Flow-Based Network Intrusion Detection using Hybrid Machine Learning Techniques

MSc Research Project
MSc in Cybersecurity

Ashutosh Datta Ganeshkar
Student ID:x23142171

School of Computing
National College of Ireland

Supervisor: Mr.Eugene McLaughlin

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Ashutosh Datta Ganeshkar ………

Student ID: x23142171 ………

Programme: MSc in Cyber Security …………………………………………………………… **Year:** 2023-24 …………………………..

Module: MSc Research Project ………

Supervisor: Mr. Eugene McLaughlin ………

Submission Due Date: 12/08/2024 ………

Project Title: Flow-Based Network Intrusion Detection using Hybrid Machine Learning Techniques. ………

Word Count: 7208 ……………………………………… **Page Count** 20 ……………………………………………………………

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: Ashutosh Datta Ganeshkar ………

Date: 12/08/2024 ………

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):	

AI Acknowledgement Supplement

MSc Research Project

Flow-based Network Intrusion Detection using Hybrid Machine Learning Techniques

Your Name/Student Number	Course	Date
Ashutosh Datta Ganeshkar	MSc in Cyber Security	12/08/2024

This section is a supplement to the main assignment, to be used if AI was used in any capacity in the creation of your assignment; if you have queries about how to do this, please contact your lecturer. For an example of how to fill these sections out, please click here.

AI Acknowledgment

This section acknowledges the AI tools that were utilized in the process of completing this assignment.

Tool Name	Brief Description	Link to tool
NA	NA	NA

Description of AI Usage

This section provides a more detailed description of how the AI tools were used in the assignment. It includes information about the prompts given to the AI tool, the responses received, and how these responses were utilized or modified in the assignment. **One table should be used for each tool used**.

NA	
NA	
NA	NA

Evidence of AI Usage

This section includes evidence of significant prompts and responses used or generated through the AI tool. It should provide a clear understanding of the extent to which the AI tool was used in the assignment. Evidence may be attached via screenshots or text.

Additional Evidence:

NA

Additional Evidence:

NA

Flow-Based Network Intrusion Detection using Hybrid Machine Learning Techniques

Ashutosh Datta Ganeshkar
x23142171

Abstract

Modern network intrusion systems require the detection of breaches to be secure, but it is a difficult task as the network data is diverse and dynamic. By integrating oversampling technique with ensemble Machine Learning methods, this study investigates ways to enhance Intrusion Detection in flow-based data. The purpose of the study is to evaluate how well the hybrid ensemble technique detects the intrusions. The UNSW-NB15 dataset was evaluated to train algorithms such as AdaBoost, XGBoost, Gradient Boosting and Stacking classifier/Hybrid Ensemble. Using SMOTE technique, the class imbalances were addressed by balancing the data. The findings show that, other referenced studies showed accuracy above 90%, the Hybrid Ensemble model performed better with accuracy of 89.59% on balanced data using SMOTE method To identify different network attacks, the results shows that ensemble learning method significantly improves detecting the attacks and accuracy. The advantages of combining different ensemble methods with oversampling technique like SMOTE is very essential. To raise network security, model suggests creating sophisticated detection system with hybrid model. To make sure these solutions are broadly applicable and reliable, the future work should focus on optimizing model parameters and test new methods.

Keywords: Network Intrusion Detection System (NIDS), Machine Learning, SMOTE (Synthetic Minority Oversampling technique), Ensemble Learning, AdaBoost, XGBoost, Gradient Boosting.

1 Introduction

Since, Internet technologies have developed, securing our network is more complicated, which has increased the frequency and complexity of cyberattacks. To identify accurately between malicious and authorized network traffic, it is required to develop advanced **Network Intrusion Detection Systems or NIDS**. It is very crucial parts of cybersecurity, as they are made to detect illegal activities which may compromise network integrity and data security. The traditional methods for detecting intrusions have shown serious drawbacks (Verma et al., 2018). The methods like anomaly and signature-based systems are one of them. Systems that use anomaly-based algorithms identify anomalies to minimize threats and establish a baseline of typical network behaviour. On other hand, signature-based are ineffective against new attacks as they cannot identify them properly because they depend on preset patterns of recognized threats.

Machine learning is one of the effective ways to improve NIDS's capabilities to tackle different issues. These ML models can recognize complex patterns and increase detection rates, also it can analyse large volumes of network traffic data. The ensemble learning or boosting algorithms have shown promising results. By recent experiments in detecting network intrusions have proven efficacy of these boosting algorithms (Husain et al., 2019). XGBoost can handle more large and complex datasets, well known for its efficiency and scalability. Adaboost had been used to focus on difficult data points to increase detection rates and keeping false positive rates low. By decreasing the errors and raising accuracy the Gradient Boosting improves the model using the technique of gradient descent, enhancing this method.

To create a successful NIDS still faces several challenges. The main problem with the imbalance in network traffic datasets, where unauthorized activities are less frequent than normal traffic. The second problem is identifying the important characteristics from high-dimensional network traffic data, however it is necessary to create accurate and successful models. To address these issues, this **research focus** on a **Hybrid ensemble model** or **Stacking classifier** (Adegboyega, 2024) a novel method that combines the **AdaBoost, gradient boosting** and **XGBoost** classifiers with methods like **Synthetic Minority Over-sampling Technique (SMOTE)** for sharing data for class imbalance and feature selecAtion (Md. Alamin Talukder et al., 2024). This approach seeks to enhance the NIDS efficiency and accuracy providing a safe and secure defence against constantly changing threats which are affecting our network security.

1.1 Motivation

The main motivation of this study is to investigate and verify the complex methods, addressing the current attacks and helping in creation of stronger network intrusion detection systems. Due to Increasingly complex and frequent cyberattacks it offers a serious risk to network security, which leads for creating an enhanced detection system. The traditional methods suffer from high false positive rates and a fail to identify new attacks. As the network environments get more complicated it requires more advanced techniques which can effectively identify the malicious activities. The Machine learning methods offers promising results by improving detection accuracy and lowering the false positives, in particular, ensemble methods when combined with oversampling techniques will offer an extra layer to the network intrusion detection systems.

1.2 Research Question

- How can the use of Oversampling techniques in combination with ensemble machine learning methods enhance the detection of network intrusions in Flow-Based network data?

1.3 Research Objective

- The objective is to implement and establish a Novel Ensemble Method/ Stacking Classifier which combines the different features of AdaBoost, XGBoost and Gradient Boosting for intrusion detection.

- Using SMOTE Method (Synthetic Minority Over-Sampling Technique) it will balance the imbalanced dataset to ensure that minority classes are equally displayed while training the model.

1.4 Research Hypothesis

H0: The Hybrid ensemble method cannot identify network traffic accurately than traditional methods.

H1: The hybrid ensemble method can identify network traffic accurately than traditional methods.

H0: The Hybrid ensemble method used by intrusion detection model cannot determine that the data is malicious or not.

H1: The Hybrid ensemble method used by intrusion detection model gives high accuracy and can determine the data is malicious or not.

1.5 Structure of the report

Chapters	Description	Details
2	Related Work	This section will cover the literature review to network intrusion detection using machine learning.
3	Research Methodology	This third section will cover the overall approach and methods/algorithms carried out.
4	Design Specification	This section will shed a light on design specification of the proposed model for the thesis.
5	Implementation	This section will describe the steps taken to implement the proposed machine learning models.
6	Evaluation	This section will explain the results of ML models evaluation using different performance metrics and findings.
7	Conclusion and Future work	This will highlight the findings and contribution made and will explain what will be explored in future for more better performance.

Table 1: Report Structure

2 Related Work

In this section, I will discuss the materials I read to develop my thesis, highlighting the advantages and disadvantages of different machine learning and ensemble techniques.

2.1 Machine Learning methods in Intrusion Detection

The UNSW-NB15 dataset was used in this research paper (Husain et al., 2019). This study describes the development of a network intrusion detection system (NIDS) using XGBoost Algorithm to improve its detection accuracy and operating efficiency. The dataset is very well-known for its effectiveness in the intrusion detection. They have chosen this algorithm because of its novel approach in managing big, complex datasets and has excellent intrusion detection accuracy. The disadvantage in this study which is mentioned was its high computational complexity of the model and its requirement for rigorous hyperparameter modifications, whereas the advantage it has is that it shows that the use of XGBoost significantly boosts NIDS efficiency, which makes it a useful method for modern cyber-attacks.

The author in his research study (Verma et al., 2018) is describing a combination of clustering techniques with machine learning algorithms such as XGBoost and AdaBoost to improve the detection. The model is trained and tested by using a well-known dataset which is NSL-KDD dataset which has wide ranges of network traffic characteristics. Increasing the detection accuracy, this combined method. Seeks to decrease false positives. The limitation in this study includes possible overfitting and additional difficulty of combining various approaches. It has also discovered that by utilizing the advantages of both clustering and boosting techniques, the hybrid model is performing better than conventional single method approaches.

To detect cyber-attacks, the author in this paper (More et al., 2024) uses the techniques of logistic regression, support vector machine (SVM), decision tree and random forest (RF). The study focuses on exploratory data analysis and on feature selection through correlation analysis and random sampling. The results shows that random forest can detect cyber-attacks in the best way, this model works better with a low false rate of 1.36% and accuracy of 98.63%. The study's thorough analysis and strong accuracy rates tells us that how much it offers to improve IDS performance. The disadvantage of this model is its processing cost of random forest algorithm and its complex nature of feature selection.

(Kocher and Kumar, 2020) in this paper, a variety of ML classifiers are evaluated to determine its efficacy how well they are in detecting network intrusions. The UNSW-NB15 dataset is used to research because of it modern and broad dataset for network intrusion detection. The classifiers which are used to investigate are Naive Bayes (NB), Random Forest (RF), Decision Tree, K-Nearest Neighbors (KNN), Logistic Regression (LR) and stochastic gradient descent (SGD). The findings show us that the Random forest performs better then other algorithms with an accuracy of 95.43%. The evaluation is performed using metrics like accuracy, precision, recall, F1-score, MSE (mean squared error), True and false positive rate. The limitations discussed for each classifiers includes the large mean squared error of the Naive bayes classifier and cost of computing for training the decision tree. The classifier which showed strong durability and accuracy was Random Forest which makes it dependable option for network intrusion detection.

2.2 Handling Imbalanced data in intrusion detection

This research paper (Zhang, Jia and Shang, 2022) is addressing the problem of categorizing imbalanced datasets by using XGBoost method. In this paper the author is providing a better classification algorithm that combines XGBoost with over-sampling method SVM-Smote and under sampling method EasyEnsemble. Using Bayesian optimization, the optimal situations are automatically identified and modified. The dataset utilized in this study are two publicly available datasets, first one is credit card dataset from UCI and second dataset is credit fraud dataset from ULB. The modified XGBoost algorithm performs better when the testing results are compared to other models like RUSBoost, CatBoost and LightGBM. The performance of classification on imbalanced datasets is greatly improved by combining complex sampling approaches with XGBoost. The one limitation which this model has is the computational complexity of Bayesian optimization process.

The main topic of this research paper (Md. Alamin Talukder et al., 2024) are the challenges involved for enormous and unbalanced data in network intrusion detection. The author has provided a thorough framework that integrates feature extraction, stacking feature embedding and oversampling technique to increase the detection accuracy. To validate the performance for several Machine Learning methods the UNSW-NB15 dataset was used. The findings have a good result in which it showed a considerable increase in the accuracy and a decrease in false positive. Despite of extra complexity and processing resources needed the advantages are that it has improved the detection capabilities for intrusions and improved handling of class imbalance.

The paper (Gouveia and Correia, 2020) is highlighting about XGBoost algorithm's use for network intrusion detection and it is better at predicting accuracy and easy to handle huge datasets such as UNSW-NB15. The model shows excellent accuracy and is successful in detecting intrusion. The disadvantage is the computational load and hyperparameter tuning to get good performance. This model has a capacity to manage unbalanced datasets as well as its robustness in providing precise and trustworthy predictions in securing our networks.

2.3 Ensemble Methods for intrusion detection

The Hybrid approach to intrusion detection is presented in this study (Mashuqur Rahman Mazumder et al., 2021) which combines supervised and unsupervised ML algorithms. The NSL-KDD dataset is used by the researcher in this study. K-means clustering and light gradient boosting algorithms are combined in this hybrid model. It is performing traditional methods like adaboost, xgboost and random forest with a detection accuracy of 90.41%. The advantage of this model is its significant increase in detection rates which makes it extremely useful for network security. Whereas, the drawback is its complexity of combining several algorithms and the time taken to train the model.

In this research study (Kiflay, Tsokanos and Kirner, 2021) the author suggests an ensemble method that uses indirect voting approach to integrate Random forest, adaboost, xgboost and gradient boosting decision tree. Increasing the detection performance, the goal of the study is to decrease the false alarm rates. To validate the model two datasets are used one UNSW-NB15 dataset and second is NSL-KDD dataset. The limitations involved are the possibility of increased operational complexity and the requirement for thorough modification of ensemble parameters. The NIDS's improved robustness and detection ability is the major advantage. This strategy reduces false alarms and improves the detection rate by combining the different ensemble methods.

The author in his paper (Zoghi and Serpen, 2022) have examined the development and evaluation of machine learning classifier which are built for UNSW-NB15 dataset. The two major problems such as class overlap, and imbalance is identified in this paper. To overcome the problem the author is using ensemble approaches such as random forest improved by Hellinger distance decision tree (RF-HDDT), XGboost and balanced bagging (BB). The two unique approaches that modify the final classification options are put forth to address the class overlap issue. The ensemble classifier which uses a majority vote combines performs better than other models when it is tested for binary and multi category classification. The drawback highlighted are its complexity and computational requirements.

The paper (Almomani et al., 2023) has three foundation models random forest, decision tree and k-nearest are integrated into the authors proposed Intrusion Detection System (IDS) along with a meta-model represented by logistic regression. They have implemented a stacking ensemble technique to enhance intrusion detection system's performance. It uses UNSW-NB15 dataset for testing and has obtain a testing phase accuracy of 97.95%. By utilizing the capabilities of individual classifiers, the ensemble approach increases overall accuracy and decreases error rates. The main benefit of this method is that it can detect network intrusions with greater accuracy and robustness, whereas the disadvantage is the complexity and processing required for combining different techniques.

In this study (Adegboyega, 2024) the author suggests a Hybrid strategy to improve intrusion detection that combines AdaBoost and stochastic gradient descent classifier (SGDC). The analyses is done by using KDD CUP 99 dataset. After evaluating the two hybrid models, SGDC_ADA and ADA_SGDC, the first one i.e SGDC_ADA model gave a accuracy rate of 97%. In terms of precision, recall and F1-score the hybrid model outperforms then the individual model. The complexity and training time required to train the model is the main disadvantage, but model is also useful because of enhanced accuracy and decreased error rates.

An ensemble methodology to enhance the detection system performance is presented in this research paper (Ngamba Thockchom, Moirangthem Marjit Singh and Nandi, 2023). The model uses stacking ensemble technique in which it combines decision tree, logistic regression and gaussian naïve bayes as base classifiers with stochastic gradient descent as the meta classifier. There are three different types of datasets used to evaluate the performance, first is UNSW-NB15, CIC-IDS2017 and KDD cup 1999. The main advantage of this model is that it improved detection accuracy and effective handing of unbalanced datasets. It was difficult to combine multiple algorithms and increase analyzing demands. This approach focuses on choosing the most important features by using Chi-square test for feature selection. The outputs showed that the ensemble model performs better and beats individual classifiers in binary and multiclass classifications.

3 Research Methodology

This research investigates in developing a secure and efficient **Network Intrusion Detection System (NIDS)** which can be used to detect different malicious attacks in flow-based network data. Initially the **UNSW-NB15 dataset** was gathered from a very well-known site called as **Kaggle** (www.kaggle.com, n.d.). The dataset contains over two million records with different types of attacks such as worms, shellcode, fuzzers, analysis, backdoor, denial-of-service, exploits and generic attacks, etc and it also captures a hybrid of real and synthetic network traffic. In the next step the necessary libraries were loaded such as NumPy, Seaborn, Scikit-learn, Pandas and machine learning algorithms like AdaBoost, XGBoost and Gradient Boosting. After that, to ensure all the data is in a correct format, the CSV files containing the dataset were loaded into Panda's data frames.

The next step is data cleaning which is very important. The data cleaning process was then utilized to fill up the gaps, format the data properly and eliminate duplicates so it can be analysed. To study the distribution patterns and feature correlations and to visualize the data, which was cleaned, I have plotted Histograms, Bar plot and scatter plot. For normalizing the data, the pre-processing step includes scaling numerical features and encoding categorical categories. By using the bar plot, I was able to identify the imbalance data, to address class imbalance in the dataset **SMOTE (Synthetic Minority Over-Sampling Technique)** (Brownlee, 2020) was used to create synthetic samples for missing classes. Now the next stage is splitting the data, I utilize an **80-20 split**, in this the balanced dataset is then divided into **training** and **testing** sets to train the machine learning models.

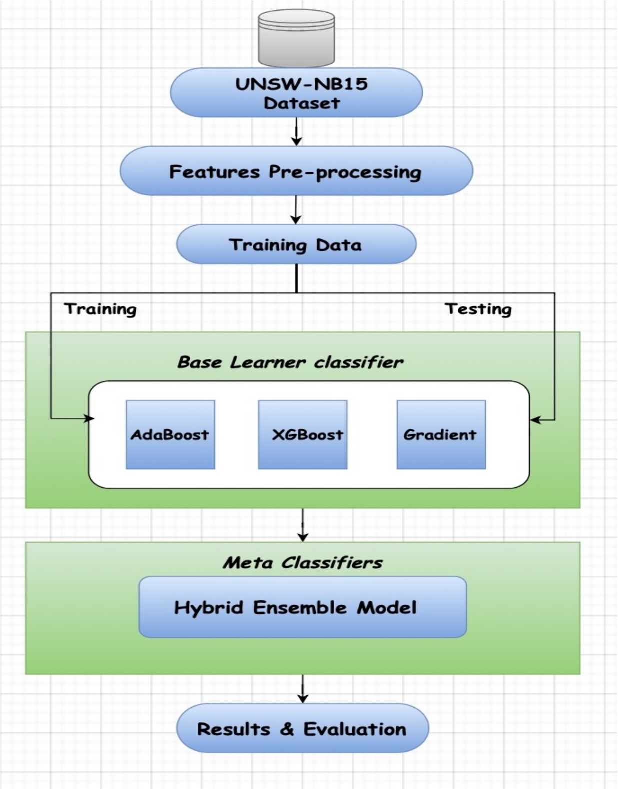


Figure 2: Flow diagram for research model

The main component of the approach is training machine learning models, which include **AdaBoost, XGBoost, Gradient Boosting, and Hybrid Ensemble Model/Stacking classifier** that will combine these techniques with Gradient Boosting acting as the meta-classifier. To train each models pre-processed and balanced dataset using SMOTE is used, to utilize its strengths to improve the performance. The models are assessed using accuracy score, classification report that include information on each class's precision, recall, f1-score, and confusion matrix. Enhancing the precision and dependability of network intrusion detection in flow-based network data, this structured approach guarantees a strong model training and evaluation by handing important issues like class imbalance and utilizing the potential of ensemble learning. Being a rigorous approach, it should give a efficient response to today's cyber-attacks, providing significant advantages over traditional detection methods.

4 Design Specification

This section will outline the machine learning models and their framework and assessment metrics used in this proposed model.

4.1 AdaBoost Classifier

AdaBoost also known as **Adaptive Boosting**, is an ensemble learning method which builds strong classifier by combining different weak classifier. The algorithm iteratively modifies the weights of training instances, by focusing on situations that are very difficult to categorize. The accuracy and resilience of every new model are increased in this approach as it fixes the mistakes of the previous ones. This algorithm is well known for its simplicity for using it and the way it works for binary classification issues (www.datacamp.com, n.d.).

4.2 XGBoost Classifier

Extreme Gradient Boosting or **XGBoost** is a complex and robust gradient boosting implementation. To better handle the limited information, this model combines multiple improvements, including parallel processing for faster computations, regularization to prevent overfitting and a weighted quantile model. XGBoost is a well-liked choice for numerous machine learning challenges and real world issues because of its excellent results with structured or tabular data (Nvidia, n.d.).

4.3 Gradient Boosting Classifier

Gradient Boosting is an ensemble technique. In this method the models are built one after the other, with each new model aiming to fix the mistakes of previous models. The new model is adapted to the remaining errors/issues of the sum of the earlier models to achieve it. The overall accuracy is increased and improved due to the process of minimizing the loss function. It works well for both regression and classification applications because of its great versatility for different loss functions (Saini, 2021).

4.4 Hybrid Ensemble Model

The **Hybrid Ensemble Model** incorporates several base classifiers, including gradient boosting, xgboost and adaboost using a stacking classifier. The predictions made by the base classifiers, this method uses a meta classifier's characteristics. To get the final predictions, the meta classifier is trained, which is another powerful model gradient boosting. This hybrid technique improves the prediction performance and resilience by utilizing the advantages of each basic classifier (rasbt.github.io, n.d.). This model really works good with complicated datasets that have a wide range of patterns and features.

In my model the **UNSW-NB15** is the suitable dataset as it is **complicate** and has **wide range of patterns and features**. Hence, I choose this dataset to create this model.

4.5 Proposed Model Training and Evaluation Metrics

The Important components of a machine learning models are the Evaluation Metrics. In my proposed model, the 80% of the dataset is utilized for training and 20% is used for testing. To train the basic classifiers the pre-processed and balanced dataset are used. Later, by combining their predictions the meta-classifier is trained. To determine its efficacy, several performance criteria are used to evaluate the final hybrid ensemble model. Among these metrics are as follows:

4.5.1 Accuracy

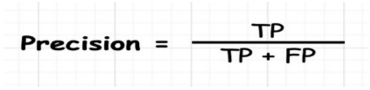
Accuracy is the percentage of accurate predictions the model makes out of all the predictions. It is calculated as:

$$\text{Accuracy} = \frac{TP + FP}{TP + TN + FP + FN}$$

In the situation of imbalanced datasets when the metric can be dominated by the number of true negatives, accuracy might not be enough on its own even it is an easy measure of overall correctness.

4.5.2 Precision

The precision model is determined by dividing all positive predictions by the percentage of true positive predictions. It is calculated as follows:

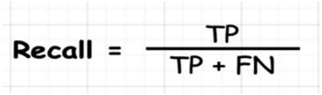


$$\text{Precision} = \frac{TP}{TP + FP}$$

A low false positive rate is indicated by high precision, in situations where the cost of false positives is essential.

4.5.3 Recall

Recall also known as sensitivity, to calculate the recall percentage of accurate positive predictions among any actual positive cases is measured. It can be calculated as below:

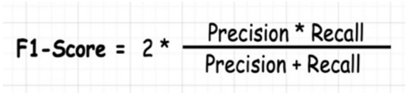


$$\text{Recall} = \frac{TP}{TP + FN}$$

If the recall rate is high that means the model can properly detect many positive events.

4.5.4 F1-score

The F1-score is known as a statistic that balances precision and recall by taking the harmonic mean of the two. It is calculated as follows:



$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F1-score is very helpful when we are evaluating the trade-off between precision and recall, especially when dealing with imbalanced datasets.

4.5.5 Support

A support is the number of real instances of each class in the dataset. A model’s performance on that class is based on a greater number of instances if the class with high support has more confidence in its evaluation metrics.

4.5.6 Confusion Matrix

In confusion matrix, a table that compares actual classification vs predicted classification to give a thorough analysis of the model’s performance. It consists of:

True Positives (TP): In True Positive, the cases that are correctly predicted as positive as known as TP.
False Positives (FP): In False Positive, the cases that are incorrectly predicted as positive are known as FP.
True Negatives (TN): In True Negative, the cases that are correctly predicted as negative are known as TN.

False Negatives (FN): In False Negative, the cases that are incorrectly predicted as negative are known as FN.

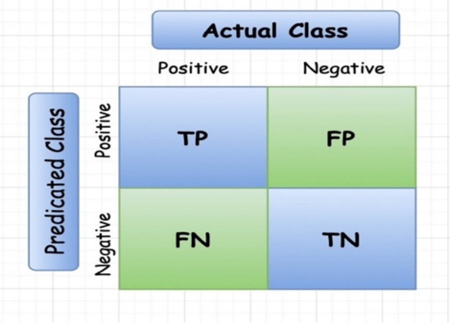


Figure 2: Confusion Matrix

The visualization of the classification model's performance can be improved by the confusion matrix, if it displays both kinds of errors being produced and the quantity of accurate predictions.

5 Implementation

In this stage, I will highlight how machine learning techniques and ensemble learning techniques were used to create a machine learning model.

5.1 Software and Programming Language

The programming language which I have selected to carry out my project is Python. The **python 3.6.3** version was used for writing the code. I have selected python because of its huge number of libraries and frameworks that are designed for machine learning models. For my project, the testing environment which I have chosen is a cloud based platform known as **Google Colab** as its very easy to use, code and has robust capabilities. The python supporting Jupyter notebooks were accessed for free via Colab which makes it a best option for all the machine learning applications. It also provides access to GPU (Graphical Processing Unit) and TPU (Tensor Processing Unit) which can speed up the complex model training drastically. Furthermore, colabs interaction with google drive makes it simple to collaborate and retrieve datasets, which is especially helpful for big datasets like UNSW-NB15.

5.2 Description of Dataset

In this research study, the UNSW-NB15 dataset is used which is kept on google drive. The UNSW-NB-15 dataset is generated by the **IXIA PerfectStorm** program in the cyber range lab of the **Australian centre for cyber security (ACCS)**. It combines modern, ethical activities with artificial and modern attack behaviour. The 100 GB of raw traffic data was captured by using Tcpdump by generating Pcap files. The dataset has nine different types of attack such as worms, shellcode, fuzzers, analysis, backdoor, denial-of-service, exploits and generic attacks. The UNSW-NB15_features.csv file has more information about all these features. It has four CSV files which includes **2,540,044 records** in the dataset. The actual records are stored in UNSW-NB15_GT.csv. The dataset is divided into training and testing set. 175,341 records are for training set and 82,332 records are for testing set which covers both normal and attack incidents for training and evaluating the model. Hence, the data's accurate feature of collection and comprehensive nature makes it perfect fit for building the network intrusion detection models (www.kaggle.com, n.d.).

5.3 List of Installed Libraries

Library	Objective	Functions
NumPy	Numerical Operation	It supports mathematical function and array operations, which are critical for the activities which involve managing of data and mathematical computation.
Pandas	Data Analysis and manipulation	It is required for dataset loading, cleaning, transformation, and data analysis. It allows us to manage and execute complex data efficiently.
seaborn	Making graphs using statistics	It is used for creating good statistical graphs.
Matplotlib	Data plotting	This is used for creating basic bar plots, graphs, and charts
Scikit-learn	Evaluation of models and Machine Learning	This library provides an enormous selection of tools for feature selection, preprocessing, model evaluation and training. The key elements are as follows: **“ensemble”** is used to implement adaboost, gradient boosting and other techniques. **“Feature selection”** is used for selecting critical features and **“model selection”** to split the data into training and testing. **“metrics”** such as confusion matrix, classification report and accuracy to evaluate the model’s performance. **“pre-processing”** is used to transform the data like label encoding and scaling.
imblearn (imblanced-learn)	Imbalanced dataset handling	It is necessary to enhance the model’s performance on minority classes, this library is important for balancing the class distribution in the dataset by using oversampling techniques such as **SMOTE (synthetic minority oversampling technique)** (GitHub, 2020).
mlxtend	It is used for stacking classifier	This library is used for stacking ensemble models which will combine the predictions from different base classifiers.

Pickle	Serialization and Deserialization	This library is used in which the python objects can be serialized into byte streams and deserialized them into python objects.

Table 2: Python Libraries (parthmanchanda81, 2021).

5.4 Data Pre-processing and Feature Selection

The pre-processing stage is said as the most essential stage to make sure that the dataset is prepared for training machine learning models. Initially, the dataset is first imported into Pandas Dataframes, which will offer a simple structure for processing the data, from the google drive. The data cleaning is the next stage in which we need to ensure that only clean and relevant data is used for analysis to handle any missing values and eliminate the unwanted characteristics. To encode the categorical variables the Scikit-learn Label Encoder is utilized, which includes different attack types, converting the text labels into the numerical values which can be processed by machine learning techniques. To enhance thee performance and accuracy of the models and to put numerical features on a common scale, the technique used are standardization or normalizing to scale the features.

Feature selection is a process of choosing the most appropriate characteristics from the dataset and it is very crucial for improving the models performance. To evaluate each feature's importance mutual information is utilized in this research. By mutual information we can determine the quantity of information obtained about one variable via another variable. Depending on this, a certain percentile of the top features with the highest mutual information scores is chosen using the select percentile method. This focuses on the most informative features which helps to improve the efficiency and accuracy of machine learning models by reducing the size of dataset and removing the redundant features.

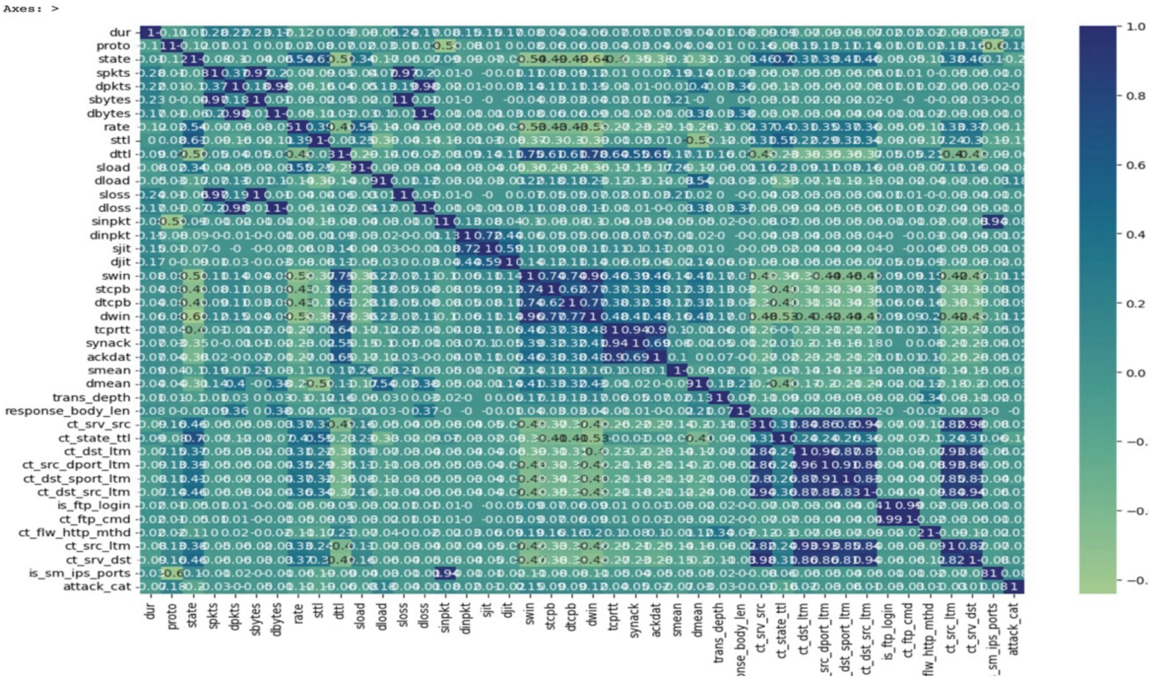


Figure 3: Correlation Matrix using Heatmap.

The above **Heat Map** shows a **correlation matrix** of the UNSW-NB15 datasets **features**. The correlation coefficient between two features is displayed using a certain range of values from -1 to 1. If the value is **1**, it is a **positive connection**, that states when one feature is raised the other is also likely to be raised and if the value is **-1**, it is a **negative connection**, which states when one feature rises the other falls. If the value is **0**, it is said as **no correlation**. Hence, the heat map makes its easier to find connection between different features which is very useful for feature selection and understanding the structure of the dataset.

5.5 Modelling

The Modelling stage is important stage in which we test and train the datasets. In the modelling stage, the pre-processed and balanced dataset was used to train the different machine learning models to accurately detect network intrusion. The ensemble learning methods such as Gradient boosting, XGboost and adaboost which are used because they enhance the overall performance by combining the advantages of different base models. This approach will increase the accuracy, decrease overfitting produce predictions what are strong. In my last model, the stacking classifier from **mlxtend** package is used for creating the **hybrid ensemble model**, which will improve the detection rate. In my hybrid model, gradient boosting is my meta-classifier, which is trained to improve the combined predictions when combined with other predictions made from the base classifier, AdaBoost and XGBoost. The overall accuracy and resilience of intrusion detection system is raised by using this stacking technique, by utilizing the advantages of each base models. To ensure the accurate evaluation of my model's performance I have determined it by using different measures such as accuracy, precision, recall, F1-score and confusion matrix.

6 Evaluation

In this section the overall output of my proposed Machine Learning models will be covered. The outcomes from each model will be compared by using accuracy, classification report and confusion matrix. The model's findings are displayed below as they are implemented, and their efficacy is examined. The Hybrid Ensemble model is the novel model which gave the best results for network intrusion detection.

Models	Test Accuracy	Precision	Recall	F1-Score
AdaBoost	49.55%	54%	50%	48%
XGBoost	81.53%	82%	79%	81%
Gradient Boosting	74.01%	75%	74%	74%
Hybrid Ensemble/Stacking Classifier	88.59%	90%	89%	89%

Table 3: Results of Machine Learning Models

For better understanding of all the results, the different attack categories and its count which are visualized throughout the network intrusion detection are categorised below.

Class	Types of Attack (UNSW-NB15 dataset)	Count
Class 0	Normal	37000
Class 1	Generic	18871
Class 2	Exploits	11132
Class 3	Fuzzers	6062

Class 4	DoS	4089
Class 5	Reconnaissance	3496
Class 6	Backdoor	583
Class 7	Shellcode	378
Class 8	Worms	44

Table 4: Class of UNSW-NB15 dataset

6.1 Case Study 1: AdaBoost Classifier

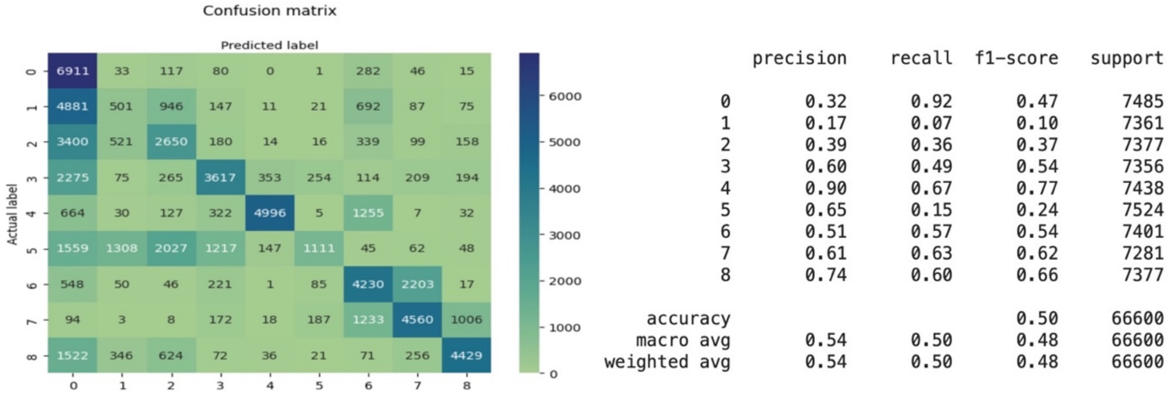
Accuracy for AdaBoost classifier in detecting the different attacks is 49.56%.

Confusion Matrix

The image below is showing the confusion matrix for AdaBoost classifier. The highlighted diagonal boxes are indicating the True positive (TP) values, which indicates a good correlation between the actual samples and predicated samples of AdaBoost trained machine learning model. 32,836 were classified correctly of the 66,600 samples, while 33,764 were incorrectly classified.

Classification Report

The below figure will display the detection score for precision, recall, f1 score for AdaBoost classifier.



Confusion matrix

Predicted label

Actual label

	0	1	2	3	4	5	6	7	8
0	6911	33	117	80	0	1	282	46	15
1	4881	501	946	147	11	21	692	87	75
2	3400	521	2650	180	14	16	339	99	158
3	2275	75	265	3617	353	254	114	209	194
4	664	30	127	322	4996	5	1255	7	32
5	1559	1308	2027	1217	147	1111	45	62	48
6	548	50	46	221	1	85	4230	2203	17
7	94	3	8	172	18	187	1233	4560	1006
8	1522	346	624	72	36	21	71	256	4429

```
              precision    recall  f1-score   support

           0       0.32      0.92      0.47      7485
           1       0.17      0.07      0.10      7361
           2       0.39      0.36      0.37      7377
           3       0.60      0.49      0.54      7356
           4       0.90      0.67      0.77      7438
           5       0.65      0.15      0.24      7524
           6       0.51      0.57      0.54      7401
           7       0.61      0.63      0.62      7281
           8       0.74      0.60      0.66      7377

    accuracy                           0.50     66600
   macro avg       0.54      0.50      0.48     66600
weighted avg       0.54      0.50      0.48     66600
```

Figure 4: Confusion Matrix and Classification Report of AdaBoost

6.2 Case Study 2: XGBoost Classifier

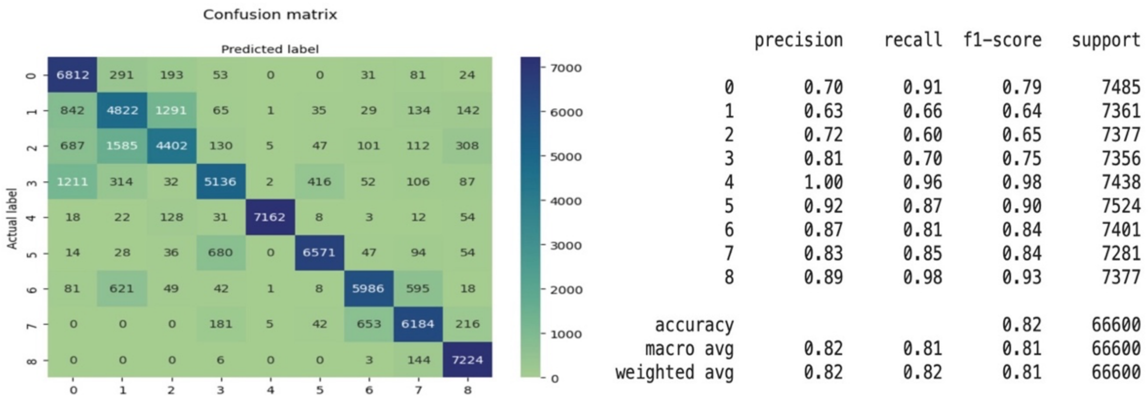
Accuracy for XGBoost classifier in detecting the different attacks is 81.53%.

Confusion Matrix

The image below is showing the confusion matrix for XGBoost classifier. The highlighted diagonal boxes are indicating the true positive (TP) values, which indicates a good correlation between the actual samples and predicated samples of XGBoost model. 54,201 were classified correctly of the 66,600 samples, while 12,399 were incorrectly classified.

Classification Report

The below figure will display the detection score for precision, recall, f1 score for XGBoost classifier.



	0	1	2	3	4	5	6	7	8
0	6812	291	193	53	0	0	31	81	24
1	842	4822	1291	65	1	35	29	134	142
2	687	1585	4402	130	5	47	101	112	308
3	1211	314	32	5136	2	416	52	106	87
4	18	22	128	31	7162	8	3	12	54
5	14	28	36	680	0	6571	47	94	54
6	81	621	49	42	1	8	5986	595	18
7	0	0	0	181	5	42	653	6184	216
8	0	0	0	6	0	0	3	144	7224

```
              precision    recall  f1-score   support

           0       0.70      0.91      0.79      7485
           1       0.63      0.66      0.64      7361
           2       0.72      0.60      0.65      7377
           3       0.81      0.70      0.75      7356
           4       1.00      0.96      0.98      7438
           5       0.92      0.87      0.90      7524
           6       0.87      0.81      0.84      7401
           7       0.83      0.85      0.84      7281
           8       0.89      0.98      0.93      7377

    accuracy                           0.82     66600
   macro avg       0.82      0.81      0.81     66600
weighted avg       0.82      0.82      0.81     66600
```

Figure 5: Confusion Matrix and Classification Report of XGBoost

6.3 Case Study 3: Gradient Boosting

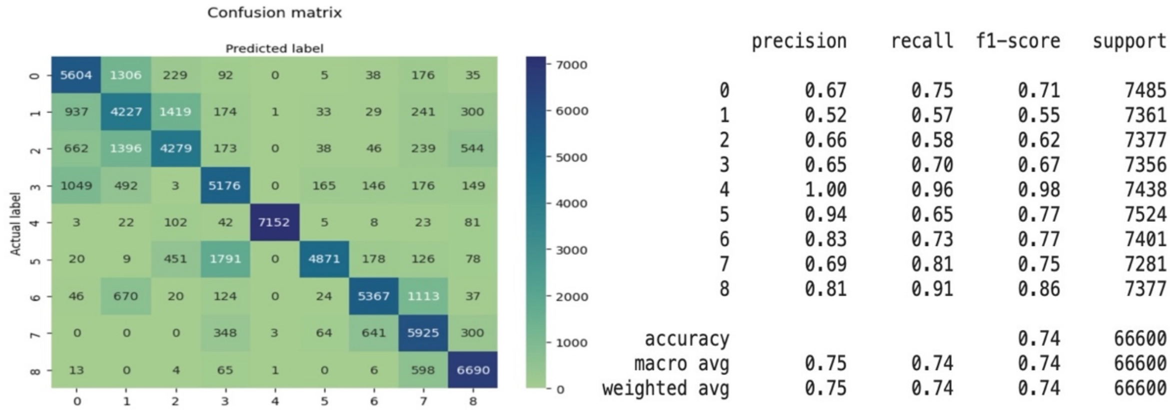
Accuracy for Gradient boosting in detecting the different attacks is 74.01%.

Confusion Matrix

The image below is showing the confusion matrix for Gradient Boosting. The highlighted diagonal boxes are indicating the true positive (TP) values, which indicates a good correlation between the actual samples and predicated samples of Gradient Boosting Model. 49,906 were classified correctly of the 66,600 samples, while 16,694 were incorrectly classified.

Classification Report

The below figure will display the detection score for precision, recall, f1 score for Gradient Boosting.



	0	1	2	3	4	5	6	7	8
0	5604	1306	229	92	0	5	38	176	35
1	937	4227	1419	174	1	33	29	241	300
2	662	1396	4279	173	0	38	46	239	544
3	1049	492	3	5176	0	165	146	176	149
4	3	22	102	42	7152	5	8	23	81
5	20	9	451	1791	0	4871	178	126	78
6	46	670	20	124	0	24	5367	1113	37
7	0	0	0	348	3	64	641	5925	300
8	13	0	4	65	1	0	6	598	6690

```
              precision    recall  f1-score   support

           0       0.67      0.75      0.71      7485
           1       0.52      0.57      0.55      7361
           2       0.66      0.58      0.62      7377
           3       0.65      0.70      0.67      7356
           4       1.00      0.96      0.98      7438
           5       0.94      0.65      0.77      7524
           6       0.83      0.73      0.77      7401
           7       0.69      0.81      0.75      7281
           8       0.81      0.91      0.86      7377

    accuracy                           0.74     66600
   macro avg       0.75      0.74      0.74     66600
weighted avg       0.75      0.74      0.74     66600
```

Figure 6: Confusion Matrix and Classification Report of Gradient Boosting

6.4 Case Study 4: Hybrid Ensemble Model/ Stacking Classifier

Accuracy for Stacking Classifier/Hybrid Ensemble Model in detecting the different attacks is 88.59%.

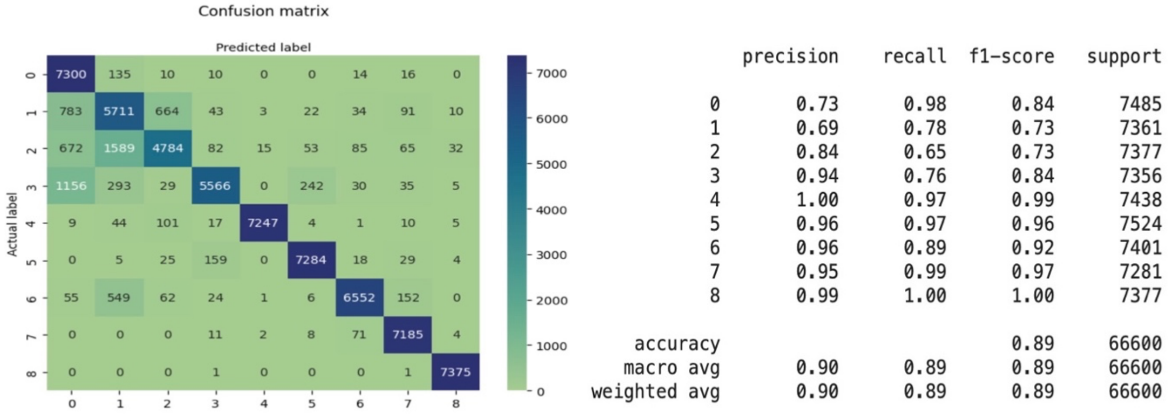
Confusion Matrix

The image below is showing the confusion matrix for Stacking Classifier/Hybrid Ensemble Model. The highlighted diagonal boxes are indicating the true positive (TP) values, which indicates a good correlation between the actual samples and predicated samples of Stacking

Classifier Model. 59,052 were classified correctly of the 66,600 samples, while 7,548 were incorrectly classified.

Classification Report

The below figure will display the detection score for precision, recall, f1 score for Hybrid Ensemble Model/Stacking Classifier.



	0	1	2	3	4	5	6	7	8
0	7300	135	10	10	0	0	14	16	0
1	783	5711	664	43	3	22	34	91	10
2	672	1589	4784	82	15	53	85	65	32
3	1156	293	29	5566	0	242	30	35	5
4	9	44	101	17	7247	4	1	10	5
5	0	5	25	159	0	7284	18	29	4
6	55	549	62	24	1	6	6552	152	0
7	0	0	0	11	2	8	71	7185	4
8	0	0	0	1	0	0	0	1	7375

```
              precision    recall  f1-score   support

           0       0.73      0.98      0.84      7485
           1       0.69      0.78      0.73      7361
           2       0.84      0.65      0.73      7377
           3       0.94      0.76      0.84      7356
           4       1.00      0.97      0.99      7438
           5       0.96      0.97      0.96      7524
           6       0.96      0.89      0.92      7401
           7       0.95      0.99      0.97      7281
           8       0.99      1.00      1.00      7377

    accuracy                           0.89     66600
   macro avg       0.90      0.89      0.89     66600
weighted avg       0.90      0.89      0.89     66600
```

Figure 7: Confusion Matrix and Classification Report of Hybrid Model

6.5 Discussion

In my research project, the Hybrid Ensemble Model scored the greatest accuracy of 88.59% among the tested models. This proved the advantages of ensemble model in managing imbalanced and complex datasets. The important understanding of identifying intrusions is obtained from this research project in which I have utilized a variety of ensemble learning models such as Gradient Boosting, AdaBoost, XGBoost and Stacking classifier which was the future work in few referenced papers. The limitations found while researching on different models are that the AdaBoost model had difficulty to handle the complexity even when I have used over sampling technique (SMOTE) to balance the dataset. The comprehensive approach to error reduction was offered by the Gradient Boosting Model but was lagged below XGboost. It needs few areas to be improved such as Hyperparameter tuning. XGBoost was the strongest amongst them showing a good accuracy. It was able to handle the classes and big datasets.

In my Hybrid Ensemble Model, the model misclassified 7,548 samples out of 66,600 which indicates that there is potential of improvement. After researching more to address this issue I was able to find that the Analysis class in the dataset was misleading the model, due to which it was impacting the overall performance. After removing that class the performance was slightly increased from 81% to 88%. I also found that adding more pre-processing stages, including sophisticated feature selection or extraction method can help me in improving my models performance. I tried using imbalanced dataset without using SMOTE technique, which gave me a good accuracy for all my models. In future improvement can be made by using deep learning techniques or any other ensemble technique. Even though the balanced data increases reliability, this shows that even in lack of data balancing the model still have potential. To guarantee generalizability stronger cross-validation procedures should be used even if the tests showed that ensemble technique is beneficial. As the depth of contextual evaluation is limited because of less research study available for direct assessment and comparison. The findings show the effectiveness of ensemble learning in intrusion detection and highlights the need for improvement. In the future research I should concentrate on optimizing model's parameters, investigating hybrid combinations with more advanced methods and testing results across a variety of datasets.

7 Conclusion and Future Work

The objective of this research project was to investigate the modern methods for network intrusion detection to develop a hybrid ensemble model using machine learning models. The research question of this study was how to identify network intrusions in flow-based data be improved by the combination of ensemble methods using oversampling technique. To answer the research question different machine learning algorithms were trained and evaluated on the UNSW-NB15 dataset using SMOTE oversampling technique to balance the data. In this study I implemented four algorithms they are Gradient Boosting, AdaBoost, XGBoost and Stacking Classifier/Hybrid Ensemble model. To identify the appropriate approach, a thorough analysis of every model's performance was conducted in this work.

The study showed that in comparison to individual models such as AdaBoost, XGboost and Gradient Boosting models, the hybrid ensemble model has greatly enhanced the detection of different attacks. The model gave an accuracy score of 88.59% and was able to identify different types of network attacks. The Gradient Boosting gave accuracy of 74.01% gave a balanced approach to error reduction. AdaBoost Algorithm gave 49.55% demonstrating a modest level of accuracy. The XGBoost gave a remarkable accuracy of 81.53% which demonstrated its usefulness for big and balanced datasets. This emphasize how crucial ensemble learning are for raising the precision and resilience.

Using SMOTE to balance the dataset the models performance was enhance. Even after achieving a good score the hybrid ensemble model failed to classify few samples correctly. These tells that it needs few improvements using different enhanced machine learning and ensemble learning methods. In future research I might concentrate on combining deep learning methods with ensemble model. Results from different dataset validation will add a contribution to establish a wider application. To improve the real time detection method, it may require an adaptive model that will always learn from fresh data. In the further investigation it might concentrate on optimizing model parameters, conduct tests in real world situations and investigate novel hybrid configurations to evaluate the performance.

8 Acknowledgement

I would highly like to appreciate my supervisor, Mr Eugene McLaughlin, for his outstanding assistance and support during this journey. The way he motivated me to make sure that I have improved my research performance. His prompt advice and constant encouragement enabled me to have a deeper comprehension of the research issue and to grasp it better. I truly value his guidance and to motivate me to finish my project on scheduled time.

References

Adegboyega, T.A. (2024). *A HYBRID MACHINE LEARNING MODEL FOR NETWORK INTRUSION DETECTION*. [online] Available at: http://dx.doi.org/10.13140/RG.2.2.24355.26405.

Almomani, A., Akour, I., Manasrah, A., Almomani, O., Alauthman, M., Abdullah, E., Shwait, A. and Sharaa, R. (2023). Ensemble-Based Approach for Efficient Intrusion Detection in Network Traffic. *Intelligent Automation & Soft Computing*, [online] 37(2), pp.2499–2517. doi:https://doi.org/10.32604/iasc.2023.039687.

Gouveia, A. and Correia, M. (2020). Network Intrusion Detection with XGBoost. *Chapman and Hall/CRC eBooks*, pp.137–166. doi:https://doi.org/10.1201/9780429270567-6.

Husain, A., Salem, A., Jim, C. and Dimitoglou, G. (2019). *Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ISSPIT47144.2019.9001867.

Kiflay, A.Z., Tsokanos, A. and Kirner, R. (2021). A Network Intrusion Detection System Using Ensemble Machine Learning. *2021 International Carnahan Conference on Security Technology (ICCST)*. doi:https://doi.org/10.1109/iccst49569.2021.9717397.

Kocher, G. and Kumar, G. (2020). Performance Analysis of Machine Learning Classifiers for Intrusion Detection using UNSW-NB15 Dataset. doi:https://doi.org/10.5121/csit.2020.102004.

Mashuqur Rahman Mazumder, A.K.M., Mohammed Kamruzzaman, N., Akter, N., Arbe, N. and Rahman, M.M. (2021). Network Intrusion Detection Using Hybrid Machine Learning Model. *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. doi:https://doi.org/10.1109/icaect49130.2021.9392483.

Md. Alamin Talukder, Md. Manowarul Islam, Md Ashraf Uddin, Khondokar Fida Hasan, Sharmin, S., Alyami, S.A. and Mohammad Ali Moni (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data*, 11(1). doi:https://doi.org/10.1186/s40537-024-00886-w.

More, S., Moad Idrissi, Mahmoud, H. and A. Taufiq Asyhari (2024). Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis. *Algorithms*, 17(2), pp.64–64. doi:https://doi.org/10.3390/a17020064.

Ngamba Thockchom, Moirangthem Marjit Singh and Nandi, U. (2023). A novel ensemble learning-based model for network intrusion detection. *Complex & Intelligent Systems*. doi:https://doi.org/10.1007/s40747-023-01013-7.

Verma, P., Anwar, S., Khan, S. and Mane, S.B. (2018). *Network Intrusion Detection Using Clustering and Gradient Boosting*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICCCNT.2018.8494186.

Weiming Hu, Wei Hu and Maybank, S. (2008). AdaBoost-Based Algorithm for Network Intrusion Detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(2), pp.577–583. doi:https://doi.org/10.1109/tsmcb.2007.914695.

Zhang, P., Jia, Y. and Shang, Y. (2022). Research and application of XGBoost in imbalanced data. *International Journal of Distributed Sensor Networks*, 18(6), p.155013292211069. doi:https://doi.org/10.1177/15501329221106935.

Zoghi , Z. and Serpen, G. (2022). Ensemble Classifier Design Tuned to Dataset Characteristics for Network Intrusion Detection. [online] DeepAI. Available at: https://deepai.org/publication/ensemble-classifier-design-tuned-to-dataset-characteristics-for-network-intrusion-detection.

GitHub. (2020). scikit-learn-contrib/imbalanced-learn. [online] Available at: https://github.com/scikit-learn-contrib/imbalanced-learn.

parthmanchanda81, G. (2021). Libraries in Python. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/libraries-in-python/.

www.kaggle.com. (n.d.). UNSW_NB15. [online] Available at: https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15.

Dhaliwal, S., Nahid, A.-A. and Abbas, R. (2018). Effective Intrusion Detection System Using XGBoost. Information, 9(7), p.149. doi:https://doi.org/10.3390/info9070149.

Hany Abdelghany Gouda, Mohamed Abdelslam Ahmed and Mohamed Ismail Roushdy (2023). Optimizing anomaly-based attack detection using classification machine learning. Neural Computing and Applications. doi:https://doi.org/10.1007/s00521-023-09309-y.

Martindale, N., Ismail, M. and Talbert, D.A. (2020). Ensemble-Based Online Machine Learning Algorithms for Network Intrusion Detection Systems Using Streaming Data. Information, 11(6), p.315. doi:https://doi.org/10.3390/info11060315.

Yerima, S.Y. and Sezer, S. (2019). DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection. IEEE Transactions on Cybernetics, 49(2), pp.453–466. doi:https://doi.org/10.1109/tcyb.2017.2777960.

Nvidia (n.d.). What is XGBoost? [online] NVIDIA Data Science Glossary. Available at: https://www.nvidia.com/en-us/glossary/xgboost/.

rasbt.github.io. (n.d.). StackingClassifier: Simple stacking - mlxtend. [online] Available at: https://rasbt.github.io/mlxtend/user_guide/classifier/StackingClassifier/.

Saini, A. (2021). Gradient Boosting Algorithm: A Complete Guide for Beginners. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/#:~:text=our%20test%20data.-.

www.datacamp.com. (n.d.). AdaBoost Classifier Algorithms using Python Sklearn Tutorial. [online] Available at: https://www.datacamp.com/tutorial/adaboost-classifier-python.

Brownlee, J. (2020). SMOTE for Imbalanced Classification with Python. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/.

Maklin, C. (2022). Synthetic Minority Over-sampling TEchnique (SMOTE). [online] Medium. Available at: https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c.

Kausar, R.A. (2023). Market Basket Analysis Using MLxtend Library in Python. [online] Medium. Available at: https://medium.com/@rifatalkausar/market-basket-analysis-using-mlxtend-library-in-python-7e56bd41a569.