Enhancing Network Security Using Machine Learning Model-Agnostic Approach on Diverse Datasets

MSc Research Project
Data Analytics

Muhammad Zaeem
Student ID: x23108088

School of Computing
National College of Ireland

Supervisor: Arjun Chikkankod

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Muhammad Zaeem

Student ID: X23108088

Programme: MSc Data Analytics **Year:** 2024

Module: MSc Research Project

Supervisor: Arjun Chikkankod

Submission Due Date: 12th August,2024

Project Title: Enhancing Network Security Using Machine Learning Model-Agnostic Approach on Diverse Datasets

Word Count: 8000+ **Page Count** 20 pages

I hereby certify that the information contained in this (my submission) is information about 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: Muhammad Zaeem

Date: 12th August, 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 Network Security Using Machine Learning Model-Agnostic Approach on Diverse Datasets

Muhammad Zaeem
X23108088

Abstract

With the evolution of technology, everything has moved to online channels making it easier for hackers and other cyber threats to make intrusion attacks and cause the breach of personal or official data. This has increased the demand for more secure and updated intrusion detection systems (IDS). Firewalls serve as the first line of defence against cyber-attacks however, their signature-based architecture fails to defend against newer attacks. This study proposes using machine learning algorithms with current firewalls to increase network classification and intrusion detection efficiency. The research incorporates three different network classification and intrusion detection datasets that are analysed using a model-agnostic pipeline comprising of logistic regression, naïve bayes, random forest, XGBoost, KNN, SVM and ANN while exploring challenges like outlier analysis, data imbalance and feature diversity. The Grid Search algorithm is used for the efficient hyperparameter tuning of all the models. All models' results are comparatively evaluated to find the best machine learning model to deal with the intrusion problem. The performance of all models was found to increase after oversampling of minority classes, feature selection and hyperparameter tuning of models. Random Forest and XGBoost emerged as the best-performing models with an F1-score of 0.99 on two datasets and 0.80 on the third dataset.

1 Introduction

1.1 Background

In the present world, almost all tasks involve the use of computer systems connected to online platforms and internet networks to provide easy access to resources and spend less time performing certain tasks. Individuals and organisations use different types of databases and clouds to store their data, which is accessed using internet servers. In our daily lives, we allow multiple phone or web applications to access our storage and other system resources for the more optimised working of applications which puts our private data at the stake of being stolen and used for any unethical purposes. Data being accessible via the internet makes it prone to any expected or unexpected attacks (Manoj et al., 2021). That’s why the security of this data is a very difficult task. The importance of this problem can be realised from the fact that to deal with these threats a whole new field of cyber security has been introduced in which individuals are trained to tackle such attacks. However, there is a need for automated mechanisms to handle such problems and avoid the loss of any personal or official data.

1.2 Research Gap

Internet firewalls act as the first line of defence against any potential intrusions. Still, the problem is that most of the firewalls work on predefined rule-based algorithms for network classification and to prevent potential attacks (Md Habibur Rahman et al., 2023). The static nature of firewalls makes them strong against known threats. While being effective to some extent, the problem with such rule-based algorithms is that they are not capable of protecting against any new or modern attacks done with the help of artificial intelligence (AI), as seen in the tragic event of the WannaCry Ransomware attack that affected thousands of computers (Hsiao and Kao, 2018). These AI-based attacks are more difficult to defend against as they use algorithms that are too strong for traditional firewalls to handle making them more capable of causing data breaches (Aswal et al., 2021). However, it is a complex process to increase the defensive mechanism of these firewalls as there might be hundreds or thousands of functions and rules defined in sequence to make it work (Lin and Yao, 2022). Changing these rule-based algorithms manually for each new breach or threat is difficult and more time-consuming, so there is an urgent need for some easy-to-implement yet more intelligent security solutions.

Machine learning (ML) emerges as an effective tool to handle these challenges. By using ML, it is possible to develop such solutions that can train on large amounts of data, identify distinct patterns and make informed decision-making for network traffic classification and intrusion detection while minimising the risk of attacks in real-time. If such ML algorithms are developed to work along with the current firewalls, the protective capability of firewalls would be increased to detect new intrusions. However, it is not possible to tackle all the anomalies by just implementing any random ML algorithm as each algorithm works differently with different types of data. Machine learning algorithms in their general form might not give the best results when implemented on intrusion data. There are specific preprocessing tasks that need to be carried out according to the nature and conditions of the data as different datasets have different sizes, number of classes(attacks) and feature diversity. Similarly, models are to be applied and evaluated iteratively based on the predictions made by them until they give the best possible solution

1.3 Research Question and Objectives

In our research we aim to find the answer to the following research questions:

- How do variations in dataset size, feature selection/importance, and class imbalance affect the predictive performance of machine learning models in network intrusion detection and network traffic classification?
- Which machine learning algorithm is most reliable to be suggested for real-life integration with firewalls for network traffic classification and Intrusion Detection?

These questions are aimed at exploring the efficiency of different machine-learning techniques for enhanced network security by detecting and classifying network attacks while considering the constraints of different datasets.

For this purpose, we will be using three different datasets, one for simple network classification and two datasets for intrusion detection taken from Kaggle and UCI Machine Learning Repository. Each dataset has different types of records, number of features, instances and number of classes in the dependent column. After that, the Model-Agnostic approach will be used which reflects creating an environment suitable for all models and not specifically favourable to one. A model pipeline will be created to apply different ML models like logistic regression, naïve bayes, random forest, XGboost, K-nearest neighbour (KNN), SVM and ANN. This will allow us to compare machine learning, ensemble learning, and neural network models without a bias towards specific type models.

The most focus will be placed on data preprocessing, feature engineering and evaluation metrics instead of putting all focus on one specific model. The goal is to make sure that the process works with any chosen model. Hyperparameter tuning of models will also be done to improve their overall performance.

The Structure of the report is as follows: Section 2 covers the literature review of related works highlighting their contributions and gaps. Section 3 covers the methodology used for data collection, data preparation, model implementation and metrics used for evaluation. Section 4 describes the model-agnostic approach, model selection and implementation structure across different datasets. Section 5 covers the evaluation of the performance of models on imbalanced and balanced datasets after hyperparameter tuning along with the discussion of the results from different steps. Lastly, Section 6 concludes the whole research by summarizing the key findings, and limitations, and proposes directions for future work.

2 Related Work

Internet traffic classification and intrusion detection has been an important research area for the past few decades. A lot of work has been done to enhance network security and prevent the loss of important data. Here we will be reviewing some of the studies conducted in the past regarding network security.

2.1 Need for Enhanced Security Measures

The survey paper by Kishan et al. studied the limitations of traditional firewalls in handling modern and more complex cyber-attacks highlighting the need for updated Next-Generation Firewalls (NGFW) that would have advanced technologies like deep packet inspection and intrusion prevention systems (Neupane, Haddad and Chen, 2018). In simple words, the paper supports the authenticity of our research problem that traditional firewalls are not strong enough to keep up with new attacks so there is a need to introduce new versions of firewalls with updated technology. Similarly, another research on the limitations of intrusion detection for network-based and host-based systems highlighted the issues of traditional intrusion detection systems (IDS) with manual classification being time-consuming, repetitive and producing high false alarms (Samrin and Vasumathi, 2017). The research acknowledges the capability of data mining techniques like clustering and classification in exploring large datasets and proposes that hybrid systems made by the combination of statistical, machine learning and knowledge-based detection methods can provide high detection accuracy while minimising the time consumed.

Applebaum et al. (2021) in their paper review two different Web Application Firewalls (WAFs) and their ability to protect web applications against attacks. They compared signature-based WAFs and machine learning-based WAFs and analysed that signature-based WAFs use specific rules to stop malicious traffic but require continuous improvement and high skills to manage them against new attacks which makes it difficult for them to handle zero-day attacks. As compared to that machine learning-based WAFs can easily adapt to new threats and are comparatively easier to configure and maintain. However, they highlight the lack of real-world testing and benchmark comparisons of machine learning-based WAFs and the need to assess their effectiveness against new attacks. To support their idea, Appelt et al. (2018) proposed the integration of machine learning with evolutionary algorithms to test and enhance the security of WAFs against SQL-injection attacks. They developed different variants of ML-driven approaches from which the ML-driven E variant that consisted of random tree and random forest algorithm outperformed all other variants in attack detection. However, the paper's focus only on SQL-injection attacks makes the performance of their proposed approach questionable for handling other types of attacks. In their future work, they proposed to address the challenges of data imbalance and attack variability in a broader context.

While studying the application of machine learning for enhancing IDS, Haripriya and Jabbar (2018) studied different machine learning algorithms like random forest, naïve bayes, K means and average one dependence estimator (AODE) and highlighted their importance in reducing false alarms and increased detection rate. The paper concluded that machine learning shows clear improvement over traditional IDS approaches. However, they admitted the lack of a sufficient amount of data for better training of the models.

The research paper by Yang et al., (2022) which is a systematic literature review of 119 papers provides detailed information about the present state of intrusion detection. Multiple datasets and techniques used for intrusion detection were analysed in this paper including the preprocessing steps, attack detection techniques and evaluation metrics. The study showed the need for improved preprocessing and evaluation methods while highlighting the importance of integrating machine learning into IDS. For future work, they suggested more focus on dealing with data imbalance issues. This research acts as a strong guide for future research as it details the current situation of IDS while highlighting areas that need future improvement.

2.2 Machine learning models

The research paper by Subba, Biswas and Karmakar (2015) used linear discriminant analysis (LDA) and logistic regression for anomaly-based intrusion detection based on the NSL-KDD dataset and compared their results with models like naïve bayes, C4.5, and SVM. They claimed their proposed models gave better accuracy and anomaly detection rate than more complex models like SVM. However, their research fails to cover important preprocessing steps like handling class imbalance and the model's performance on balanced data compared to unbalanced data. Similarly, another study based on the wireless sensor network was done using the KNN algorithm to detect intrusion attacks such as DoS, replay, integrity and flooding (Li et al. 2014). The results showed the KNN model to be performing well and giving higher accuracy, however, the study lacks the information about preprocessing of data and comparisons of KNN with other models to better interpret the performance of KNN.

Compared to that, Ding et al. (2022) explained in detail the challenge of class imbalance in intrusion detection and proposed a new approach TACGAN, a hybrid of KNN and Generative Adversarial Network (GAN) models, that uses KNN for under-sampling the normal class and TACGAN for oversampling of attack class variables to make the dataset balanced. The research included the use of three real-world datasets and their proposed approach showed clear improvements in accuracy, recall and F1 scores after data balancing. The study suggested future integration of deep learning and clustering algorithms for a more enhanced detection rate.

A paper on NIDS (network intrusion detection system) explored the working of discriminative multinomial Naïve Bayes (DMNB) along with various filtering techniques for the development of a strong NIDS based on the NSL-KDD dataset which is an improved version of the KDDCup 1999 dataset (Panda, Abraham and Patra, 2010). They performed binary classification of connections as normal or attack and validated their model through 10-fold cross-validation. The results showed that the DMNB model when used with Nominal to Binary Supervised Filtering, outperformed other methods like decision tree, random forest and SVM. However, the study lacks model evaluation on other datasets with multiclass classification problems. To explore the working of naïve bayes for multiclass classification, (Koc et al. 2012) presented a Hidden Naïve Bayes model that overcomes the limitations of conditional independence among features faced by previous naïve bayes approaches. The results showed their proposed model gave better detection accuracy than SVM and traditional naïve bayes models. However, the study's dependence on the KDDCup 1999, an old dataset, makes the model's performance questionable for updated datasets.

The Limitations of the KDDCUP99 dataset were addressed by Jing et al. (2019) who highlighted a more comprehensive and modern UNSW-NB15 dataset that could be used for both binary and multiclass intrusion detection. They proposed an enhanced SVM using a non-linear scaling method, that showed improved accuracy and detection rate as compared to different linear and tree-based models by achieving an accuracy of 85.99% for binary class and 75.77% accuracy for multiclass classification. The study demonstrates the feature importance of the UNSW-NB15 dataset and the efficiency of enhanced SVM for network security. Similarly, another research paper highlighted the limitations of outdated datasets while opting for a relatively better Kyoto 2006+ dataset and proposed using the J48 decision tree for detecting cyber-attacks (Sahu and Mehtre, 2015). Using a 10-fold cross-validation method, the J48 decision tree achieved an accuracy of 97.2% and a high true-positive rate for attacks. However, the research lacks the comparative analysis of other models with the J48 model to better evaluate it.

2.3 Neural Networks

The article by Ahmad et al. (2020) compared different machine learning and deep learning methods for IDS and claimed deep learning models such as Autoencoders (AE), Deep Neural Networks (DNN), CNN and RNN tend to perform better than the traditional ML models. However, neural networks need high computational resources and face difficulty handling outdated datasets. The study highlights the need for more updated and balanced datasets for easier detection of low-frequency attacks. Similarly, a Journal on the importance of data balancing analysed different data resampling techniques on the performance of the ANN classifier with respect to KDD99, UNSW-NB15, UNSW-NB17, and UNSW-NB18 cyber security datasets (Bagui and Li, 2021). The study concludes that oversampling data increases the training time but also improves the recall score by correctly classifying the minority classes. The paper suggests using appropriate resampling techniques based on the level of data imbalance.

A new dataset, NF-UQ-NIDS-v2 was used by Gouda et al. (2023) in their research to enhance anomaly-based attack detection using both classical ML and neural network models. Their Random Forest achieved a high accuracy of 99.07% followed by 98.87% by the Long Short-Term Memory (LSTM) Model and 98.56% by CNNs. However, they overlooked the idea of having equal class importance for anomalies and some attacks remained undetected due to their low frequency. They suggested future work to address this detection gap.

A hybrid model named DLNID was proposed by Yanfang Fu et al. (2022) to handle the issues of data imbalance and low attack detection accuracy. The model used CNN for sequence feature extraction, attention mechanism to do weighted focus on important features, Bidirectional LSTM (Bi-LSTM) to learn feature sequence for better predictions, and Adaptive Synthetic Sampling (ADASYN) for minority class augmentation to handle data imbalance. Additionally, a stacked autoencoder is used for dimensionality reduction and enhancing information fusion. When tested on the NSL-KDD dataset, the DLNID model outperformed all other models, achieving an accuracy of 90.73% and an F1 score of 89.65%. The paper suggested the implementation of the proposed model in real-time network environments. Similarly, Shone et al. introduced Nonsymmetric Deep Autoencoders (NDAE) for unsupervised feature learning and deep learning classification made by combining stacked NDAEs with random forests. Their model achieved 98.81% accuracy which was almost 5% higher than the deep belief networks (DBNs) (Shone et al. 2018). However, for zero-day attack detection and real-world implementation future work is suggested.

2.4 Ensemble Learning Models

Farnaaz and her colleague used random forest on the NSL-KDD dataset achieving relatively higher detection and lower false alarm rates when compared to its accuracy with the J48 classifier (Farnaaz and Jabbar, 2016). The research highlights the importance of symmetrical uncertainty for dimensionality reduction and removing irrelevant features. Similarly, Sharma et al. (2021) proposed a heterogeneous voting model using algorithms like KNN, logistic regression, SVM, decision tree and stochastic gradient descent along with an ensemble stacking model using the random forest as the main classifier for efficient packet filtering and network classification of internet firewall log data for network connections handling tasks such as accept, deny, drop or reset-both classes. Their ensemble stacking model achieved a precision of 91% and an accuracy of 99.8% outperforming all other models.

A paper on intrusion detection introduced a Multitree algorithm made by combining multiple decision trees, random forests, KNNs and DNNs in an adaptive voting algorithm that detected intrusion based on the contributed voting of all models achieving a final accuracy of 85.2% (Gao et al., 2019). The study concluded the practical importance of this adaptive ensemble learning approach over traditional models. Similarly, the paper by Das et al. (2021) uses a novel ensemble feature selection (EnFS) technique made by the combination of multiple feature selection methods that enhance feature optimisation. Those features are then passed to an ensemble framework incorporating multiple ML classifiers. The study claims that ensemble models outperform individual classifiers achieving a detection rate of

99.3% and a low false alarm rate of 0.5%. The paper suggested for future validation, new datasets need to be explored using the same technique.

The paper by Jiang et al. (2020) proposed a PSO-XGBoost model using particle swarm optimisation for hyperparameter tunning of the XGBoost Algorithm. When applied to the dataset, their model showed better performance as compared to other ensemble models like the random forest, and Adaboost, especially in identifying the minority attacks. However, for the higher number of particles and iterations to find the global optimum solution the model demands more computational time that is suggested to be addressed in future.

2.5 Limitations and Takeaways from Literature

The presence of so much previous work done for network traffic classification and intrusion detection emphasizes the importance of our area of research and indicates that this problem is worth exploring. From the literature, we found that most studies were targeted towards a single type of attack detection i.e. binary classification. However, with the updated technologies, new attacks are being done demanding the development of models capable of multiclass intrusion detection. Another shortcoming from related work shows that most research is targeted towards the final model implementation and improvement while overlooking the importance of preprocessing the data. Some studies have highlighted the problem of class imbalance to be looked at in future work. So, in our research, we will be doing an in-depth analysis of class imbalance on the predictive capability of models and evaluation metrics while looking into the strategies to provide equal class importance to all anomalies and compare the results with the base case.

Most studies from subsections 2.2 and 2.3 have discussed the shortcomings of outdated datasets and demanded more updated datasets to carry out more adaptable research, in correspondence to this we will use relatively newer datasets, published within the last decade to ensure the authenticity of our research. Lastly, most research in the literature focussed on the improvement of their one proposed model only and compared it to the base case of other models, however, we will not be favouring any single model in our research and rather apply different machine learning, ensemble learning or neural network models in a model agnostic environment while improving them all to determine the best model avoiding biases.

3 Research Methodology

The methodology followed in our research is the KDD (Knowledge Discovery in Databases) methodology for machine learning. The KDD methodology consists of different steps arranged in a sequenced process starting from data selection and ending at the extraction of useful knowledge with several data processing steps in between. This methodology is suitable for our network classification and intrusion detection research due to its structured data analysis and model evaluation approach.

Figure 1 shows the steps involved in our KDD methodology. The detailed implementation of these steps for our proposed models on the network classification and intrusion detection datasets can be seen below.

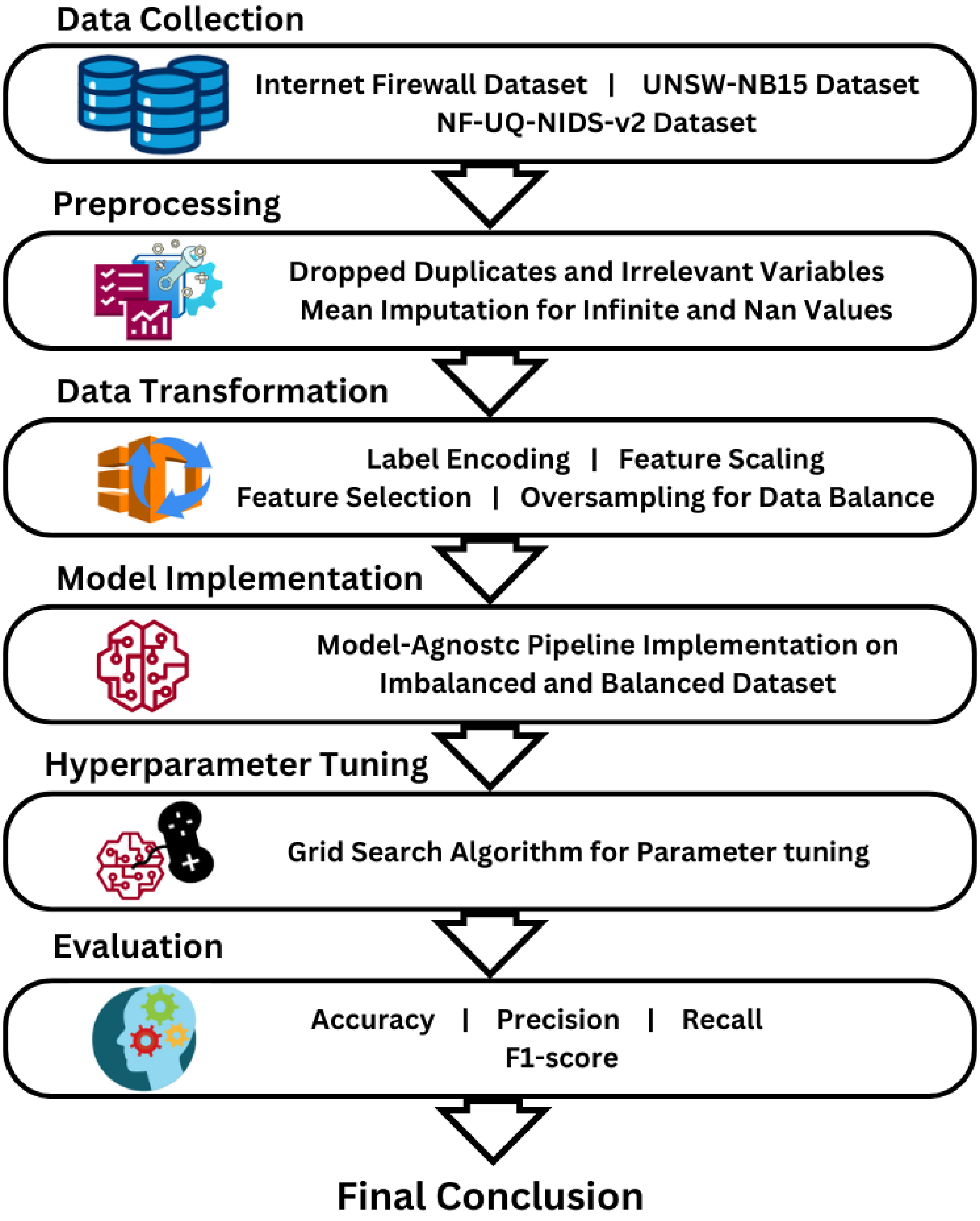
Data Collection
Internet Firewall Dataset | UNSW-NB15 Dataset
NF-UQ-NIDS-v2 Dataset
Preprocessing
Dropped Duplicates and Irrelevant Variables
Mean Imputation for Infinite and Nan Values
Data Transformation
Label Encoding | Feature Scaling
Feature Selection | Oversampling for Data Balance
Model Implementation
Model-Agnostc Pipeline Implementation on
Imbalanced and Balanced Dataset
Hyperparameter Tuning
Grid Search Algorithm for Parameter tuning
Evaluation
Accuracy | Precision | Recall
F1-score
Final Conclusion

Figure 1: Research Architecture

3.1 Datasets Collection

For the implementation of our proposed study, we will be using three different CSV datasets out of which one dataset is for network classification and two datasets are for network intrusion detection. To overcome the shortcomings of old datasets highlighted in the literature, we used only the datasets that were published within the last decade. The following are the three datasets used in our research.

3.1.1 Internet Firewall Dataset (IFD)

It is a dataset taken from the UCI machine-learning repository[1] which is an open-source repository that allows their datasets to be used for research purposes. The dataset is made from internet traffic records of a university's firewalls. It consists of 12 features and 65,000+ instances. Its dependent variable "Action" has four classes: allow, deny, drop and reset.

3.1.2 UNSW-NB15 Dataset

It is a dataset taken from Kaggle[2] available under the user licence CC BY-NC-SA 4.0 that allows to share and build upon work for noncommercial purposes as long the citation to the source is provided which is a paper by the UNSW University (Moustafa and Slay, 2015). This data set comprises millions of records however for our research we will use its testing set which comprises 175,000+ records and 45 features. Its dependent variable has about 10 attack classes such as 'Backdoor', 'Analysis', 'Fuzzers', 'Shellcode', 'Reconnaissance' etc. that will be used in our research to train our models.

3.1.3 NF-UQ-NIDS-v2 Dataset

This dataset is also taken from Kaggle[3] with a 10.00 usability rate and is available for public use under a CC0: Public Domain licence that allows this dataset to be used for any personal or commercial purpose without any restriction or copyright claims. This dataset consists of 43 features and more than 10 million records out of which we will be considering the first 300,000 records that consist of almost 20 types of attack classes.

3.2 Exploratory Data Analysis (EDA)

The dataset CSV files stored on the local storage are imported into Jupyter Notebook IDE for data exploration. During EDA, basic attributes such as the shape of data, data types of variables, presence of outliers, missing values and duplicate records are analysed. In our case, all our datasets had different numbers of features and instances with two datasets IFD and NF-UQ-NIDS-v2 having all the numeric variables and only the dependent variable having categorical data while UNSW-NB15 datasets had four categorical variables. During EDA, we also checked for the missing values in our datasets and found none. At this step, we made initial visualisations such as Boxplots for outliers' analysis, heatmap to check the correlation of numeric variables and bar plot to check the class distribution of dependent variables. From these visualisations, we found that there were some outliers in the datasets and the dependent variable classes were highly imbalanced.

3.3 Data Preprocessing

Based on EDA, we performed the initial preprocessing of the data to remove irrelevant data features and instances to make it more suitable for our research, such as the Internet Firewall dataset had some duplicate values that we simply dropped from the dataset as the

[1] https://archive.ics.uci.edu/dataset/542/internet+firewall+data
[2] https://www.kaggle.com/datasets/dhoogla/unswnb15
[3] https://www.kaggle.com/datasets/aryashah2k/nfuqnidsv2-network-intrusion-detection-dataset

presence of these duplicates influence the overall working of models and lead to biased predictions.

3.3.1 Removing Irrelevant Features

There were some irrelevant features in datasets such as the UNSW-NB15 Dataset had a whole column of the index named 'id' and the NF-UQ-NIDS-v2 Dataset had a column with names of different small datasets that were combined to generate that big dataset, based on domain knowledge these features had no influence on predictions so removing such features helps decrease the training time for models.

3.3.2 Ethical Considerations

The NF-UQ-NIDS-v2 Dataset had 2 features comprising of the IP addresses of connected network devices that were removed due to ethical considerations of IP addresses being personal data that should not be made public and can put individuals at potential risk (Lundevall and Tranvik, 2010). The reason for removing rather than anonymizing it that IP addresses do not play any role in threat predictions and can also be used to trace or identify the individuals or organisations making them prone to further harms. So it was in best interest to remove it for avoiding the unauthorized exposure of sensitive information during research.

3.3.3 Outliers Analysis

Outliers were analysed using boxplots during EDA. Initially, we removed these outliers but later, we realised removing outliers caused the loss of data related to some attacks, as these outliers are specific to certain attacks as analysed by Jabez and Muthukumar (2015) who proposed an IDS only based on detecting outliers in the data. So, we decided to keep these outliers in the dataset. Only the extreme outliers having too large values for the model were handled using the mean imputation method.

3.3.4 Handling Infinite and Nan Values

During the early implementation of models, they could not process the data and gave errors indicating the presence of some Infinite or Nan values that were not detected while checking for missing values or outliers. So, to remove these infinite values, initially, we replaced them with nans and then we imputed nans with the mean values of the columns.

3.4 Data Transformation

Transformation of data into a form that is suitable for ML models is performed in this step including label encoding of categorical variables, feature engineering, feature scaling, normalization etc.

3.4.1 Handling Categorical Data

Some machine-learning models such as logistic regression, SVM, KNN and ANN struggle with handling categorical data So, for the efficient working of these models, the categorical data is encoded into numerical using LabelEncoder from the sklearn preprocessing library.

3.4.2 Feature Scaling

Feature scaling is crucial for the efficient performance of ML models like logistic regression, SVM and KNN as they are affected by the features having large numerical values that can dominate the overall learning process. Ensemble models like random forest and XGBoost can handle features with varying values however scaled features can also contribute to the improvement of their performance. We used Standard Scaler in our model's pipeline that uses z-score standardization to scale all the features to have a mean of 0 and a standard deviation of 1 ensuring that all features contribute equally to the training of models. This standardization process can decrease the training time and improve performance for models sensitive to feature scales.

3.4.3 Handling Class Imbalance

As analysed in the literature, class imbalance is a major issue in network traffic datasets as the number of records for normal traffic is higher than the intrusion records. Similarly, our datasets had the same problem of class imbalance. Class imbalance can highly influence the performance of models that will be discussed in coming sections of the report. It is important to make the dataset balanced either by under-sampling the majority classes to the level of minority classes or by over-sampling the minority classes to the level of the majority class.

For our specific problem, we used the oversampling technique influenced by the research of Mohammed, who concluded that compared to under-sampling, oversampling tends to perform better for different classifiers and gives high scores for different evaluation metrics (Mohammed, Rawashdeh and Abdullah, 2020). Another reason we chose oversampling is because some of our minority intrusions had as low as 5 or 10 records, so under-sampling the whole dataset would have left us with too little data to train the models. Random Oversampler technique that generates samples by duplicating the data from the minority class until the dataset becomes balanced is used. By doing so, we not only reduce the bias towards the majority class but also increase the chances of getting better precision and recall scores. The effectiveness of this approach is evaluated in the validation phase.

3.5 Feature Selection

The presence of irrelevant features, especially in the case of big datasets, delays the model training and predicting process because models process all the features fed to them for better output. In our feature selection phase, we used the feature importance score generated by random forest as an index to identify the less important features based on the research conducted by Hasan et al (2016) that explored the enhancement of IDS through feature selection using random forest. They highlighted the challenges faced by having redundant features that are irrelevant to dependent variables and proposed a two-step feature selection process in which firstly features are ranked by permutation importance and then the most relevant subset is selected using random forest. Their results concluded that the reduced feature set showed a clear improvement in accuracy along with decreased processing time, highlighting the effectiveness of random forest in feature selection. Similarly, we also visualised the importance of features using the random forest and manually removed non-contributing features to increase model efficiency with reduced processing time.

3.6 Model Implementation

After data preprocessing and transformation steps, datasets were split into train and test sets with a split value of 0.3, meaning that our training set would have 70% of the data to train the models and 30% would be used to check the predictive capability of models.

After splitting the data, training data will be given to the machine learning classification models pipeline, including logistic regression, naïve bayes, random forest, XGBoost, KNN, SVM, and ANN models. Once the models are trained, test sets will be used to check the predictive capability of models.

The reason for using so many models is that they work on different architectures and train on data based on their predefined frameworks which can help to perform an in-depth analysis of the suitability of models with intrusion detection problems.

3.7 Hyperparameter Tuning

Hyperparameter tuning is an important step for optimising the performance of models by tuning them at their best set of parameters. The process involves searching and evaluating different parameters until the best hyperparameter that gives the highest accuracy on the validation set is not found. It is done to get improved accuracy while reducing the model overfitting to develop more robust models. For our research, we applied Grid Search to the same model pipeline used during model implementation where we defined the parameters for each model that needed to be tuned. Grid search iterates and evaluates each model on the datasets until the best parameters for that model are not found.

An important consideration while performing hyperparameter tuning is that you cannot always expect an increased accuracy as sometimes models overfit and tuning them removes the risk of overfitting, resulting in reduced accuracy. However, these models are predictively more reliable than the ones giving high accuracy while overfitting.

3.8 Evaluation

After training the models, their performance is checked on the test set. The predictions made by the model are interpreted using relevant evaluation metrics and their performance in solving the relevant problem is analysed. For this purpose, evaluation metrics common for all models (Model-Agnostic) are used to better compare the predictions made by models using the same scales. In our case, for visual analysis of results confusion matrix is made that shows the spread of True positives (TP), True negatives (TN), False positives (FP) and False negatives (FN) in a heatmap. In addition to that, we have used model accuracy, precision, recall and F1-score as the evaluation metrics.

- **Accuracy** is the ratio of true positive and true negative prediction to the total number of instances. It is used to calculate the overall correctness of the model

$$\textbf{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

- **Precision** is the ratio of true positive predictions to all the positive predictions made by the model. It calculates the accuracy of all the positive predictions.

$$\textbf{Precision} = TP / (TP + FP)$$

- **Recall** also known as sensitivity is the ratio of correctly predicted positives to all the instances that are actually positive.

Recall = TP / (TP + FN)

- **F1 score** is the harmonic mean of precision and recall that provides the balance between both.

F1 Score = 2 × (Precision × Recall) / (Precision + Recall)

4 Design and Implementation

We have followed the model-agnostic approach to ensure that all the models are given similar scenarios so that no specific model type is favoured. It's done to avoid the bias of research towards one model. As explained by Mishra, the concept of model agnosticism refers to model independence and the ability to explain machine learning models without assuming the structure or type of the model (Mishra, 2021). This approach is important for evaluating different models as it allows us to evaluate and compare different models based on their performance rather than their specific nature or working mechanism details. As seen in methodology, data preprocessing and transformation techniques common to all models are applied whose benefits are not limited to one model type.

4.1 Model Selection

Following this approach, we developed a machine learning models pipeline including logistic regression, naive bayes, random forest, XGBoost, KNN, SVM, and ANN integrated with the standard scaler for feature scaling. This pipeline trains the models on the training dataset and then evaluates them on both training and test sets. After that, it displays the results for each model using evaluation metrics such as training accuracy, test accuracy, precision, recall, F1 score and confusion matrices.

The reason for using models that follow different working mechanisms and architectures is to carry out a broad-spectrum analysis for our specific research aimed at enhancing network security. The evaluations made by these models will be compared to identify our inference model.

4.2 Implementation Structure

A specific model implementation structure is followed as shown in Figure 1. to analyse the generalizability of the proposed approach across diverse datasets. After initial data preprocessing, the model pipeline is first applied and evaluated on the imbalanced dataset. After performing oversampling of minority classes to make the datasets balanced and feature selection to make the datasets less redundant, the same model pipeline is applied again to the balanced and less redundant dataset. In the end, hyperparameter tuning of all models is performed and applied again on the balanced datasets. The results obtained from all three model implementations are stored and compared to analyse the overall improvement after each step. These steps aim to check the effect of class imbalances, oversampling, feature selection and model tuning on intrusion datasets aiming to fill the gap of data imbalance issues highlighted in the literature review. The results will help us to finalise our inference model to be suggested for real-life implementation based on performance.

5 Evaluation

After the implementation of models on the training and testing sets at different phases starting from the initial imbalanced dataset, moving on to balanced data after feature selection with both hyperparameter-tuned models and the models with their default parameters, the results from each phase are recorded and will be evaluated based on the evaluation metrics used. This evaluation will help us to better understand the impact of data preprocessing, transformation and hyperparameter tuning on the model performance for network traffic classification and intrusion detection. The same evaluation will be carried out for all three datasets to make a more generalised conclusion.

5.1 Case Study 1: Internet Firewall Dataset

In our first case study, we will deal with the results obtained from the IFD dataset. It was a relatively smaller dataset having an Action column to deal with network traffic and classify the connection as allow, deny, drop or reset. The results obtained from different models when applied using model agnostic pipeline on the imbalanced and balanced dataset with and without hyperparameter tuning are given in Table 1. of the report.

Table 1: Evaluation of ML Models for IFD data

Model	Dataset	Training Accuracy	Test Accuracy	Recall (Macro)	Precision (Macro)	F1 Score (Macro)
Logistic Regression	Imbalanced	98.75	98.49	73.22	73.12	73.13
	Balanced	81.50	81.43	81.44	81.25	81.32
	Balanced (Tuned)	81.73	81.67	81.68	81.46	81.54
Naive Bayes	Imbalanced	99.16	99.02	78.63	75.26	75.55
	Balanced	80.00	80.04	80.10	87.67	76.64
	Balanced (Tuned)	80.00	80.04	80.10	87.67	76.64
Random Forest	Imbalanced	99.99	99.76	80.31	94.63	83.39
	Balanced	99.99	99.97	99.97	99.97	99.97
	Balanced (Tuned)	99.99	99.97	99.97	99.97	99.97
XGBoost	Imbalanced	99.94	99.77	81.73	90.31	84.35
	Balanced	99.89	99.87	99.87	99.87	99.87
	Balanced (Tuned)	99.89	99.87	99.87	99.87	99.87
KNN	Imbalanced	99.71	99.56	74.57	74.39	74.48
	Balanced	99.91	99.82	99.82	99.82	99.82
	Balanced (Tuned)	99.95	99.90	99.90	99.90	99.90
SVM	Imbalanced	98.67	98.47	73.48	73.04	73.23
	Balanced	84.09	84.04	84.05	84.18	83.85
	Balanced (Tuned)	85.76	85.84	85.84	85.68	85.73
ANN	Imbalanced	85.11	84.90	49.70	39.30	43.06
	Balanced	77.55	77.48	77.54	85.82	72.98
	Balanced (Tuned)	91.61	91.60	91.60	91.52	91.55

5.1.1 Is Accuracy a valid evaluation metric for Imbalanced Data?

It may seem rare that all the models tend to give higher training accuracy when applied to the imbalanced dataset. To ensure the models are not overfitting, test accuracies can also be seen to be giving the same scores as training. Especially linear models like logistic regression and simple models like naïve Bayes giving more than 95% accuracy on both training and test sets made us question the authenticity of using only accuracy as an evaluation metric since accuracy only measures correctly classified values, and since datasets are highly imbalanced i.e. one or two classes cover 90% records in dependent class, then accuracy gets influenced by the correct classification of those majority classes to give a higher score, doesn't matter how many minority classes are overlooked by the models. That's why for the imbalanced dataset, precision, recall and F1-score are prioritized as authentic evaluation metrics.

5.1.2 Performance on Imbalanced Data

From the F1 scores, we can see that none of the models other than ensemble models were able to achieve more than 80% accuracy. Even ANN seems to struggle with imbalanced data and performed worst among all models when applied to an imbalanced dataset. So, in case of imbalanced data, we can declare ensemble models to be better performing with XGBoost having an edge over a random forest.

5.1.3 Balanced data

In this step, models are applied to cleaned and balanced data achieved after feature selection and oversampling of minority classes. After class balancing, a significant improvement in the performance of models can be seen. Even the ANN model that performed poorly on an imbalanced dataset showed an increase of almost 30% in its F1 score.

Another thing worth noting is that the training and testing accuracy for logistic regression, naïve Bayes, SVM and ANN have decreased in the case of a balanced dataset proving our claim of majority classes influencing the Accuracy scores to give a biased evaluation. On the other hand, random forest, XGBoost and KNN have shown an unexpected performance with almost 100% efficiency for all the evaluation metrics including accuracies, precision, recall, and F1-score. However, the random forest has outperformed both KNN and XGBoost in the case of balanced data.

5.1.4 Model Tuning

Even though the ensemble and KNN had achieved maximum efficiency, still hyperparameter tuning is done using Grid Search, aiming to achieve an ideal performance from these models while also providing a chance of improvement to other models. Gaussian Naïve Bayes is inherently simple and does not have tunable hyperparameters. However, there was not much improvement in the case of random forest and XGboost. KNN and logistic regression had a slight improvement of less than 1%. SVM showed an improvement of almost 2% in its F1 score. While ANN, which had not performed well on imbalanced and balanced datasets and had less efficiency as compared to all models, unexpectedly showed an excellent improvement in its performance achieving almost a 30% increase in its F1-score efficiency outperforming logistic regression, naïve Bayes, SVM. However, in the overall result, random forest outperformed all the models even though the hyperparameter tuning didn't seem to benefit it much.

5.2 Case Study 2: UNSW-NB15 Dataset

In this case study, we analysed the second largest dataset of our research that had almost 45 features, out of which we dropped the most and only 28 were left after feature selection. It has about 10 classes in the dependent variable namely: Normal, Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode, Worms making it an extensive multiclass classification. Table 2 gives the results for the UNSW-NB15 Dataset.

Table 2: Evaluation of ML Models for UNSW-NB15 Dataset

Model	Dataset	Training Accuracy	Test Accuracy	Recall (Macro)	Precision (Macro)	F1 Score (Macro)
Logistic Regression	Imbalanced	82.51	82.59	44.63	66.73	44.71
	Balanced	58.29	58.57	58.40	58.44	57.10
	Balanced (Tuned)	58.4	59.03	58.86	58.96	57.64
Naive Bayes	Imbalanced	63.43	63.68	50.46	42.59	30.86
	Balanced	49.98	50.28	50.09	55.61	43.63
	Balanced (Tuned)	49.98	50.28	50.09	55.61	43.63
Random Forest	Imbalanced	91.26	86.67	58.90	75.08	62.14
	Balanced	80.53	79.52	79.47	83.62	80.47
	Balanced (Tuned)	80.53	79.52	79.47	83.62	80.47
XGBoost	Imbalanced	89.48	87.61	62.14	78.40	64.76
	Balanced	78.71	78.15	78.10	82.58	78.94
	Balanced (Tuned)	79.42	78.78	78.73	83.14	79.58
KNN	Imbalanced	85.16	82.38	49.28	59.35	51.34
	Balanced	77.95	76.84	76.75	78.14	77.16
	Balanced (Tuned)	78.59	77.66	77.57	79.51	78.10
SVM	Imbalanced	83.69	83.71	44.72	63.67	44.41
	Balanced	56.94	57.09	56.90	55.53	53.91
	Balanced (Tuned)	57.06	57.21	57.02	55.86	54.20
ANN	Imbalanced	87.38	86.80	55.68	70.65	57.80
	Balanced	77.12	76.72	76.68	82.32	77.49
	Balanced (Tuned)	72.42	72.33	72.29	76.35	72.80

5.2.1 Performance on Imbalanced Data

It was the most complex dataset of our research because most of the models could not reach satisfactory performance on the imbalanced distribution of data. The accuracy scores influenced by majority classes as explained in **Subsection 5.1.1**, seemed higher but precision, recall and F1-scores were not satisfactory, especially for logistic regression, naïve bayes and SVM, since they had less than 0.5 F1-score. KNN and ANN also had a low F1-score and could not even reach 60% efficiency. However, Random Forest and XGBoost seemed to be performing alright having more than 0.60 F1-score, with XGBoost showing its dominance in handling the imbalance data having 2% higher accuracy than Random Forest.

5.2.2 Balanced data

After data balancing, accuracy scores for all the models dropped proving the impact of the majority classes on accuracy for imbalanced data. Overall precision, recall and f1-score increased but the performance-wise sequence of models remained the same as logistic regression, naïve Bayes and SVM showed less performance than other models while KNN and ANN still performed the same. Random forest showed an accuracy of 80% surpassing XGBoost by 2%.

5.2.3 Model Tuning

Considering the complexity of the dataset, hyperparameter tuning showed minimum to no improvement for most models. Only SVM and KNN showed an increase of 1% accuracy while ANN showed a drop of 5% when tuned. Random forest remained the best-performing model on balanced data before and after hyperparameter tuning.

5.3 Case Study 3: NF-UQ-NIDS-v2 Dataset

This is the largest dataset of our research, having a dependent variable named 'Attack' that has 20 types of network attack classes in it such as DoS, Benign, scanning, DDoS, XSS, Bot, Reconnaissance, password, Fuzzers, injection, Theft, Brute Force, Infiltration, Exploits, Generic, Analysis, Backdoor, mitm, Shellcode, 'ransomware, Worms. The model agnostic pipeline is applied to this dataset at different phases and the results shown in Table 3 are recorded at each phase.

Table 3: Evaluation of ML Models for NF-UQ-NIDS-v2 Dataset

Model	Dataset	Training Accuracy	Test Accuracy	Recall (Macro)	Precision (Macro)	F1 Score (Macro)
Logistic Regression	Imbalanced	95.90	95.84	56.14	61.84	57.19
	Balanced	89.85	89.86	89.85	90.61	89.88
	Balanced (Tuned)	90.43	90.48	90.47	91.18	90.52
Naive Bayes	Imbalanced	59.21	59.12	60.83	50.60	45.29
	Balanced	76.96	76.82	76.85	80.21	75.11
	Balanced (Tuned)	76.96	76.82	76.85	80.21	75.11
Random Forest	Imbalanced	99.98	98.99	74.05	80.88	76.57
	Balanced	100.00	99.93	99.93	99.93	99.93
	Balanced (Tuned)	100.00	99.93	99.93	99.93	99.93
XGBoost	Imbalanced	99.32	98.85	78.46	85.29	79.73
	Balanced	99.86	99.78	99.78	99.78	99.78
	Balanced (Tuned)	99.93	99.86	99.86	99.86	99.86
KNN	Imbalanced	98.51	98.05	65.52	72.77	67.71
	Balanced	99.79	99.66	99.66	99.66	99.65
	Balanced (Tuned)	99.88	99.74	99.74	99.74	99.74
SVM	Imbalanced	96.57	96.55	58.89	68.57	60.38
	Balanced	93.64	93.60	93.60	93.79	93.56
	Balanced (Tuned)	95.13	95.17	95.16	95.29	95.15
ANN	Imbalanced	98.54	98.31	66.00	70.60	67.31
	Balanced	98.24	98.51	98.51	98.52	98.51
	Balanced (Tuned)	95.22	95.22	95.21	95.27	95.19

5.3.1 Performance on Imbalanced Data

The results of this case study show similar patterns as from Case Study 1 as there is a clear difference between accuracy and precision, recall, F1-scores for each model depicting the influence of majority classes on train and test accuracy as discussed in **Subsection 5.1.1.** So, we will be evaluating imbalanced dataset results based on the F1-Score rather than the accuracy except for naïve bayes, whose accuracy score is less than 60% and whose F1 score is also lowest than all the models proving that it is not suitable for this imbalanced dataset. Similarly, logistic regression and SVM couldn't perform well with the imbalanced dataset and failed to achieve satisfactory precision, recall and f1-scores. ANN and KNN models performed almost the same while could not achieve more than 70% performance efficiency, which was only achieved by ensemble models. From ensemble models, XGBoost outperformed random forest having 3% more efficiency for precision, recall and F1-score.

5.3.2 Balanced data

After data balance, a noticeable increase is seen in the performance of all the models showing more than 90% results for train and test accuracy, precision, recall and F1-score, except for naïve Bayes and logistic regression. Random forest, XGboost and KNN almost achieved an ideal predictive capability of a 100% efficient system with random forest being the best of all.

5.3.3 Model Tuning

With hyperparameter tuning almost all the models showed an improvement of less than 1% in their performance, except for SVM which showed a bit higher improvement while the performance of ANN reduced by 3% when set to specific parameters. Overall, the performance-wise sequence of models remained the same as it was without model tuning.

5.4 Discussion

From the case studies using different datasets and machine learning models, we have analysed that different sizes or numbers of features have no significant effect on the predictive capability of models except for the need for more computational power and time. It is about the quality of the dataset and how its features contribute to the predictive process of models. Internet Firewall Dataset and NF-UQ-NIDS-v2 were two completely different data but the machine learning models performed well on them. Similarly, the UNSW-NB15 dataset was a medium size data but was a bit complex for models to process. Still, significant improvements were analysed after addressing the class imbalance, feature selection and hyperparameter tuning of models.

Another important finding that we analysed in this research is that while dealing with imbalanced datasets, evaluation metrics such as train and test accuracies cannot be trusted alone, hence precision, recall and f1-score are important for the proper evaluation of model performance. Similarly, while dealing with intrusion datasets, for better training of models it's better not to remove outliers unless they are too big or infinite to affect model performance because most of the time intrusions are the outliers, that are important for better predictive training of model. Hyperparameter tuning is an important step, however, it cannot always make the models perform better, so there is a need to focus more on better preprocessing, feature scaling, encoding and handling class imbalance issues. Class imbalance is an important issue that was overlooked by most studies in the literature.

Ensemble learning Random Forest and XG Boost models continuously performed well across imbalanced and balanced datasets outperforming all other models. Balanced datasets showed a significant increase in the performance of KNN and ANN. Hyperparameter tuning showed a little improvement for all models, however, ANN and SVM had significant changes after hyperparameter tuning, at some phases, they showed improved performance after tuning and at some phases showed a decrease in their predictive capability. SVM also requires a lot of computing power and time and did not perform well with one of the datasets, making it less preferable for intrusion detection problems. Logistic regression and Naïve Bayes being suitable for simplistic problems cannot be used for intrusion detection problems. Overall, this model-agnostic machine-learning approach has been useful in finding our inference models which are XGBoost and random forest. If dealing with an imbalanced dataset without oversampling, then XGBoost performs better and on balanced data Random Forest outperforms all models. This study has helped us narrow down our range of model selection, now instead of randomly picking any model we know that models like logistic regression, naïve bayes, and SVM should be avoided. Instead, focus should be put on KNN, ANN, XGBoost and most importantly Random Forest, when planning to integrate machine learning along with Traditional Firewalls for real-life network security purposes.

The inference models, Random Forest and XGBoost, can handle hundreds of thousands of instances efficiently however, for real-time network security, we propose using under-sampling and oversampling techniques to balance class instances to an average threshold. Under-sampling will manage majority classes while oversampling will address minority classes, preventing data abundance issues while enhancing model efficiency and computational speed with minimal resources. To make the model agnostic approach capable of handling newer and more complex attacks, regular updates will be performed to the training dataset by including new types of attacks in it. This will ensure the models remain relevant and capable of detecting new attacks.

6 Conclusion and Future Work

To fulfil the gap of implementing machine learning along with current intrusion detection systems and firewalls, our research aims at enhancing network security and intrusion detection by analysing and evaluating different machine-learning algorithms namely Logistic Regression, Naïve Bayes, Random Forest, XGBoost, KNN, SVM and ANN tested on three different network traffic datasets. The goal is to propose such a strong predictive model that can defend against updated threats and cyber-attacks to create a safer environment. The biggest problem with intrusion datasets is the class imbalance between the normal and attack traffic which makes it difficult to train models on enough attack data. We proposed a random oversampling and feature selection technique to handle class imbalance. To further enhance the predictive capability of models, we proposed Grid-Search hyperparameter tuning that showed significant improvement in the performance of some models. Random Forest and XGBoost are the two models that not only performed well on the imbalanced datasets but also showed an improved accuracy of 99% on two and 80% accuracy on the third balanced datasets while outperforming all other models.

Despite achieving these effective results, we acknowledge shortcomings like the need for more computational cost due to oversampling techniques and for future research we suggest implementing both under-sampling and over-sampling techniques aiming to lower the amount of majority class variables and increasing minority class variables, hence creating an average balance.

Our research narrowed down only to the use of ensemble models, XGBoost and Random Forest as they require less computational power and execution time while giving way better results and we propose them to be implemented and tested in real-world scenarios. To handle the new attacks in the coming times that may not have been labelled yet, we suggest anomaly or outlier detection methods like the Isolation Forest algorithm to be utilised along with the proposed approach to separate attack data from normal data behavior and label such data as anomalies to identify it easily. These anomalies can also be oversampled to retrain the models for better efficiency. Similarly, for future work, we propose the use of more complex and updated datasets using stronger neural network models like CNNs and RNNs along with ensemble models aiming to develop a 100% efficient intrusion detection system.

Acknowledgement

I am extremely thankful to my supervisor, Mr. Arjun Vinayak Chikkankod for providing his invaluable support and guidance throughout this research project. Their encouragement, expertise and feedback played an important role in the development of this work. I'm grateful for his patience and dedication towards the successful completion of my research.

References

Byri Manoj S, Rajeshkumar M, Santhakumar R and Balaji S (2021). Privacy and Security: Internet of Things. *2021 Innovations in Power and Advanced Computing Technologies (i-PACT)*. doi:https://doi.org/10.1109/i-pact52855.2021.9696554.

Md Habibur Rahman, Islam, T., Md Masum Rana, Rehnuma Tasnim, Tanzina Rahman Mona, and Md Mamun Sakib (2023) 'Machine Learning Approach on Multiclass Classification of Internet Firewall Log Files', *Proceedings of the IEEE CISES 2023*, doi: 10.1109/CISES58720.2023.10183601.

Hsiao, S.-C. & Kao, D.-Y., (2018). The static analysis of WannaCry ransomware. *In: International Conference on Advanced Communications Technology (ICACT)*, 11-14 February 2018. IEEE. [online] Available at: https://doi.org/10.23919/ICACT.2018.8323680

Aswal, K., Rajmohan, A., Trc, A., Mukund, S., Panicker, V. and Dhivvya, J.P. (2021) 'Kavach: A Machine Learning based approach for enhancing the attack detection capability of firewalls', *Proceedings of the International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, doi: 10.1109/icccnt51525.2021.9579836.

Lin, Z. and Yao, Z. (2022) 'Firewall anomaly detection based on double decision tree', *Symmetry*, 14(12), p. 2668. doi:10.3390/sym14122668.

Neupane, K., Haddad, R. and Chen, L. (2018). Next Generation Firewall for Network Security: A Survey. *SoutheastCon 2018*. https://doi.org/10.1109/secon.2018.8478973.

Samrin, R. and Vasumathi, D., (2017). Review on anomaly based network intrusion detection system. *IEEE Xplore*. Available at: https://doi.org/10.1109/ICEECCOT.2017.8284655.

Applebaum, S., Gaber, T. and Ahmed, A. (2021). Signature-based and machine-learning-based web application firewalls: A short survey. *Procedia Computer Science,* 189, pp. 359–367. https://doi.org/10.1016/j.procs.2021.05.105.

Appelt, D., Nguyen, C.D., Panichella, A. and Briand, L.C., (2018). A machine-learning-driven evolutionary approach for testing web application firewalls. *IEEE Transactions on Reliability*, 67(3), pp.733–757. doi: https://doi.org/10.1109/tr.2018.2805763.

Haripriya, L. and Jabbar, M.A., (2018). Role of machine learning in intrusion detection system: Review. *IEEE Xplore*. Available at: https://doi.org/10.1109/ICECA.2018.8474576.

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y. and Han, H. (2022) 'A systematic literature review of methods and datasets for anomaly-based network intrusion detection', *Computers & Security*, 116, p. 102675. doi: https://doi.org/10.1016/j.cose.2022.102675.

Subba, B., Biswas, S. & Karmakar, S., (2015). Intrusion detection systems using linear discriminant analysis and logistic regression. In: *2015 Annual IEEE India Conference (INDICON)*. IEEE, pp. 1-6. doi:10.1109/INDICON.2015.7443533.

Li, W., Yi, P., Wu, Y., Pan, L. and Li, J. (2014). A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network. *Journal of Electrical and Computer Engineering*, [online] 2014, pp.1–8. doi: https://doi.org/10.1155/2014/240217.

Ding, H., Chen, L., Dong, L., Fu, Z. and Cui, X. (2022). Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Generation Computer Systems*, 131, pp.240–254. doi:https://doi.org/10.1016/j.future.2022.01.026.

Panda, M., Abraham, A. and Patra, M.R. (2010) 'Discriminative multinomial Naïve Bayes for network intrusion detection', *Proceedings of the 2010 International Conference on Systems, Man, and Cybernetics* (ISIAS), pp. 3739-3744. doi: 10.1109/ISIAS.2010.5604193.

Koc, L., Mazzuchi, T.A. and Sarkani, S. (2012). A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. *Expert Systems with Applications*, 39(18), pp.13492–13500. doi:https://doi.org/10.1016/j.eswa.2012.07.009.

Jing, D. and Chen, H.-B. (2019). SVM Based Network Intrusion Detection for the UNSW-NB15 Dataset. *IEEE Xplore*. doi:https://doi.org/10.1109/ASICON47005.2019.8983598.

Sahu, S. and Mehtre, B.M. (2015). Network intrusion detection system using J48 Decision Tree. [online] *IEEE Xplore*. doi:https://doi.org/10.1109/ICACCI.2015.7275914.

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J. and Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, [online] 32(1). doi:https://doi.org/10.1002/ett.4150.

Bagui, S. and Li, K. (2021). Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data*, 8(1). doi:https://doi.org/10.1186/s40537-020-00390-x.

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.

Fu, Y., Du, Y., Cao, Z., Li, Q. and Xiang, W. (2022). A Deep Learning Model for Network Intrusion Detection with Imbalanced Data. *Electronics*, 11(6), p.898. doi:https://doi.org/10.3390/electronics11060898.

Shone, N., Ngoc, T.N., Phai, V.D. and Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1), pp.41–50. doi:https://doi.org/10.1109/tetci.2017.2772792.

Farnaaz, N. and Jabbar, M.A. (2016). Random Forest Modeling for Network Intrusion Detection System. *Procedia Computer Science*, [online] 89, pp.213–217. doi:https://doi.org/10.1016/j.procs.2016.06.047

Sharma, D., Wason, V. and Prashant Johri (2021). Optimized Classification of Firewall Log Data using Heterogeneous Ensemble Techniques. *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. doi:https://doi.org/10.1109/icacite51222.2021.9404732.

Gao, X., Shan, C., Hu, C., Niu, Z. and Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, 7, pp.82512–82521. doi:https://doi.org/10.1109/access.2019.2923640.

Das, S., Saha, S., Priyoti, A.T., Roy, E.K., Sheldon, F.T., Haque, A. and Shiva, S. (2021). Network Intrusion Detection and Comparative Analysis using Ensemble Machine Learning and Feature Selection. *IEEE Transactions on Network and Service Management*, pp.1–1. doi:https://doi.org/10.1109/tnsm.2021.3138457.

Jiang, H., He, Z., Ye, G. and Zhang, H. (2020). Network Intrusion Detection Based on PSO-Xgboost Model. *IEEE Access*, [online] 8, pp.58392–58401. doi:https://doi.org/10.1109/ACCESS.2020.2982418.

Moustafa, N. and Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems *(UNSW-NB15 network data set)*. [online] IEEE Xplore. doi:https://doi.org/10.1109/MilCIS.2015.7348942.

Lundevall-Unger, P. and Tranvik, T. (2010). IP Addresses - Just a Number? *International Journal of Law and Information Technology*, 19(1), pp.53–73. doi:https://doi.org/10.1093/ijlit/eaq013.

Jabez, J. and Muthukumar, B. (2015). Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach. *Procedia Computer Science*, 48, pp.338–346. doi:https://doi.org/10.1016/j.procs.2015.04.191.

Mohammed, R., Rawashdeh, J. and Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. [online] *IEEE Xplore*. doi:https://doi.org/10.1109/ICICS49469.2020.239556.

Hasan, Md.A.M., Nasser, M., Ahmad, S. and Molla, K.I. (2016). Feature Selection for Intrusion Detection Using Random Forest. *Journal of Information Security*, 07(03), pp.129–140. doi:https://doi.org/10.4236/jis.2016.73009.

Mishra, P. (2021). Model-Agnostic Explanations by Identifying Prediction Invariance. *Apress eBooks*, pp.299–314. doi:https://doi.org/10.1007/978-1-4842-7158-2_12.