTrees performance.The Random Forest model has the best binomial classification accuracy.

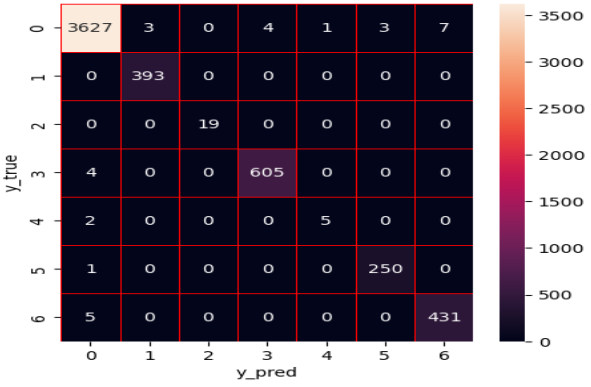
4.2 Results of Multi-Class Classification without Optimization

Based on confusion matrix-based statistics and performance data, the outcomes of the multi-class classification without optimization are displayed.

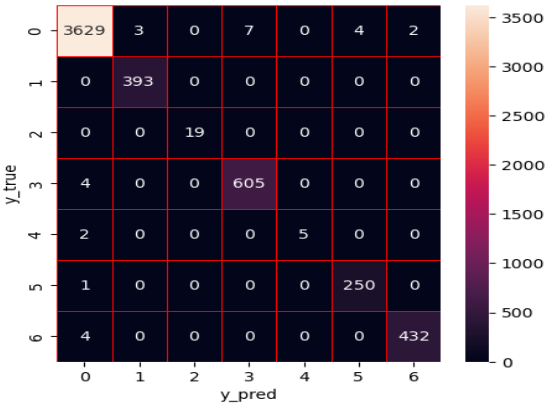
XGBoost Confusion matrix



Random Forest Confusion matrix



Extra Trees Confusion matrix



Decision Tree Confusion matrix

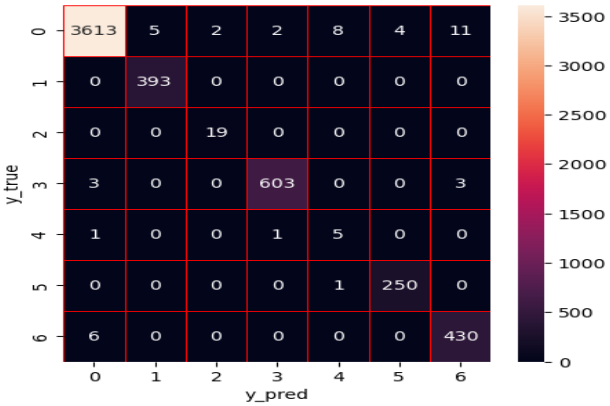


Fig 6: Results of multi-class classification without optimization in terms of confusion matrix

As seen in Figure 6, the projected labels for every class and each of the four machine learning models are compared to the ground truth labels. The performance statistics that were computed using the confusion matrix are shown in Table 5.

Table 5: Intrusion detection performance of models with multi-class classification without optimization

Multi-Class Classification Performance (Without Optimization)				
Intrusion Detection Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
Random Forest	0.7661	0.9206	0.8052	0.8862
XGBoost	0.8499	0.8076	0.8754	0.9054
ExtraTrees	0.8942	0.9135	0.9941	0.9089
DecisionTree	0.9278	0.9461	0.9214	0.9167

Table 5 displays the effectiveness of each detection model for multi-class classification in terms of accuracy and other metrics without tuning. It was shown that every model could classify binomial data with greater than or equal to 99% accuracy.

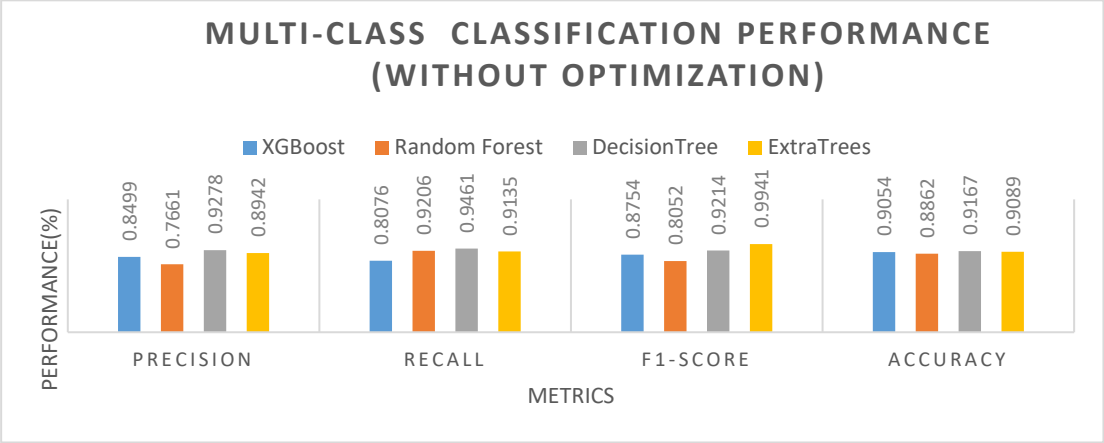


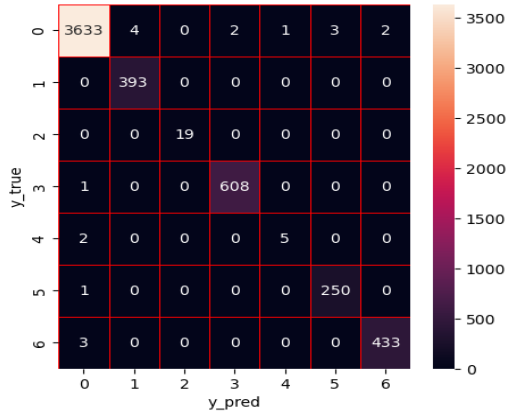
Figure 7: Results of multi-class classification without optimizations

When doing the multi-class classification experiment without optimizations each model is observed with the efficiency of Anomaly detection. The higher in value for each metric indicates better performance. The XGBoost performance Precision is 84.99%, Recall is 80.76%, F1-score is 87.54% and Accuracy is 90.54%. DecisionTree performance Precision is 92.78%, Recall is 94.61%, F1-score is 92.14% and Accuracy is 91.67%. Random Forest performance Precision is 76.61%, Recall is 92.06%, F1-score is 80.52% and Accuracy is 88.62%. ExtraTrees performance Precision is 89.42%, Recall is 91.35%, F1-score is 99.41% and Accuracy is 90.89%.The highest accuracy for without any optimizations, multi-class categorization is demonstrated by DecisionTree model.

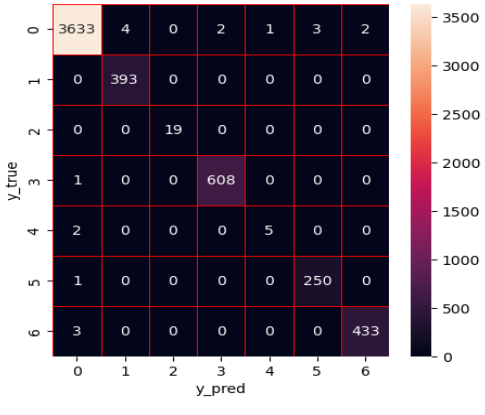
4.3 Results of Multi-Class Classification with Optimization

Performance statistics and statistics based on the confusion matrix are used to present the results of the multi-class classification with optimization.

XGBoost Confusion matrix



Random Forest Confusion matrix



Extra Trees Confusion matrix

Decision Tree Confusion matrix

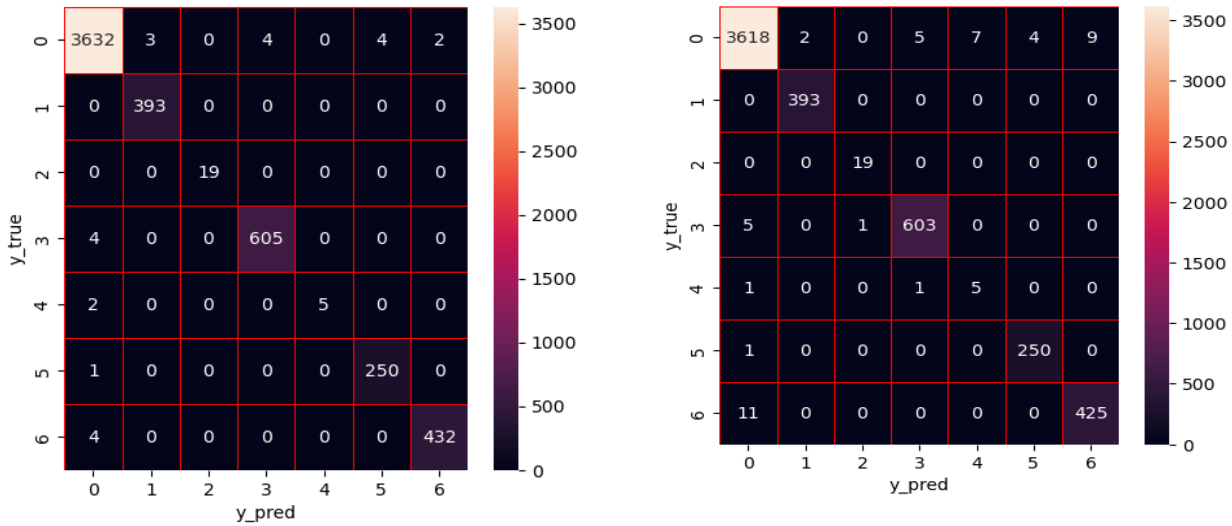


Fig 8: Results of multi-class classification with optimization in terms of confusion matrix

Figure 8 illustrates the comparison between the projected labels for each of the four machine learning models and each class and the ground truth labels. Table 6 presents calculated performance metrics that were obtained from the confusion matrix.

Table 6: Intrusion detection performance of models with multi-class classification with optimization

Multi-Class Classification Performance (With Optimization)				
Intrusion Detection Model	Precision	Recall	F1-score	Accuracy
DecisionTree	0.8499	0.9026	0.8754	0.8436
ExtraTrees	0.7661	0.8485	0.8052	0.8937
XGBoost	0.8864	0.8561	0.8709	0.9134
Random Forest	0.8337	0.8789	0.8557	0.9346

Table 6 displays the effectiveness of every detection model about multi-class classification without optimization in terms of accuracy and other parameters. Results demonstrated that every model were able to classify binomial data with more than 99% accuracy. When compared to models without optimization, the performance of models with optimization is somewhat better.

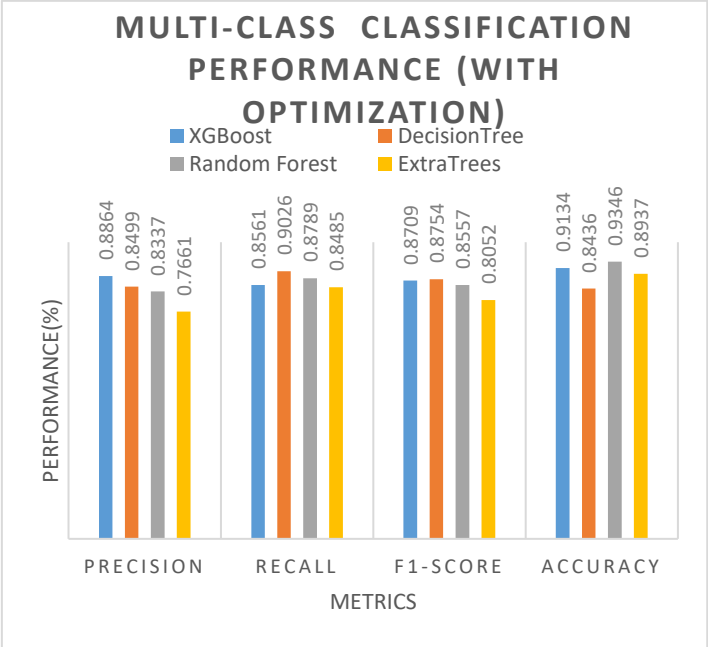


Fig 9: Results of multi-class classification with optimizations

Throughout the multi-class classification experiment with optimizations each model is observed with the efficiency of Anomaly detection. The higher in value for each metric indicates better performance. The XGBoost performance Precision is 88.64%, Recall is 85.61%, F1-score is 87.09% and Accuracy is 91.34%. Decision Tree performance Precision is 84.99%, Recall is 90.26%, F1-score is 87.54% and Accuracy is 84.36%. Random Forest performance Precision is 83.37%, Recall is 87.89%, F1-score is 85.57% and Accuracy is 93.46%. ExtraTrees performance Precision is 76.61%, Recall is 84.85%, F1-score is 80.52% and Accuracy is 89.37%.The highest accuracy for multi-class classification without optimizations is exhibited by Random Forest model.

5. Discussion

In this study, ML models are used in the planning, development, and deployment a mechanism for detecting intrusions. It is predicated on the phenomenon of supervised learning since tagged data are accessible for empirical investigation. The CICIDS2017 dataset is employed in empirical research. Despite being older and having more citations than other datasets, research on datasets revealed that this one outperforms many others in terms of benchmarking and support for various incursion types. Four tree-based techniques are selected for this study even though there are numerous machine learning models available. This is justified by the fact that the literature study provided in Section 2 indicated that the tree-based models performed better. The calibre of training data is important since the selected machine learning models rely on supervised learning. Because of this, two feature selection strategies, such as information gain and FCBF, are used to enhance the quality of training data. One of the optimization strategies applied in this study is feature selection. Hyperparameter optimization is another optimization method that makes use of the Bayesian optimization notion.

Three categories are intended for experiments. The binomial classification strategy is used to create the first category of trials using machine learning models. It does imply that test samples will be divided into two classes by each model, such as BENIGN and INTRUSION. The ML models used in the second set of trials employ a multi-class classification strategy without the previously indicated modifications. That does imply that test samples will be categorized by each model into many classifications, including Bot, BruteForce, DoS, Infiltration, PortScan, and WebAttack. The ML models are used in the third category of trials, which employs the multi-class classification strategy with the previously indicated optimizations. That does imply that test samples will be categorized by each model into many classifications, including Bot, BruteForce, DoS, Infiltration, PortScan, and WebAttack. Since it just gives the result for every test case, whether it is BENIGN or INTRUSION the first category is recommended. For many use instances, intrusion detection alone suffices and categorization is not necessary, therefore this far is helpful. In some additional use situations, the network administrator has to know the precise type of intrusion or assault. Multi-class categorization is helpful in these situations. The changes made in this study may improve performance, but not much, given the research gaps identified in the literature, such as the requirement for ML model improvement in terms of feature engineering and hyperparameter optimization.

6. Conclusion and Future Work

The goal of this project is to create and deploy intrusion detection systems based on machine learning models. Four machine learning models with two optimizations—feature selection and hyperparameter optimization—are used in the construction of the system. We proposed two algorithms named Hybrid Feature Selection (HFS) and Learning based Intrusion Detection (LbID). The system is built using two optimizations—feature selection and hyperparameter optimization—in four machine learning models. Optimizing ML models results in relatively little gain in accuracy. Stated differently, the application of enhancements yields no appreciable gain in accuracy.

There are a few possible causes for this, including limitations on the dataset. In the future, more investigation into this will be required. The binomial classification yields the greatest accuracy of 94.22% for the RF model. Decision Tree has the highest accuracy (91.67%) for multi-class classification in the absence of optimizations. With an accuracy of 93.46%, RF has the highest multi-class classification with optimizations. This study has several flaws that should be fixed in other research projects. This study's tests and observations are predicated only on an antiquated dataset. Working with more datasets in the future would be feasible if less-explored but more recent datasets are taken into consideration. Deep learning models in particular, which are based on neural networks, have the potential to enhance output. This creates an additional avenue for the investigation's potential future scope. Thirdly, the research endeavours rely on the accessible datasets. However, using actual network traffic as test data for intrusion detection is extremely desired. Test data may potentially be collected live and quickly from networks in order to do this.

References

[1] Lee, Chie-Hong; Su, Yann-Yean; Lin, Yu-Chun and Lee, Shie-Jue (2017). Machine learning based network intrusion detection, IEEE, pp.79–83. http://doi:10.1109/CIAPP.2017.8167184

[2] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., and Abuzneid, A. (2019). Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. Electronics,8(3), pp.1. http://doi:10.3390/electronics8030322

[3] Alhajjar, E., Maxwell, P., and Bastian, N. (2021). Adversarial machine learning in Network Intrusion Detection Systems. Expert Systems with Applications, 186, pp.1.13. http://doi:10.1016/j.eswa.2021.115782

[4] De Carvalho Bertoli, G., Pereira Junior, L. A., Saotome, O., Dos Santos, A. L., Verri, F. A. N., Marcondes, C. A. C. and Parente De Oliveira, J. M. (2021). An End-to-End Framework for Machine

Learning-Based Network Intrusion Detection System. IEEEAccess,9,pp.106790–106805. http://doi:10.1109/access.2021.3101188

[5] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. Pp.1-29. http://doi:10.1002/ett.4150

[6] Zaman, M., and Lung, C.-H. (2018). Evaluation of machine learning techniques for network intrusion detection. NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. Pp.1-5. http://doi:10.1109/noms.2018.8406212

[7] Sultana, N., Chilamkurti, N., Peng, W., and Alhadad, R. (2018). Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications. Pp.1-9. http://doi:10.1007/s12083-017-0630-0

[8] Li, J., Qu, Y., Chao, F., Shum, H. P. H., Ho, E. S. L., and Yang, L. (2018). Machine Learning Algorithms for Network Intrusion Detection. Intelligent Systems Reference Library, 151–179. http://doi:10.1007/978-3-319-98842-9_6

[9] Anshu Parashar, Kuljot Singh Saggu and Anupam Garg. (2022). Machine learning based framework for network intrusion detection system using stacking ensemble technique. Indian Journal of Engineering & Materials Sciences. 29, pp.509-518.

[10] Taher, K. A., Mohammed Yasin Jisan, B., & Rahman, M. M. (2019). Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection. 2019 International Conference on Robotics,Electrical and Signal Processing Techniques (ICREST). Pp.1-4. http://doi:10.1109/icrest.2019.8644161

[11] Magán-Carrión, R., Urda, D., Díaz-Cano, I., and Dorronsoro, B. (2020). Towards a Reliable Comparison and Evaluation of Network Intrusion Detection Systems Based on Machine Learning Approaches. Applied Sciences, 10(5), pp.1-21. http://doi:10.3390/app10051775

[12] A, Anish Halimaa and Sundarakantham, K. (2019). Machine Learning Based Intrusion Detection System. , IEEE, pp.916–920. http://doi:10.1109/ICOEI.2019.8862784

[13] Phadke, A., Kulkarni, M., Bhawalkar, P., and Bhattad, R. (2019). A Review of Machine Learning Methodologies for Network Intrusion Detection. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). Pp.1-4. http://doi:10.1109/iccmc.2019.8819748

[14] Dini, P. and Saponara, S. (2021). Analysis, Design, and Comparison of Machine-Learning Techniques for Networking Intrusion Detection. Designs, 5(1), pp.1-21. http://doi:10.3390/designs5010009

[15] Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2018). A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection. IEEE Communications Surveys & Tutorials, pp.1–46. http://doi:10.1109/comst.2018.2847722

[16] Da Costa, K. A. P., Papa, J. P., Lisboa, C. O., Munoz, R. and de Albuquerque, V. H. C. (2019). Internet of Things: A survey on machine learning-based intrusion detection approaches. Computer Networks, 151, pp.147–157. http://doi:10.1016/j.comnet.2019.01.023

[17] Seraphim, B. I., Palit, S., Srivastava, K. and Poovammal, E. (2018). A Survey on Machine Learning Techniques in Network Intrusion Detection System. 2018 4th International Conference on Computing Communication and Automation (ICCCA). Pp.1-5. http://doi:10.1109/ccaa.2018.8777596

[18] Satapathy, Suresh Chandra; Raju, K. Srujan; Shyamala, K.; Krishna, D. Rama and Favorskaya, Margarita N. (2020). A Review on Network Intrusion Detection System Using Machine Learning, ICETE, pp.598–607. http://doi:10.1007/978-3-030-24318-0_69

[19] Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs and Mouhammd Alkasassbeh. (2017). Evaluation of Machine Learning Algorithms for Intrusion Detection System. IEEE 15th International Symposium on Intelligent Systems and Informatics, pp.1-6.

[20] Liu, L., Wang, P., Lin, J., & Liu, L. (2021). Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning. IEEEAccess,9,pp.7550–7563. http://doi:10.1109/access.2020.3048198

[21] Wu, F., Li, T., Wu, Z., Wu, S., & Xiao, C. (2021). Research on Network Intrusion Detection Technology Based on Machine Learning. International Journal of Wireless Information Networks, 28(3), pp.262–275. http://doi:10.1007/s10776-021-00520-z

[22] Achmad Akbar Megantara and Tohari Ahmad. (2021). A hybrid machine learning method for increasing the performance of network intrusion detection systems. Megantara and Ahmad J Big Data, pp.1-19.

[23] Chang, Y., Li, W., & Yang, Z. (2017). Network Intrusion Detection Based on Random Forest and Support Vector Machine. 22017 IEEE International Conference on Computational Science and

Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC). Pp.1-4. http://doi:10.1109/cse-euc.2017.118

[24] Abubakar, A., & Pranggono, B. (2017). Machine learning based intrusion detection system for software defined networks. 2017 Seventh International Conference on Emerging Security Technologies (EST). pp.1-6. http://doi:10.1109/est.2017.8090413

[25] Farrukh Aslam Khan and Abdu Gumaei. (2019). A Comparative Study of Machine Learning Classifiers for Network Intrusion Detection. Springer, p.75–86.

[26] Thomas Rincy N 1 and Roopam Gupta. (2021). Design and Development of an Efficient Network Intrusion Detection System Using Machine Learning Techniques. Hindawi Wireless Communications and Mobile Computing, pp.1-35.

[27] Jing Li, Mohd Shahizan Othman, Hewan Chen and Lizawati Mi Yusuf. (2024). Optimizing IoT intrusion detection system feature selection versus feature extraction in machine learning. Journal of Big Data, pp.1-44.

[28] Ameera S. Jaradat, Malek M. Barhoush and Rawan Bani Easa. (2022). Network intrusion detection system: machine learning approach. Indonesian Journal of Electrical Engineering and Computer Science. 25(2), p.1151~1158.

[29] Zhihui Fan and Zhixuan You. (2024). Research on network intrusion detection based on XGBoost algorithm and multiple machine learning algorithms. Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics, pp.162-167.

[30] Chunying Zhang, Wenjie Wang, Lu Liu, Jing Ren 1 and Liya Wang. (2022). Three-Branch Random Forest Intrusion Detection Model. MDPI, pp.1-21.

[31] Md. Alamin Talukder, Md. Manowarul Islam, Md Ashraf Uddin and Khondokar Fid. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embe. Journal of Big Data, pp.1-44.

[32] Dhaliwal, S., Nahid, A.-A., and Abbas, R. (2018). Effective Intrusion Detection System Using XGBoost. Information, 9(7), pp.1-24. doi:10.3390/info9070149

[33] Chunying Zhang, Wenjie Wang, Lu Liu, Jing Ren and Liya Wang. (2022). Three-Branch Random Forest Intrusion Detection Model. MDPI, pp.1-21.

[34] Witcha Chimphlee and Siriporn Chimphlee. (2023). INTRUSION DETECTION SYSTEM (IDS) DEVELOPMENT USING TREE- BASED MACHINE LEARNING ALGORITHMS. International Journal of Computer Networks & Communications (IJCNC). 15(4), pp.93-109.

[35] Intrusion detection evaluation dataset (CIC-IDS2017). Retrieved from https://www.unb.ca/cic/datasets/ids-2017.html

[36] Maldorad S and Weber R (2009) A wrapper method for feature selection using support vector. machines. Information Sciences 179:2208–2217.

[37] Rice JA (2006) Mathematical Statistics and Data Analysis. Third Edition.

[38] Kullback, S. and Leibler, R.A. (1951). "On Information and Sufficiency". Annals of Mathematical Statistics 22 (1): 79–86.