National College of Ireland

Enhancing Efficiency of Machine Learning Techniques with Feature Selection and Hyperparameter Tuning for Intrusion Detection towards Leveraging Cybersecurity

MSc Research Project
MSc in Cyber Security

Abdur Razzaq Shaik
Student ID: X22178333

School of Computing
National College of Ireland

Supervisor: Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. ……Abdur Razzaq Shaik…
……

Student ID: ……X22178333
……

Programme: ……MSc in Cyber Security **Year:** ……2024
…… ……..

Module: ……MSc Research Project
……

Supervisor: ……Arghir Nicolae Moldovan
……

Submission Due Date: ……06-03-2024
……

Project Title: ……Enhancing Efficiency of machine Learning Techniques with Feature Selection Techniques and Hyper Parameter Tuning For Intrusion Detection towards Leveraging Cyber Security
……

……7582……
Word Count: **Page Count**……22……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Abdur Razzaq Shaik
……

Date: ……05-03-2024
……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Efficiency of Machine Learning Techniques with Feature Selection and Hyperparameter Tuning for Intrusion Detection towards Leveraging Cybersecurity

Abdur Razzaq Shaik
X22178333

Abstract

Cybersecurity enhancement is a continuous process due to increasing cyber-attacks of late. Traditional security mechanisms were based on certain heuristics that could detect intrusions based on their detection process. However, with the emergence of Artificial Intelligence (AI) methods such as machine learning (ML) approaches, learningbased models are found efficient due to their ability to learn from labelled data continuously. It is found in the literature that ML models based on supervised learning show deteriorated intrusion detection performance when training samples are not with designed quantity and quality. Therefore, it is important to leverage performance of ML models with certain optimizations. This is the motivation behind this research which is aimed at building an intrusion detection system based on ML models with feature engineering and hyperparameter tuning optimizations. The system is evaluated using CICIDS2017 dataset. Intrusion detection system is implemented using binomial classification and also multi-class classification. In the binomial classification highest accuracy is achieved by RF model with 99.87%. In case of multi-class classification without optimizations, highest accuracy is exhibited by RF with 99.44%. In case of multi-class classification with optimizations, highest accuracy is exhibited by XGBoost with 99.64%.

1 Introduction

Cybersecurity in the contemporary era has become very challenging (Razan et al., 2018). Because traditional security is ineffective against sophisticated assaults, machine learning is essential (Kamran et al., 2020). With machine learning, it is possible to have learning based incremental knowledge for intrusion detection systems to detect known an unknown attacks. In the context of growing cyber-attacks, it is indispensable to continue research on enhancing cybersecurity. Since security is not one-time effort, it needs continuous improvement of the systems and intrusion detection mechanisms. This is the motivating factor for the current research which is significant for the insights it is expected to give on suitable machine learning (ML) models for intrusion detection, optimization techniques and training datasets. Aim of this research is to build a machine learning framework with optimizations for efficient intrusion detection. The following are the research questions.

RQ1: How Can machine learning models be used for realizing an intrusion detection system?

RQ2: How Can optimizations like feature selection and hyperparameter optimization have impact on intrusion detection performance of ML models?

These research questions are based on the literature findings. The first questions probe into the capabilities of ML models for intrusion detection while the second question investigates on the two optimizations for intrusion detection performance improvement of ML models. Towards this end, the following are the research objectives.

- To investigate on the existing ML methods and related works used to realize intrusion detection systems.
- To propose a ML based framework with its mechanisms and optimizations for efficient intrusion detection.
- To explore feature selection and hyperparameter optimization for improving performance of ML models in intrusion detection.
- To evaluate the proposed framework.
- To draw conclusions and provide recommendations for future work.

To fulfil these objectives, research is carried out with empirical study. Here are the main contributions of this research.

- First, a ML based framework is designed and implemented for intrusion detection.
- Second, different ML models are evaluated for intrusion detection with binomial classification, multi-class classification with and without optimizations such as feature engineering and hyperparameter tuning. The remainder of this report is structured as follows. Section 2 reviews literature on various existing methods used for intrusion detection.
- Section 3 presents our research methodology.
- Section 4 presents design specification.
- Section5 and Section 6 provide details of implementation and evaluation.
- Section 6 also provides discussion of the significance of the research. Section provides conclusions drawn and future scope of the research.

2 Related Work

This section reviews literature on different existing methods used for intrusion detection. Dilara et al. (2020) examined the application of machine learning, particularly deep learning, to cyber security intrusion detection, evaluating approaches, results, and benchmark datasets to ensure impartial assessment. Hongyu and Bo (2019) with IDSs keeping an eye on networks, cyber security is essential. Accuracy is increased and unknown assaults are detected using ML. The future of deep learning seems bright. Guojun et al. (2019) the IDSs are essential for thwarting assaults, therefore cyber security is key. KNN performs better than the other seven machine learning algorithms when tested on the CICIDS2017 dataset. Iqbal H et al. (2020) as IoT grows, cyber security becomes more important. Prioritizing characteristics for efficient intrusion detection, lowering complexity, and contrasting with conventional techniques are all part of the "IntruDTree" paradigm. Kilincer et al. (2021)

issued arise with increased internet use. IDS recognizes malicious activity. Examining research on SVM, KNN, and DT datasets: CSE-CIC, UNSW-NB15, ISCX-2012, NSL-KD and CIDDS-001. Next: several assaults for DL/ML assessment, along with a new dataset featuring vLAN network architecture.

Razan et al. (2018) unbalanced datasets require creative solutions in light of growing cyber security concerns. The CIDDS-001 study assesses DNN, Random Forest, Voting, VAE, and stacking. Ustun et al. (2021) for smart grids without cyber security, IEC 61850 connectivity is crucial. The intrusion detection system for GOOSE messages developed in this study works well. Ferrag et al. (2020) examined the deep learning techniques for intrusion detection, testing seven models using datasets from Bot-IoT and CSE-CIC-IDS2018. Dini et al. (2023) highlighted the potential for future integration of deep learning by comparing performance across three datasets. Strong performers in intrusion detection include machine learning models, particularly Decision Trees and Random Forests. Vigneswaran et al. (2018) because of the growing threat of cyber-attacks, DNNs are essential to current IDS. Using KDDCup-'99' dataset, a 3-layer DNN beats standard ML techniques.

Liu et al. (2021) recognized covert cyber-attacks is difficult in unbalanced networks. In the CSE-CIC-IDS2018 and NSL-KDD datasets, the DSSTE method improves classification by reducing imbalance. Kamran et al. (2020) analysed machine learning methods for identifying online threats such as malware, spam, fraud, and intrusion. It talks about dataset comparisons, issues with different metrics, and the temporal complexity of ML models. Upcoming research aims to strengthen machine learning models for cyber security, particularly in the face of hostile inputs. Leslie F (2019) an important danger is network infiltration. Systems that use machine learning are able to identify and reduce harmful traffic. In this chapter, KDD 99 is used to assess AI systems, and future cyber security issues are covered. Maseer et al. (2021) analyses the efficacy of IDS, emphasizing the detection capabilities of online assaults of the k-NN-AIDS, DT-AIDS, and NB-AIDS models. Costa et al. (2019) explored IoT security via Machine Learning Techniques and Intrusion Detection, looking at more than 95 papers in the field. Using contemporary research to improve data security, it draws attention to the difficulties in IoT intrusion detection. With an emphasis on resolving false positives, the research investigates clever strategies to increase intrusion detection precision.

Kamran et al. (2020) expanded of the Internet has led to a rise in cyber risks. Because traditional security is ineffective against sophisticated assaults, machine learning is essential. In this study, ML strategies for cyber security concerns are reviewed, and the significance of datasets, ML tools, and assessment metrics are discussed. Difficulties include limited dataset availability, evasive attacks, and the requirement for reliable machine learning. Iram et al. (2020) Increased Internet usage has led to a rise in cyber-attacks on network security. You need robust IDSs. Over 99% accuracy on NSL-KDD is demonstrated by ML classifiers such as SVM, KNN, and RF, which are useful for intrusion detection. It may be possible to enhance accuracy in the future by investigating ensemble approaches and refining IDS. Al-Al-Omari et al. (2021) suggested a Decision Tree-based model that takes feature ranking into account for increased precision and decreased complexity. The model's effectiveness in comparison to conventional approaches is demonstrated by its performance measures, which include accuracy, precision, recall, and F-score. Future research will focus on improving feature selection with integrated techniques and anticipating cyber-attacks. Hernandez-Jaimes

et al. (2023) presented security risks, inspiring the development of novel intrusion detection techniques. In addition to analysing datasets and addressing cyber security concerns, this survey classifies IoMT intrusion detection techniques. ML/DL techniques that focus on network traffic monitoring for IoMT devices with limited resources are noteworthy for their ability to identify suspicious activity. Not many works combine mitigation with SDN and NFV. Keshk et al. (2023) with the use of the SPIP framework and the LSTM model, an IDS with excellent accuracy and interpretability was created for IoT networks. To ensure a thorough understanding of attack behaviour, the SPIP architecture integrates DL and XAI techniques. While it matches attack characteristics and works better than the original features, vulnerability identification might be the main focus of future study.

Nasrin et al. (2018) with the use of ML/DL algorithms in NIDS, SDN provides effective network security monitoring. With regard to SDN-NIDS, DL is examined using ML tool coverage, emphasizing obstacles and potential paths forward. Manickam and Rajagopalan (2018) by managing risks and maximizing productivity, cloud computing provides effective infrastructure usage. Rule-based pattern matching is the method used by current IDS systems to protect networks. In this study, a multi-layer ANN optimized for cloud-based IDS with improved performance using GSO-TS is presented. Reducing convergence time and problems with local optima, the suggested GSO-TS optimizes the ANN structure. Comparing the results to conventional ANN models, higher detection rates were found. Anthi et al. (2019) presented a three-layer intrusion detection system (IDS) for Internet of Things networks that can identify different types of cyber-attacks. It categorizes threats, recognizes malicious packets, and creates device behaviour profiles. Tested on a smart home and received F-measures for its functionalities Liu et al. (2021) improved the explain ability of intrusion detection alarms, the FAIXID system combines XAI with data cleansing. Modules for attribution, assessment, post-modelling, modelling, and pre-modelling are all included. Analysis shows that decision-making for analysts has improved. Upcoming projects will compare data cleaning methods and automate the attribution module. Marek et al. (2020) focused on protecting machine learning-based cyber-attack detectors from hostile assaults. The paper suggests a detection strategy based on brain activations and assesses how well intrusion detection algorithms function against four assault techniques. The encouraging recall rates in the results point to the possibility of enhanced intrusion detection system protection. Reducing false positives, however, is essential for further advancement.

Ferrag et al. (2019) offered an overview and comparison of deep learning techniques for cyber security intrusion detection. It focuses on accuracy, false alarm rate, and detection rate while analysing seven models on two actual datasets. Kabir and hartmann (2018) with the surge in hostile network activity, effective intrusion detection is essential. IDSs identify security lapses like as buffer overflows, malware, and denial-of-service attacks. By using effective indexing techniques, our Snort-based NIDS improves real-time detection while lowering false positives. Kocher and Kumar (2021) examined the growing application of deep learning (DL) in conjunction with conventional machine learning (ML) techniques, especially in intrusion detection. It contrasts ML techniques—such as ANN, SVM, and others—with DL techniques, emphasizing current studies and issues in this area. Tama and Lim (2021) analysed 124 papers as part of a methodical mapping research on the application of ensemble learners in intrusion detection systems (IDSs). It looks at patterns, popular

techniques like random forests, and how well they work in comparison to standalone classifiers. Mohan and Kumar (2020) for transportation systems, IoT-connected cars must prioritize security, privacy, and trust. Protecting wireless networks requires intrusion detection. This study discusses future research fields and advises safeguarding Intelligent Transportation Systems utilizing Blockchain (PChain) to overcome IoT vulnerabilities.

Osama et al. (2020) explored how intrusion detection, block chain applications, and cloud systems interact to thwart cyber-attacks. In addition to highlighting the need of virtualization, containerization, and block chain for safe intrusion detection in the cloud, it describes cooperative anomaly detection for insider and outer assaults in cloud centres. The review highlights problems and future research directions and covers cloud architecture, security event categorization, cloud-based NIDS, and collaborative NIDSs for block chain applications. Leonel et al. (2018) examined research on IoT intrusion detection systems (IDS) from 2009 to 2017 and emphasizes the necessity for specialized IoT security solutions. IDS is suggested as an IoT solution as it is essential in conventional networks. The analysis of twenty studies shows that the field of IoT intrusion detection systems is still in its infancy, with little agreement on deployment tactics and detection techniques. Subsequent investigations have to concentrate on reaching a consensus in these domains, broadening the scope of threat detection, and tackling diverse IoT technologies. Macas and Wu (2020) with the increase in cyberattacks, cybersecurity is becoming more and more important. Threat analysis and identification have a great deal of potential thanks to artificial intelligence (AI) and machine learning (ML), especially deep learning (DL). In order to detect network intrusions, this research investigates deep learning approaches and shows encouraging outcomes. Perez et al. (2021) combined multiplex networks with time series analysis, a novel approach to developing intrusion detection systems (IDS) lowers warnings and computational burden for efficient corporate intrusion detection. It improves detection by representing IP linkages in a multiplex network. Future research will focus on forecasting network activity, employing supervised clustering approaches, and validating in corporate environments. Pooja and Purohit (2021) investigated the use of Bi-directional LSTM in an Intrusion Detection System (IDS), attaining better accuracy on the KDDCUP-99 and UNSW-NB15 datasets. This shows how deep learning has the ability to enhance IDS over state-of-the-art techniques. Online deployment and GPU testing are tasks for the future.

Ashiku and Dagli (2021) suggested by utilizing adaptive network intrusion detection systems (IDS) with deep learning to counteract emerging cyber threats. The model's accuracy on the UNSW-NB15 dataset was 95.4% and 95.6%, respectively, indicating room for improvement in feature reduction and transfer learning. Upcoming projects will improve multiclass classification and mitigate zero-day attacks. Azam et al. (2020) expanded M2M services and resents a new M2M communication architecture with upcoming wireless tech. It suggests a strategy for choosing M2M gateways and shows how the SD approach improves availability and energy efficiency. BT v4.2 offers safe connections using IPv6/6LoWPAN, among other security features. Yang et al. (2022) addressed the cyber-attack vulnerabilities of the external and intra-vehicle networks of contemporary cars and suggests a multi-tiered hybrid intrusion detection system. The system obtains F1-scores of 0.800 and 0.963 for zero-day attacks, respectively, and demonstrates great accuracy in identifying known assaults. Effectiveness and viability for real-time deployment in automotive systems are demonstrated by the

suggested IDS. Traditional machine learning stages, supervised learners for known attack detection, unsupervised models for zero-day attacks, and optimization strategies for increased accuracy are all included in the multi-tiered hybrid intrusion detection system (IDS) model. Ramesh et al. (2024) examined the function of datasets, IDS, and its techniques. This study investigates how network security might be improved by integrating machine learning with intrusion detection systems. Through the evaluation of performance parameters such as accuracy, precision, recall, F1 score, and recall over 10 samples, valuable insights into cyber-attack identification are obtained. By highlighting asset protection for businesses, this research helps cyber security experts upgrade IDS for resilience against changing threats. Sarhan et al. (2022) used a variety of Feature Reduction (FR) and Machine Learning (ML) approaches, researchers hope to enhance NIDSs. Using three FE methods and six machine learning models against the benchmark datasets UNSW-NB15, ToN-IoT and CSE-CIC-IDS2018, this study concludes that no one approach outperforms the others on all of the datasets. Noted are optimal dimensions and reduced performance with LDA, highlighting the necessity for a benchmark feature set to promote NIDS research.

2.1 Literature Review Findings

From the review of literature, it was understood that tree-based models could perform well in intrusion detection research. Different ML models identified suitable for this research include Decision Tree (Dini et al., 202; Al-Omari et al., 2021), Random Forest (Razan et al., 2018; Dini et al., 2023), Extra Trees (Razan et al., 2018) and XGBoost (Iram et al., 2020). In this research, these models are preferred for the empirical study. All these are tree based supervised learning models. These models perform well when there is quality data used for training. If not, they exhibit deteriorated performance. To overcome this problem, from the literature review, it is observed that feature selection (Al-Omari et al., 2021) and hyperparameter tuning (Yang et al., 2022) could be investigated for leveraging performance of ML models.

3 Research Methodology

Aim of this research is to build a machine learning framework with optimizations for efficient intrusion detection. Towards achieving the aim of the research, the research methodology followed is illustrated in Figure 1. Methodology includes various tasks to be done in a logical order to arrive at the research outcomes expected.

3.1 Literature Review

It starts with a literature review which provides a strong basis for the research. Literature review insights help in understanding the ML models to be used for intrusion detection research, the datasets, limitations of ML models and the possible optimization approaches that could leverage performance of ML models for intrusion detection. Literature review also provides specific research gaps that could trigger further investigation into developing intrusion detection system as described in Section 2.

3.2 Data Collection

Since the proposed intrusion detection system is based on supervised learning process, dataset with labelled data plays crucial role in this research. In the process of data collection process, the following criteria are employed.

1. Dataset should be designed for intrusion detection research.
2. Dataset should have sufficient samples for training
3. Dataset should be relatively new with less citations
4. Dataset should have been used by other researchers

The criteria provided above were used to choose dataset that helps in improving quality in investigations.

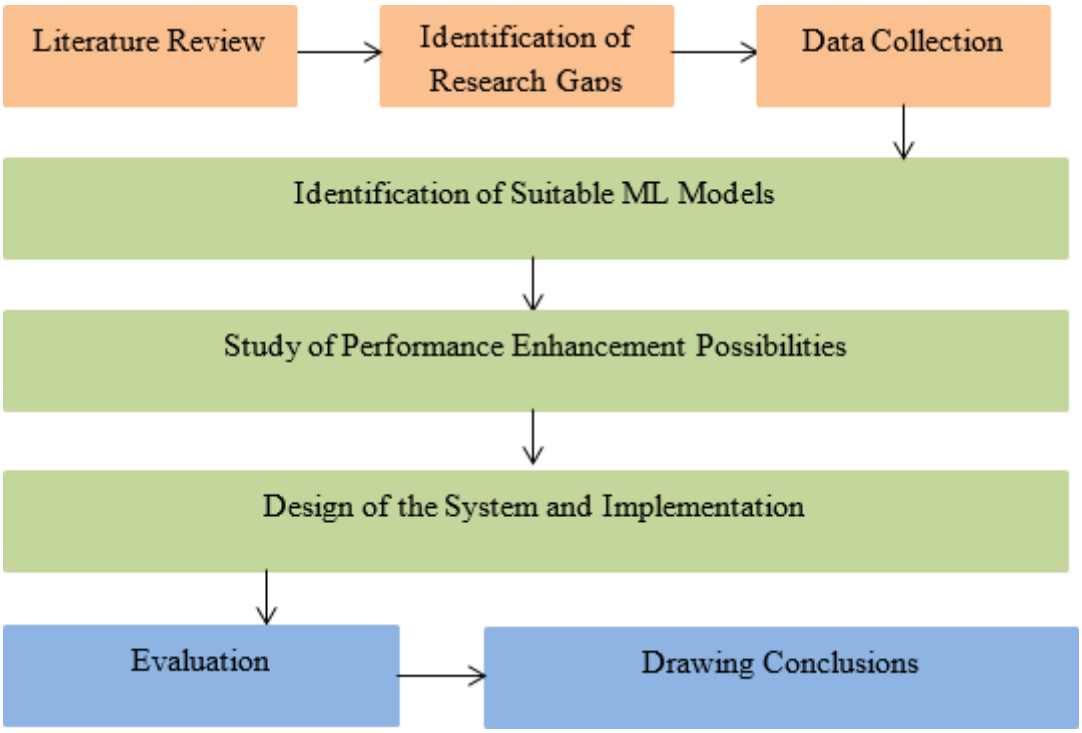

Figure 1: Overview of the research methodology

3.3 Identification of Suitable Machine Learning Models

With the help of literature review and study of different ML tutorials, suitable models that are efficient and widely used are identified. It is also understood that supervised learning models are suitable for this research. The rationale behind this is that, there is sufficient labelled data that can be used for intrusion detection research.

3.4 Study of Performance Enhancement Possibilities

Since the proposed intrusion detection system is based on supervised ML, there are number of existing models like RF. However, using such models directly might result in mediocre performance. Therefore, investigation is made through literature review to know the possible means of improving prediction performance of ML models.

3.5 Design of the System and its Implementation

Based on the observations in Section 3.2, 3.3 and 3.4, an intrusion system is designed as presented in Section 4 and implementation details are provided in Section 5.

3.6 Evaluation Methodology

Performance of the proposed intrusion detection system is evaluated using different metrics widely used in the literature. Those metrics are known as precision, recall, F1-score and accuracy. They are derived based on the confusion matrix illustrated in Figure 2.

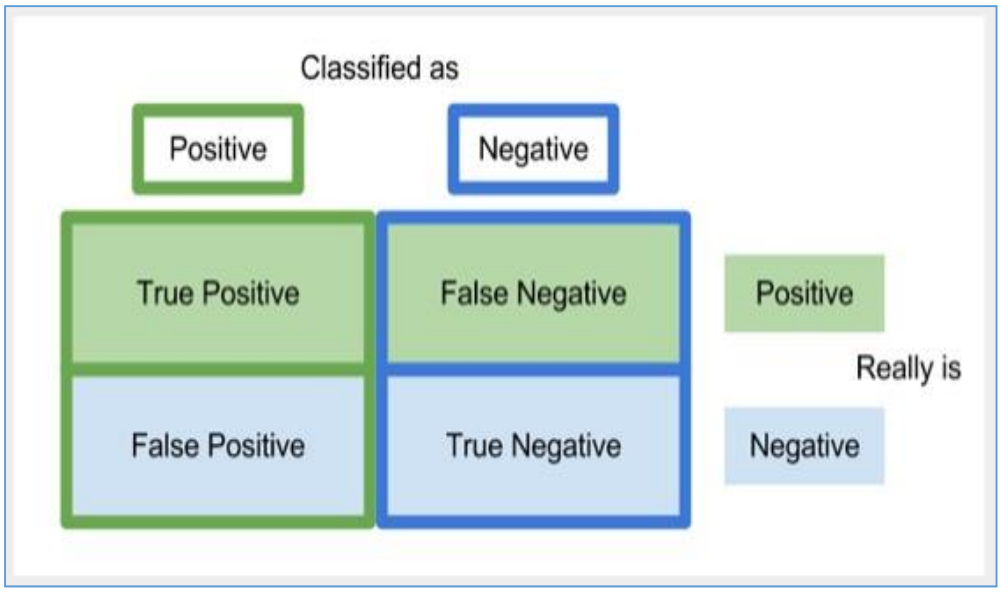

Figure 2: Confusion matrix

If the given test sample has INTRUSION and the algorithm prediction is also INTRUSION, this case is known as True Positive (TP). If the given test sample is BENIGN and the algorithm prediction is also BENIGN, this case is known as True Negative (TN). If the given test sample is BENIGN and the algorithm prediction is INTRUSION, this case is known as False Positive (FP). If the given test sample has INTRUSION and the algorithm prediction is BENIGN, this case is known as False Negative (FN).

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)

Based on the four cases shown in confusion matrix, Eq. 1, Eq. 2, Eq. 3 and Eq. 4 express the computation of precision, recall, F1-score and accuracy respectively.

4 Design Specification

This section provides design specification which is based on the research methodology described in Section 3. Data collection process as described in Section 3.2 is followed. Initially different datasets are found and studied. The datasets on which the selection criteria applied are as follows.

1. NSL-KDD
2. IEC 60870-5
3. CIC Bell DNS 2021
4. UNSW NB15
5. CICIDS2017
6. MQTT

These datasets are studied and criteria provided in Section 3.2 applied. Out of them CICIDS2017 is found to have better benchmarking and relatively new with different kinds of intrusions. Therefore, CICIDS2017[1] is the dataset used for empirical study. It is also found widely used in the literature such as (Guojun et al., 2019; Maseer et al., 2021). Different ML models identified suitable for this research are as follows.

1. Decision Tree (Dini et al., 202; Al-Omari et al., 2021)
2. Random Forest (Razan et al., 2018; Dini et al., 2023)
3. Extra Trees (Razan et al., 2018)
4. XGBoost (Iram et al., 2020)

All these are tree based supervised learning models. These models perform well when there is quality data used for training. If not, they exhibit deteriorated performance. To overcome this problem, from the literature review, it is observed that feature selection (Al-Omari et al., 2021) and hyperparameter tuning (Yang et al., 2022) could be investigated for leveraging performance of ML models. Based on these findings as per the proposed methodology, the proposed intrusion detection system's architecture is as shown in Figure 3.

[1] https://www.unb.ca/cic/datasets/ids-2017.html

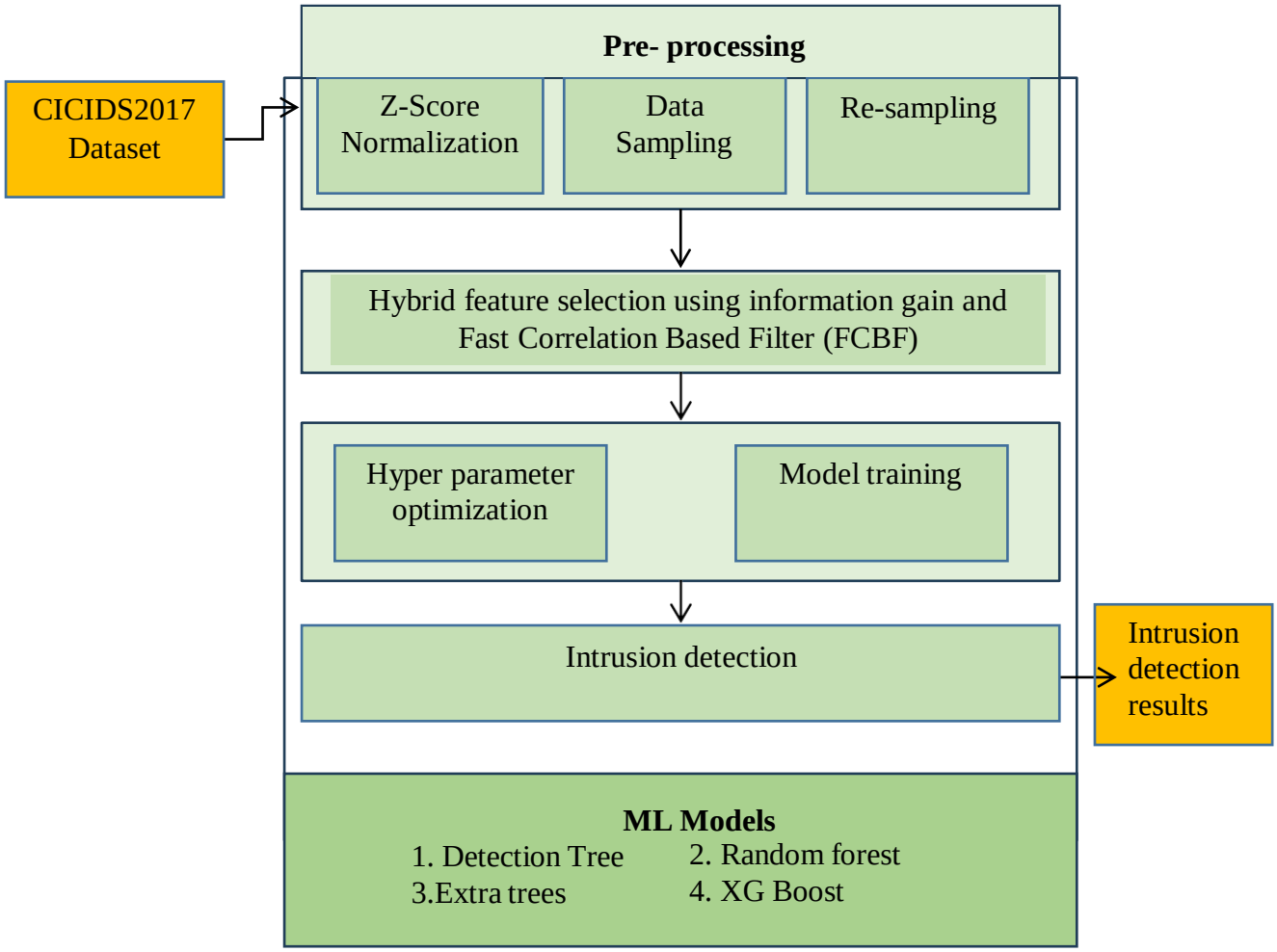

Figure 3: Architecture of the proposed intrusion detection system

The functionality of the proposed system is described her. The given dataset is subjected to pre-processing that includes Z-score normalization, filling empty values with zero, division of data into training and testing data besides re-sampling of data to get rid of overfitting. SMOTE is the tool used to solve the problem of class imbalance. Once pre-processing is completed, the data is subjected to feature engineering for choosing best performing features. Afterwards, each ML model used in the empirical study is subjected to hyperparameter tuning. This process is useful in setting optimal values for hyperparameter of given ML model for improving its performance. Then the models are trained using training data. once training is completed, the trained model is persisted for future reuse. The learned model is used for detecting intrusions in the test data. It results in intrusion detection results.

4.1 Feature Engineering

Feature selection is the process in which importance of each feature in class label prediction is computed and all features that have higher importance are chosen for training ML classifiers. Figure 4 shows generic filter-based approach for feature selection.

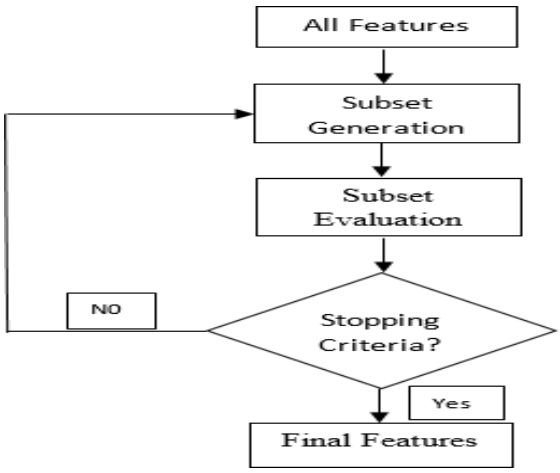

Figure 4: Illustrates feature selection process

It is known as filter-based approach that makes use of a measure for computing feature importance and based on feature importance threshold best contributing feature are elected. In this research, two metrics are used for feature selection. They are known as Information Gain (IG) and Fast Correlation Based Filter (FCBF).

4.2 Hyperparameter Optimization

Hyperparameter optimization of ML models help in tuning parameter values appropriately. This is achieved with the help of Bayesian optimization. Different parameters optimized for ML models is provided in Table 1.

Table 1: Hyperparameters of models subjected to optimization

ML Model	**Parameters Optimized**
XGBoost	• Learning rate • Max depth • Number of estimators
RF	• Criterion • Min samples leaf • Min samples split • Max features • Max depth • Number of estimators
Decision Tree	• Criterion • Min samples leaf • Min samples split • Max features • Max depth
Extra Trees	• Criterion • Min samples leaf • Min samples split • Max features • Max depth • Number of estimators

4.3 Machine Learning Techniques

In the proposed system, as shown in Figure 3, four ML models are used in the intrusion detection process. Each model is independent of other models.

Decision Tree

Among the most effective supervised learning techniques for classification and regression applications is the decision tree. It creates a tree structure that resembles a flowchart, with each internal node signifying an attribute test, each branch designating a test result, and each leaf node containing a class name. When a stopping criterion—such as the maximum depth of the tree or the minimum number of samples needed to split a node—is satisfied, the training data is recursively split into subsets depending on the values of the attributes. Based on a metric like entropy or Gini impurity, which gauges the degree of impurity or unpredictability in the subsets, the Decision Tree algorithm determines during training which attribute to split the data into. Finding the characteristic that optimizes the reduction in impurity or the information gain following the split is the aim.

Random Forest

A potent machine learning tree learning method is the Random Forest algorithm. It functions by building many Decision Trees in the training stage. A random subset of the data set is used to measure a random subset of characteristics in each partition throughout the construction of each tree. Because of the variety that this randomness brings to the individual trees, the likelihood of overfitting is decreased and the overall prediction performance is enhanced. In order to make predictions, the algorithm averages or votes over each tree's findings. With the help of several trees and their insights, this cooperative decision-making process yields findings that are accurate and consistent. For classification and regression tasks, random forests are frequently utilized because of their reputation for managing complicated data, minimizing overfitting, and producing accurate predictions.

Extra Trees

The additional trees method generates a large number of decision trees, much like the random forests technique, except it does so randomly and without replacement for each tree. In doing so, a dataset containing distinct samples is produced for every tree. Each tree additionally has a certain amount of randomly chosen features from the whole feature set. The ability of additional trees to randomly choose a splitting value for a feature is its most significant and distinctive feature. When splitting the data, the method chooses a split value at random rather than utilizing Gini or entropy to calculate a locally optimal value. The trees become uncorrelated and diverse as a result.

XGBoost

Machine learning models may be trained effectively and scalable with the help of XGBoost, an optimized distributed gradient boosting toolkit. By combining the predictions of several weak models, this ensemble learning technique generates a stronger prediction. Because of its

capacity to manage sizable datasets and attain cutting-edge results in numerous machine learning tasks, including regression and classification, XGBoost, which stands for "Extreme Gradient Boosting," has emerged as one of the most well-liked and extensively applied machine learning algorithms. XGBoost's ability to handle missing values efficiently makes it a valuable tool for managing real-world data with missing values. This capability eliminates the need for extensive pre-processing. Furthermore, XGBoost comes with built-in support for parallel processing, which enables training models on big datasets quickly.

4.4 Two Intrusion Detection Approaches

Experiments are designed in such a way that there are two intrusion detection approaches. The first approach is known as binomial classification of test samples into BENIGN and INTRUSION. The second approach is known as multi-class classification. Different classes reflect different kind of intrusion as shown in Table 2.

Table 2: Classes in multi-class classification

Intrusion Class	**Class Index**
BENIGN	0
Bot	1
BruteForce	2
DoS	3
Infiltration	4
PortScan	5
WebAttack	6

The ML models used in this research are used for both binomial classification and also multi-class classification. Section 5 presents implementation details.

5 Implementation

The proposed system meant for detecting intrusions is implemented using Python programming language. Different ML related libraries are used for implementation as shown in Table 3.

Table 3: Libraries used for implementation

Library / Module	Description
Numpy	General purpose package used to manipulate arrays.
Pandas	For different data structures and dealing with datasets
Seaborn	For visualization
Matplotlib	For interactive visualizations
Preprocessing	For various pre-processing functions
Model selection	Used to work with supervised learning
Metrics	Used to use performance metrics
Ensemble	To use ensemble based models
Tree	To use tree based models
Xgboost	To use XGBoost model
Feature selection	To use feature selection techniques
FCBF	Supports FCBF feature selection technique

For class imbalance rectification, SMOTE tools used in the implementation process. Data sampling is done on majority of classes and minority classes are retained. Data samples are clustered using K-Means clustering technique. The outputs produced in the implementation include confusion matrix for each model and for each testing approach such as binomial and multi-class.

The Following are the sources used as a reference while implementing the code:

1. https://github.com/vicky60629/Network-Intrusion-Detection-System.git
2. https://github.com/Namratha2301/IntrusionDetection.git
3. https://github.com/BinitDOX/Wireless-Intrusion-Detection-System.git

The contributions in this research includes Feature Selection using Information Gain and FCFB, Hyper Parameter optimization, Binomial and Multinomial classification.

6 Evaluation

The proposed intrusion detection system is evaluated to know performance of different ML models with and without optimizations. The evaluation also includes analysis of results with binomial classification and multi-class classification. The outcomes of the research are presented here. The results are presented in terms of binomial classification, multi-class classification without optimization and multi-class classification with optimization.

6.1 Results of Binomial Classification

Results of binomial classification are provided in terms of confusion matrix- based statistics and performance statistics.

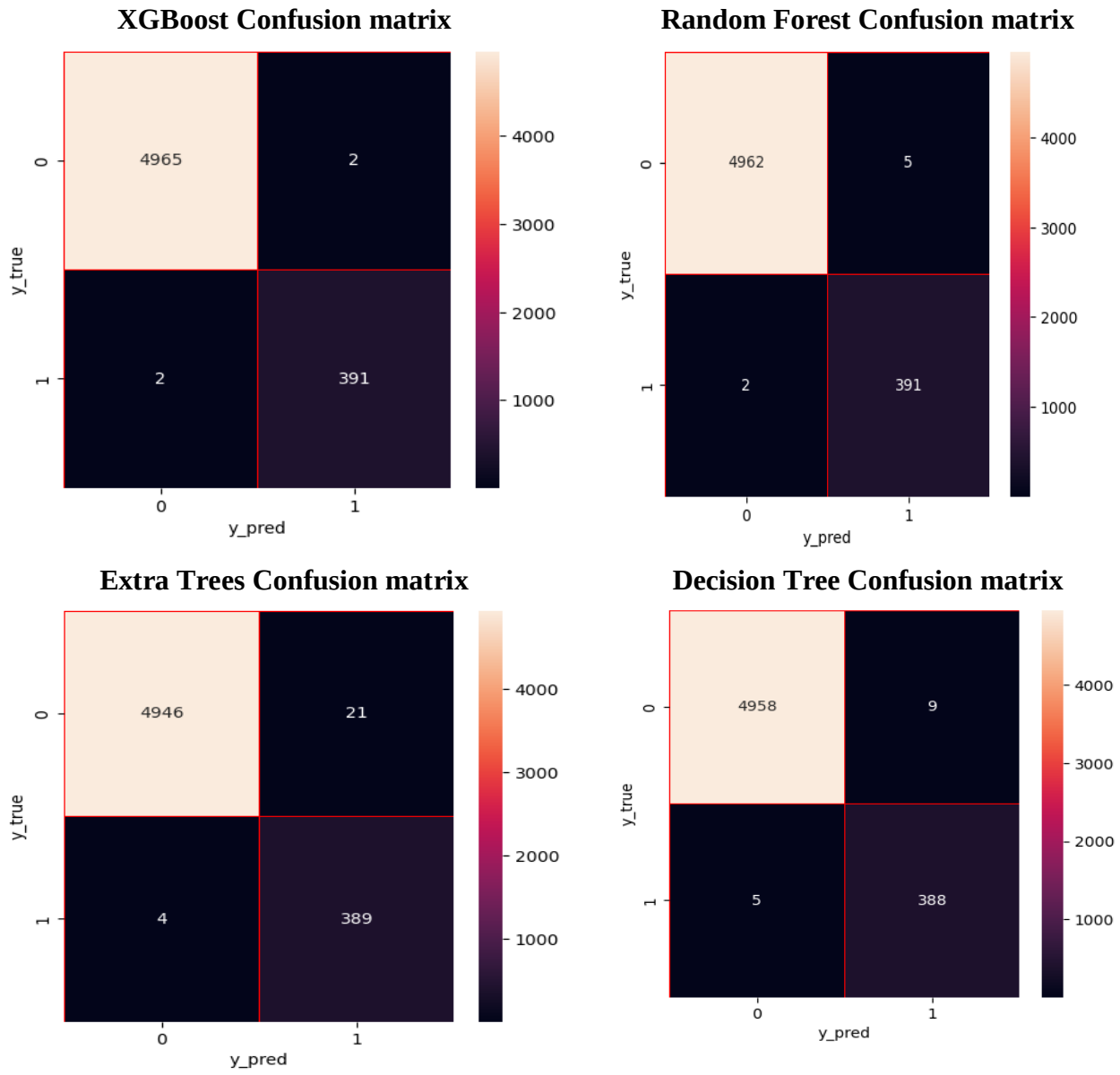

Figure 5: Results of binary classification in terms of confusion matrix

As presented in Figure 5, the predicted labels are compared with ground truth labels for all the four ML models. Table 4 shows the computed performance statistics based on confusion matrix.

Table 4: Intrusion detection performance of models with binomial classification

Binomial Classification Performance				
Intrusion Detection Model	**Precision**	**Recall**	**F1-score**	**Accuracy**
XGBoost	0.9949	0.9949	0.9949	0.9922
DecisionTree	0.9773	0.9872	0.9823	0.9974
Random Forest	0.9874	0.9949	0.9911	0.9987
ExtraTrees	0.9487	0.9898	0.9687	0.9953

As presented in Table 4, each detection model showed performance in terms of accuracy and other measures. It was observed that all models could achieve more than 99% accuracy in binomial classification.

6.2 Results of Multi-Class Classification Without Optimization

Results of multi-class classification without optimization are provided in terms of confusion matrix-based statistics and performance statistics.

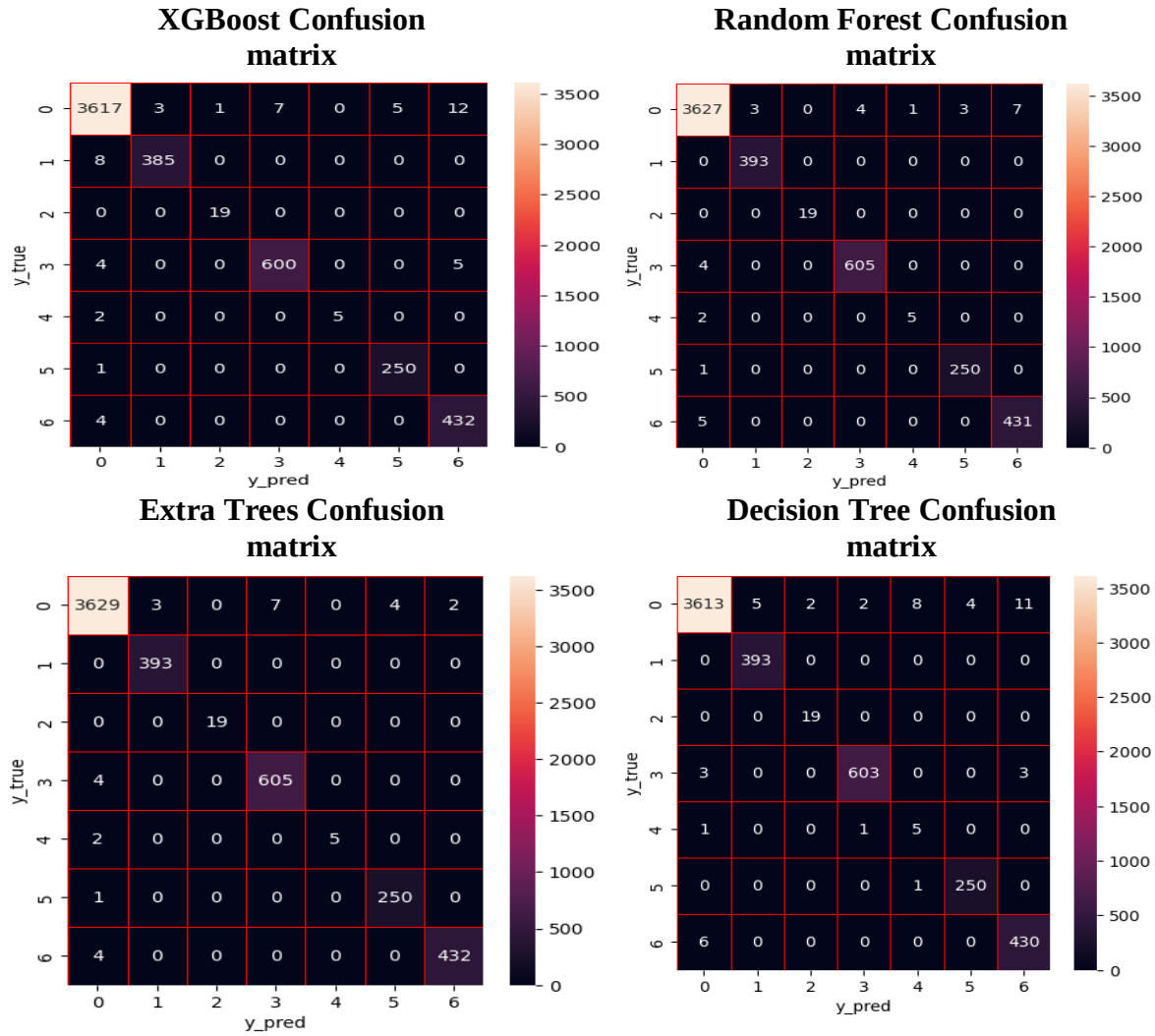

Figure 6: Results of multi-class classification without optimization in terms of confusion matrix

As presented in Figure 6, the predicted labels are compared with ground truth labels for all the four ML models and for all classes. Table 5 shows the computed performance statistics based on confusion matrix.

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**
XGBoost	0.9901	0.9901	0.9901	0.99
Random Forest	0.9943	0.9944	0.9943	0.9944
DecisionTree	0.9919	0.9914	0.9916	0.9914
ExtraTrees	0.9942	0.9942	0.9941	0.9942

As presented in Table 5, each detection model showed performance in terms of accuracy and other measures for multi-class classification without optimization. It was observed that all models could achieve more than or equal to 99% accuracy in binomial classification.

6.3 Results of Multi-Class Classification with Optimization

Results of multi-class classification with optimization are provided in terms of confusion matrix-based statistics and performance statistics.

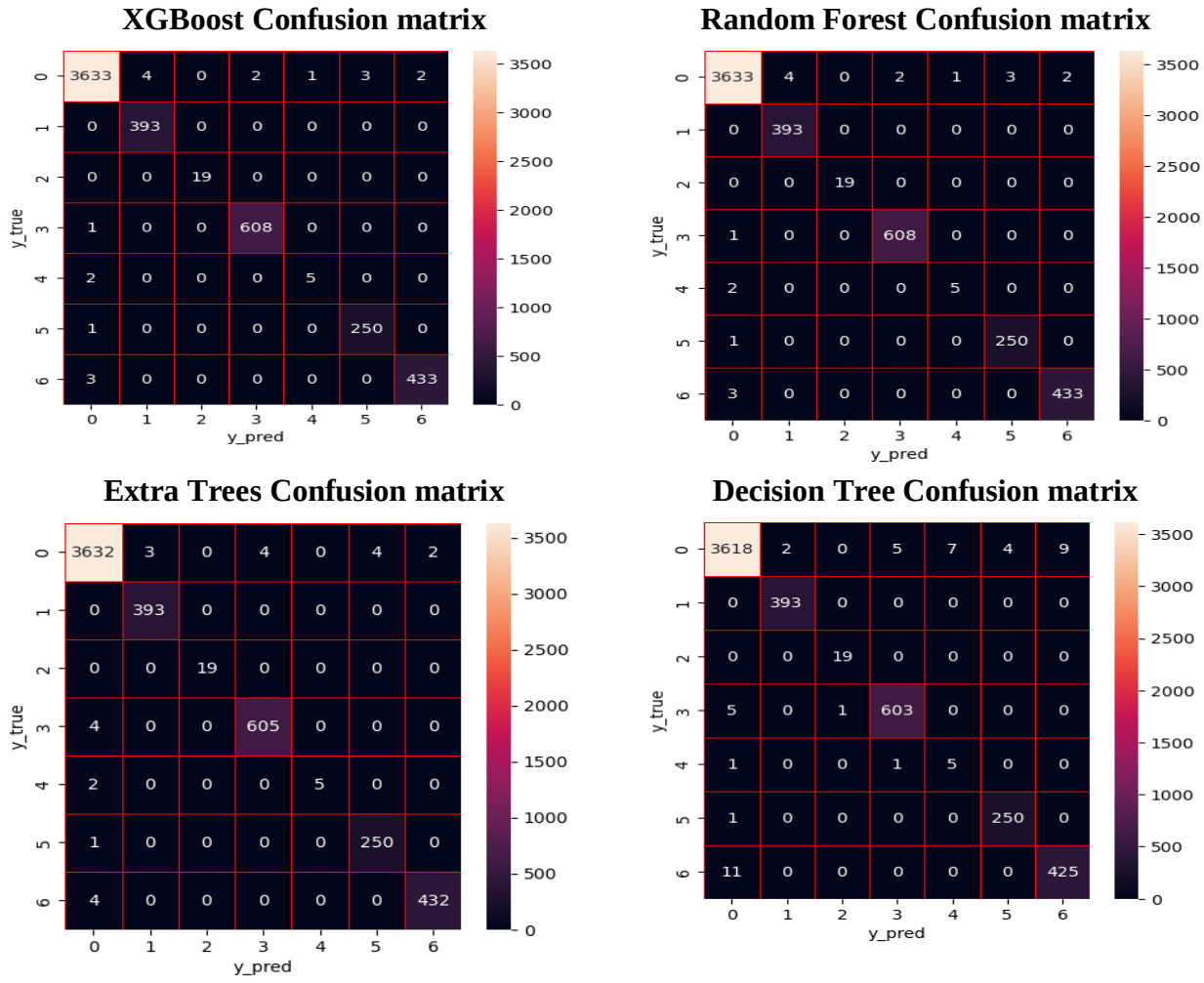

Figure 7: Results of multi-class classification with optimization in terms of confusion matrix

As presented in Figure 7, the predicted labels are compared with ground truth labels for all the four ML models and for all classes. Table 6 shows the computed performance statistics based on 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**
XGBoost	0.9964	0.9964	0.9964	0.9964
Random Forest	0.994	0.994	0.994	0.994
DecisionTree	0.9925	0.9918	0.992	0.9918
ExtraTrees	0.9955	0.9955	0.9954	0.9955

As presented in Table 6, each detection model showed performance in terms of accuracy and other measures for multi-class classification without optimization. It was observed that all models could achieve more than 99% accuracy in binomial classification. However, the performance of models with optimization is slightly increased when compared with the models without optimization.

6.4 Discussion

In this research an intrusion detection system is designed and implemented using ML models. It is based on supervised learning phenomena as there is labelled data available for empirical study. CICIDS2017 dataset is used for empirical study. Though this dataset has more citations and relatively old one, investigation on datasets found that this dataset is better than many other datasets in terms of benchmarking and provision for different kinds of intrusions. Though there are many ML models available, four tree based approaches are chosen for this research. The rationale behind this is that the tree-based models were found to have better performance as per the review of literature presented in Section 2. Since the ML models chosen are based on supervised learning, the quality of training data matters. For this reason, quality of training data is improved using two feature selection techniques such as information gain and FCBF. Feature selection is one of the optimization techniques used in this research. Other optimization technique is hyperparameter optimization and this is carried out using Bayesian optimization concept.

Experiments are designed to have three categories. The first category of experiments is made with the ML models using binomial classification approach. It does mean that each model will classify test samples into two classes such as BENIGN and INTRUSION. The second category of experiments is made with the ML models using multi-class classification approach without optimizations aforementioned. It does mean that each model will classify test samples into several classes such as BENIGN, Bot, Brute Force, DoS, Infiltration, Port Scan and Web Attack. The third category of experiments is made with the ML models using multi-class classification approach with optimizations aforementioned. It does mean that each model will classify test samples into several classes such as BENIGN, Bot, Brute Force, DoS, Infiltration, Port Scan and Web Attack. The first category is preferred as it simply provides outcome for each test instance in terms of BENIGN or INTRUSION. This far is useful for some use cases where intrusion detection is sufficient and classification is not required. In some other use cases, the administrator of network wants to know actual class of attack or intrusion. In such cases, multi-class classification is useful. As per the research gaps found in the literature such as need for optimization of ML models in terms of feature engineering and hyperparameter optimization, the optimizations in this research could improve performance but there is no much improvement.

7 Conclusion and Future Work

This research is aimed at designing and implementing an intrusion detection system based on ML models. The system is implemented using four ML models with two optimizations such as feature selection and hyperparameter optimization. These research objectives are met as evident in the results of the research. The first objective is to investigate on the existing ML methods and related works used to realize intrusion detection systems. It is achieved and outcome is presented in Section 2. The second objective is to propose a ML based framework

with its mechanisms and optimizations for efficient intrusion detection. The framework its details are presented in Section 4. The framework is realized with ML models and optimization techniques. The third objective is to explore feature selection and hyperparameter optimization for improving performance of ML models in intrusion detection. This objective is achieved by implementing feature selection and hyperparameter tuning. Quality of training data is improved using two feature selection techniques such as information gain and FCBF. Feature selection is one of the optimization techniques used in this research. Other optimization technique is hyperparameter optimization and this is carried out using Bayesian optimization concept. The fourth objective is to evaluate the proposed framework. It is done using performance metrics that are widely used in intrusion detection research. Finally, conclusions are drawn as found in this section and also future work possibilities are provided. With regard to research questions, the first research question is "Can machine learning models be used for realizing an intrusion detection system?". This research question is found affirmative answer as the ML models were found suitable for intrusion detection. The second research question is "Can optimizations like feature selection and hyperparameter optimization have impact on intrusion detection performance of ML models?". This research question also found affirmative answer in this research and in literature also. However, in this research the improvement of accuracy when optimizations are made to ML models is relatively less. In other words, there is no significant improvement in the accuracy when optimizations are applied. There might be reasons for this due to dataset limitations or other reasons. This needs to be investigated further in future. In the binomial classification highest accuracy is achieved by RF model with 99.87%. In case of multi-class classification without optimizations, highest accuracy is exhibited by RF with 99.44%. In case of multi-class classification with optimizations, highest accuracy is exhibited by XGBoost with 99.64%.

This research has certain limitations that need to be overcome in future endeavours. First, the dataset used in this research is old and the experiments and observation are only based on one dataset. In future, it is possible to work with more datasets considering more recent datasets that are less explored. Second, the ML models used are three based. Usage of neural network-based models including deep learning models could improve performance. This is another direction for possible future scope of the research. Third, the research carried out is based on the datasets available. However, intrusion detection considering a live network traffic as test data is very much desired. Towards this end, in future, test data can be captured live from networks in real time.

References

Abdulhammed, R., Faezipour, M., Abuzneid, A. and AbuMallouh, A. (2018). Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. *IEEE Sensors Letters*, 1–4. http://doi:10.1109/LSENS.2018.2879990

Abrar, I., Ayub, Z., Masoodi, F. and Bamhdi, A. M. (2020). A machine learning approach for intrusion detection system on nsl-kdd dataset, *IEEE*, 919–924. http://doi:10.1109/ICOSEC49089.2020.9215232

Alkadi, O., Nour, M. and Benjamin, T. (2020). A review of intrusion detection and blockchain applications in the cloud: approaches, challenges and solutions. *IEEE Access*, 8, 104893–104917. http://doi:10.1109/ACCESS.2020.2999715

Al-Omari, M., Rawashdeh, M., Qutaishat, F., Alshira'H, M. and Ababneh, N.. (2021). An intelligent tree-based intrusion detection model for cyber security. *Journal of Network and Systems Management*, 1–18. http://doi:10.1007/s10922-021-09591-y

Anthi, E., Williams, L., Słowinska, M. and TheodorakopoulosPe, G. (2019). A supervised intrusion detection system for smart home iot devices. *IEEE*, 1-13.
Ashiku, L. and Dagli, C. (2021). Network intrusion detection system using deep learning . *Procedia Computer Science*, 1–9. http://doi:10.1016/j.procs.2021.05.025

Azam, R., Jawaid, S. M. and Munir, A. S. (2020). Machine and deep learning based comparative analysis using hybrid approaches for intrusion detection system, *IEEE*, 1–9. http://doi:10.1109/ICACS47775.2020.9055946

Costa, K. A.P. da, Papa, J. P., Lisboa, C. O., Munoz, R. and Albuquerque, V. H. C. de .(2019). Internet of Things: A Survey on Machine Learning-based Intrusion Detection Approaches. *Computer Networks*, 1–14. http://doi:10.1016/j.comnet.2019.01.023

Dini, P., Elhanashi, A., Begni, A. and Saponara, S. (2023). Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity. *MDPI*, 1-34.

Ferrag, M. A., Maglaras, L., Moschoyiannis S. and Helge. (2019). Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 1-19.

Ferraga, M. A., Maglaras, L., Moschoyiannis, S. and Helge. (2020). Machine learning approaches to network intrusion detection for contemporary internet traffic. *Journal of Information Security and Applications*, 1-19.

Firoz, K. M. and Sven, H. (2018). Cyber security challenges: an efficient intrusion detection system design, *IEEE*, 19–24. http://doi:10.1109/YEF-ECE.2018.8368933

Gumusbas, D., Yldrm, T., Genovese, A. and Scotti, F. (2020). A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. *IEEE Systems Journal*, 1–15. http://doi:10.1109/JSYST.2020.2992966

Hernandez-Jaimes, M. L., Martinez-Cruz, A. and Alejandra, K.. (2023). Artificial intelligence for IoMT security: a review of intrusion detection systems, attacks, datasets and Cloud–Fog. *Esevier*, 1-33.

Ilhan, F. K., Fatih E. and Abdulkadir S.; (2021). Machine learning methods for cyber security intrusion detection: datasets and comparative study . *Computer Networks*, 1–16. http://doi:10.1016/j.comnet.2021.107840

Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., and Benjamin T. (2023). An explainable deep learning-enabled intrusion detection framework in IoT networks. *Elsevier*, 1-20.

Kocher, G. and Kumar, G. (2021). Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. *Soft Computing*, 1-33. http://doi:10.1007/s00500-021-05893-0

Liu, H. and Lang, B (2019). Machine learning and deep learning methods for intrusion detection systems: a survey. *Applied Sciences*, 9(20): 1–28. http://doi:10.3390/app9204396

Liu, H., Zhong, C., Alnusair, A. and Islam, S. R.. (2021). FAIXID: A framework for enhancing ai explainability of intrusion detection results using data cleaning techniques. *Journal of Network and Systems Management*, 1–30. http://doi:10.1007/s10922-021-09606-8

Liu, L., Wang, P., Lin, J and Liu, L. (2021). Intrusion detection of imbalanced network traffic based on machine learning and deep learning. *IEEE Access*, 1–14. http://doi:10.1109/access.2020.3048198

Macas, M. and Wu, C. (2020). Review: deep learning methods for cybersecurity and intrusion detection systems. *IEEE Latin-American Conference on Communications (LATINCOM)*, 1–6. http://doi:10.1109/latincom50620.2020.9282324

Manickam M. and Rajagopalan S. P. (2018). A hybrid multi-layer intrusion detection system in cloud. *Cluster Computing*, 1–9. http://doi:10.1007/s10586-018-2557-5

Marek, P., Micha, C. and Rafa, K. (2020). Defending network intrusion detection systems against adversarial evasion attacks. *Future Generation Computer Systems*, 110: 148–154. http://doi:10.1016/j.future.2020.04.013

Maseer, Z. K., Yusof, R., Bahaman, N., Mostafa, S. A., and Foozy, C. F. M. (2021). benchmarking of machine learning for anomaly based intrusion detection systems in the cicids2017 dataset. *IEEE Access*, 1–20. http://doi:10.1109/access.2021.3056614

Mohan, K. A. and Kumar, T. A. (2020). Intrusion detection in intelligent transportation system and its applications using blockchain technology, *IEEE*, 1–8. http://doi:10.1109/ic-ETITE47903.2020.332

Pérez S. I., Rubio S. M. and Criado R. (2021). A new approach to combine multiplex networks and time series attributes: building intrusion detection systems (ids) in cybersecurity. *Chaos, Solitons & Fractals*, 150: 1-11. http://doi:10.1016/j.chaos.2021.111143

Pooja, T. and Purohit, S. (2021). Evaluating neural networks using bi-directional lstm for network ids (intrusion detection systems) in cyber security. *Global Transitions Proceedings.* 1-14. http://doi:10.1016/j.gltp.2021.08.017

Ramesh, T. R., Jackulin, T., Kumar, R. A., Chanthirasekaran, K. and Bhara, M. (2024). Machine learning-based intrusion detection a comparative analysis among datasets and innovative feature reduction for en. *International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING.* 12(12): 200–206.

Santos, L., Rabadao, C. and Goncalves, R. (2018). Intrusion detection systems in internet of things: a literature review, *IEEE*, 1–7. http://doi:10.23919/CISTI.2018.8399291

Sarhan, M., Layeghy, S., Moustafa, N., and Gallagher, M. M. (2022). Feature extraction for machine learning-based intrusion detection in IoT networks. *Digital Communications and Networks*, 1-12.

Sarker, I. H., Abushark, Y. B., Alsolami, F. and Khan, A. I. (2020). intrudtree: a machine learning based cyber security intrusion detection model. *Symmetry*, 12(5): 1–15. http://doi:10.3390/sym12050754

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A. and Xu, M. (2020). A survey on machine learning techniques for cyber security in the last decade. *ieee access*, 8, 222310–222354. http://doi:10.1109/access.2020.3041951

Shaukat, K., Luo, S., Varadharajan, V., Hameed, I. A., Chen, S., Liu, D. and Li, J. (2020). Performance comparison and current challenges of using machine learning techniques in cybersecurity. *Energies*, 13(10): 1–27. http://doi:10.3390/en13102509

Sikos, L. F. (2019). Machine learning algorithms for network intrusion detection. *Intelligent Systems Reference Library*, 151: 151–179. http://doi:10.1007/978-3-319-98842-9_6

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*, 1–9. http://doi:10.1007/s12083-017-0630-0

Tama, B. A. and Lim, S. (2021). Ensemble learning for intrusion detection systems: a systematic mapping study and cross-benchmark evaluation. *Computer Science Review*, 1–27. http://doi:10.1016/j.cosrev.2020.100357

Ustun, T. S., Hussain, S. M. S., Ulutas, A., Onen, A., Roomi, M. M. and Mashima, D. (2021). Machine learning-based intrusion detection for achieving cybersecurity in smart grids using iec 61850 goose messages . *Symmetry*, 1–15. http://doi:10.3390/sym13050826

Vigneswaran, R., Soman K. P. V. and Poornachandran, P. (2018). evaluating shallow and deep neural networks for network intrusion detection systems in cyber security. *IEEE*, 1-6.

Wang, G., Feng, J., Bhuiyan, M. and Lu, Rongxing (2019). Effectiveness of machine learning based intrusion detection systems, 277–288. http://doi:10.1007/978-3-030-24907-6_21

Yang, L., Moubayed, A. and Shami, A. (2022) MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles. *IEEE Internet of Things Journal*, 1–17. http://doi:10.1109/jiot.2021.3084796