National College of Ireland

Advanced ML Approaches for Intrusion Detection: A Comprehensive Analysis Using UNSW-NB15 and NSL-KDD Datasets

MSc Research Project
Cyber security

Lourdu Mary Gade
Student ID: 23188189

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Lourdu mary gade
Student ID:	23188189
Programme:	Cyber security **Year:** 2024-2025
Module:	Final thesis
Supervisor:	Khadija Hafeez
Submission Due Date:	29 January
Project Title:	Advanced ML Approaches for Intrusion Detection A comprehensive Analysis using UNSW-NB15 and NSL-KDD Datasets
Word Count:	12467 **Page Count** 22

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

Signature:	G lourdu mary
Date:	29-01-2024

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):	

Advanced ML Approaches for Intrusion Detection: A Comprehensive Analysis Using UNSW-NB15 and NSL-KDD Datasets

LOURDU MARY GADE

23188189

Abstract

The exponentially of complex connecting systems in the information age has brought new and more difficult challenges in the cyber defence, as current Intrusion Detection Systems that rely solely on static attack signatures fail to protect against zero day attacks or advanced persistent threats (APTs). This thesis proposes a method of anomaly detection in networks that are part of large Systems of Systems SoS without prior definition of specific signatures of attacks, using the ML ML techniques. This study aims at improving the performance and precision of intrusion detection using benchmark classifiers including Random Forest, XGBoost, and Support Vector Machines (SVM), as well as benchmark datasets including NSL-KDD and UNSW-NB15. Pearson correlation factor is used in feature selection together with methods such as recursive feature elimination in order to refine inputs for enhancing the general model. The models are testing thoroughly for the accuracy, precision, recall, and F1-score that gives helpful information about discovering both known and new threats in cyber. Regarding critical issues, for example, high false-positive rates, or the need for further development of non-specific IDS models that would be able to address new threats so prevalent every time more sophisticated network structures are used, this research proposes the solutions for further development of highly effective, virtually non-resource-consuming IDS systems. The results have demonstrated that ML can be implemented as a strategic innovation in cybersecurity research and highlights a model for developing intelligent systems to counter contemporary threats.

Keywords: Network Security, Intrusion Detection, NSL-KDD, NSL-KDD, UNSW-NB15, ML, SVM, Random Forest, XGBoost.

1 Introduction

1.1 Background

Under the conditions of digital changes in society, connections of devices, systems, and networks form the basis of innovative development. This accelerated growth of network and interconnected systems referred as Internet of Things or IoT is expected to incorporate N number of connected devices and interface with mission-critical applications including health care, energy, finance, and government sectors by mid-2025. At the same time, such interconnection has opened opportunities for unprecedented development of innovations and productivity; it has also given rise to enhanced security risks. More traffic, in greater velocity and from a variety of sources, has made these systems a more open and exposed target for cyberattackers. Intrusion Detection Systems (IDSs) are critical tools in protecting any network based structures. They conduct surveillance on activities within a network and on the system as a whole with a view of identifying intruders or misuse. There are two main types of IDSs: the first generation which is signature-based and the second generation which is behaviour-based. Although these systems provide protection against such threat, they are not efficient in combating the uncertainty and variety of today's cyber threats. Advanced threats like Zero-day attack and Advanced persistent threat attack the unexplored weak points of the systems and do not raise alarms in the signature based IDS. There is an exception here due to

the deficiencies of existing system based approach which requires building new, more smart systems capable of detecting known and other, emerging, threats.
Network anomaly detection has thus been presented as a more viable solution than conventional approaches. Anomalous detection systems differ from the signature-based systems, since analyse deviations from normal network activity which helps to identify the activities that do not resemble that of normal attacker. Zero and low volume threats such as zero-day attacks are easily masked by normal traffic and this capability is very useful in identifying such threats. But, building efficient and necessary anomaly detection is not without its challenges. The measure of ordinary network traffic, meanwhile, is considerably erratic and differentiating between what is commonly a healthy portion of variance and an indication of danger often demands sophisticated analysis. Impossibly high false-positive ratios and computational costs are still unresolved issues that hamper the vast adoption of the anomaly detection systems.

1.2 Significance of Intrusion Detection

Consequently, there is a need to show a new approach to cybersecurity due to complexity and importance of cyber targets and networked systems. To address these challenges IDSs needs to integrate adaptive forms of technologies that can reason about the threats and act cinematographically IDS Superstar IDS Superstar now come of age as ML (ML) has addressed the frontier of anomaly detection as ability to modeled complex pattern of network traffic that suggests anomaly resemble a malicious activity. Offering ability to model complex patterns in network traffic and identify deviations indicative of malicious activity. System and network anomalies can be determined or flagged through ML algorithms that are, for example, fed terabytes of actual network traffic and then taught to respond to these input patterns in the absence of preprogramed signature- and rule-based definitions of "normal." This make ML-based IDSs especially useful in handling the form of intrusion because the threat evolves in due time. Furthermore, they are extensible to support the current demanding networks where huge volumes of data must be processed with great precision and speed. Such a technique could prove especially powerful in ML-based anomaly detection systems no longer limited to reactive and immediate defense, but rather proactive identification of threats.

1.3 Challenges in Network Anomaly Detection

Despite its promise, the development of ML-based anomaly detection systems faces several challenges:

- **High Variability in Network Traffic:** Normal network traffic behaviour is much more volatile and may depend heavily on the factors like user activity, network architecture, as well as the nature of applied applications. This variability has the added effect of making it difficult to properly define what different levels of normal behaviour are.
- **Evolving Threat Landscape:** There is a progressive increase in the level of evolving threats whereby malware can change forms for example through polymorphism, carriers of malware often using encryption on their payload and disguising their traffic. As seen today some attackers use different tactics and an effective IDS must cater for them.
- **Balancing Accuracy and False Positives:** The high detection accuracy is mandatory to detect threats, while false positives or normal activities being marked as an infection can flood the security team and harm productivity.

- **Scalability and Real-Time Processing:** Current and evolving network traffic is massive, demanding platforms that can make use of, analyze data in real time and without significantly impacting either the speed or the efficiency of the system.
- **Feature Selection:** Filtering out the most important features out of network traffic data is a vital stage as it determines either increased performance of the model, and either computational load.

1.3.1 Role of Benchmark Datasets

The IDS research, development, as well as the evaluation processes need quality benchmarks that reflect real traffic models. This study employs two widely recognized datasets: NSL-KDD and UNSW-NB15. These datasets have normal as well as all classes of attack traffic and are fairly representative of a typical network environment.
NSL-KDD: NSL-KDD is one of the oldest and the most frequently used datasets for intrusion detection research, which provides normal connections and four types of attacks: DoS, Probe, R2L, and U2R. A disadvantage of this dataset is that it has a lot of duplicated attacks, and some of the attacks are outdated.
UNSW-NB15: To overcome the shortcomings of previous datasets, UNSW-NB15 offer a better understanding of current distributed network traffic patterns. It provides realistic simulations of both normal and adversarial activities; the different types of attacks it contains are Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms.

1.4 Objectives

The main research questions of this thesis are as follows: 1) How to design and construct feature vector for network traffic data? 2) How to train and select the best ML algorithm for detecting network anomaly using NSL-KDD and UNSW-NB15 datasets? These datasets are generally used to establish the basis for the construction of IDSs for the evaluation of the normal and abnormal network traffic. The research also seeks to enhance the feature selection process through introduction of efficient methods like the Pearson correlation and the recursive feature elimination techniques. Thereby, these methods assist in filtering out attributes of lesser and no relevance originating from network traffic data to enhance model efficiency as well as to minimize computational intensity. To attain these objectives, this study seeks to observeand implement the performance of various ML classifiers; Random Forest, XGBoost, SVM. These are selected algorithms to work with high-dimensional data and relationship modelling allowing identification of the known threats as well as the previously unseen ones. The study also covers significant questions, including how to reduce high false positives rates that drown security teams, and how IDS systems are to be further developed and adapted to the often vast scale of learning environments.

1.5 Research Questions

That being the case, the study focus on answering 3 most important researched Questions, which define and underpin most issues of network security. The first question poses whether the applications for that work enhance zero-day and novel cyberattacks identifiers in network traffic for the respective ML models. Due to the nature of the attack being susceptibility that is not recognized before a zero-day attack, it is out of the question to employ signature based detection. By doing so this study uses ML as the method of determining patterns and detecting outliers. The second question relates to capturing of feature selection approaches used in the improvement of the proposed ML-based anomaly detection systems. But a quantum leap in cutting the operational costs arises when one minimizes the number of

features used in the models to increase the speed of detection than the previous models. Last, the study advances towards addressing the problem of designing medium-high scale networks with an acceptable level of detection accuracy, and with low computational complexity. Real threat identification in realistic scenarios coupled with concerns of scalability and efficiency is a challenging and fundamental issue that needs to be uniquely solved in the current generation of IDSs that function in today's diverse and complex network environments handling huge volumes of information.

1.6 Methodology Overview

The approach used in this research is designed in such a way that will cover the research questions and also draw out the best models for an accurate ML-based anomaly detection system. The first step that are involved include data pre-processing where all the data sets are prepared for analysis. This is all encompassing from imputing missing values, converting categorical data into numerical and scaling features to a unified form. After that, only necessary fields for the anomaly detection are chosen using the Pearson correlation coefficient and a recursive feature elimination algorithm for better performance and aspect ratio of the model. The selected features are then used to train the Random Forest, XGBoost, and Support Vector Machines (SVM) ML models. It is important usually assess all the models for performance using metrics like; accuracy, precision, recall and F1-score. Finally, the performance of all the different models is compared to ascertain which approach is most effective in identifying s network anomaly. It also guarantees a proper and strong preliminary and a consistency in dealing with the research questions.

1.7 Significance of the Study

This research is relevant to the current literature on cybersecurity because it builds an anomaly detection system for both known threats and new threats with the help of ML approaches. This is because old attack techniques cannot hold out today complex threat such as zero day, APTs amongst others and it is adaptive and automatically respond. The application of complicated feature selection algorithms and state-of-art pattern recognition classification engines employed in this paper should lead to enhanced dependability, efficiency, and scalability of the Intrusion Detection Systems. In addition to that, the findings contribute to the literature because they provide the guidelines, which should be followed when implementing ML within the framework of cybersecurity and specifically with reference to the development of the new IDS approaches. This work provides a solution for increasing detection capability while at the same time outlining a plan to modify this model without compromising effectiveness even under authentic large scale computer network environments.

2 Related Work

Malathi and Revathi (2013) employ and explain different classification models to the NSL-KDD dataset, such as decision tree, K-nearest neighbor, support vector machine, and Naive Bayes. This paper assesses the capability of these models in identifying network attacks like DoS, Probe, R2L, and U2R. Nevertheless, the authors observed that whilst both Decision Trees and SVM was effective for some of the attacks, they were less effective for detecting complex or less frequently occurring attacks such as U2R. The study also played much attention to feature selection whereby the model performance would be improved. Yet, critical issues of high FP rates were left unresolved for Decision Trees, and feature selection procedure was not fine-tunedc to achieve improved model accuracy. This thesis expands on their work by applying more sophisticated classifiers including Random Forest and XGBoost, as well as employing the latest feature selection methodologies, for example Pearson

correlation and recursive feature elimination, in an endeavor to reduce the number of false positive results and enhance detection efficacy. In this way, this research offers a more effective and efficient IDS solution on a large scale. **[1]**

In their study work, Dhanabal and Shantharajah (2015) identified classifiers – k-NN, SVM, and Random Forest – and centered their performance assessment on the NSL-KDD dataset. According to their study they found out that Random Forest was more accurate in its classification and had the lowest False Positive Value. Although, their work lacks the evaluation of feature selection methods and extensive study of the models' performance using a rich set of performance measures. This thesis builds on their work, where different methods for feature selection, such as recursive feature elimination, have been applied; as well, the performance evaluation of the models incorporate F1-score, precision, and recall. Additionally, the research seeks to use XGBoost which has been shown to outcompete Random Forest in some scenarios. **[2]**

The IDS performance enhancement study by Chae et al. (2013) concentrated on feature selection from a set of very diverse features extracted from the NSL-KDD data set using feature selection approaches including correlation analysis and Genetic Algorithms. They also determined that the success of an IDS could be improved with feature selection by lowering the dimensionality of the dataset and concentrating on meaningful features. However, they failed to investigate the performance of different classifiers in the improved attribute set and failed to address fundamental issues such as, high false positive rates and the ability of the models to scale up. The present thesis extends from their work by assessing the convergence of multiple classifiers using the chosen features while fine-tuning the assessment. It also tackles scalability since it features computational complexity which is very vital for large networks. **[3]**

Classification analysis in using ML techniques was studied by Masoodi (2021) using NSL-KDD dataset. The study also mentioned how classifiers such as Logistic Regression, SVM, and Random Forest performed in regard to detecting different forms of the attacks. In particular, Random Forest and SVM were most accurate in the identification of DoS and Probe attacks, while the Logistic Regression model was less efficient. Nevertheless, the methodology of the study did not allow using some more advanced feature selection strategies, nor it compared the classifiers' performance on less frequent types of attacks, for example, U2R. In extending the work of, the current thesis employs better feature selection techniques and benchmarks a wider array of classifiers including the complex XGBoost field. Furthermore, and more so, this study seeks to enhance the identification of less frequent attack types such as U2R while decreasing false positives. **[4]**

Xu et al. also looked at autoencoders for network anomaly detection and specifically dealt with making them work on NSL-KDD dataset. They outlined the use of autoencoders as a part of a more general model of ML methods that should be employed to enhance the anomaly detection process. Although, their proposed hybrid model revealed sign of performing well in identifying anomalous traffic, they did not study the effect of feature selection or discuss on how to handle scalability and computational cost in high dimension data sets. Building on from their work, this thesis further investigates how different classifiers, such as Random Forest, XGBoost, SVMs, and autoencoders compare. In addition, this research employs feature selection methods for enhancing model efficiency in large scale networking environments. **[5]**

In their work presented in Meftah et al. (2019), the authors employed the UNSW-NB15 dataset to test the detection capabilities of a number of ML algorithms on a network intrusion detection task. Their work and analysis revealed that Random Forest yields better performance than other classifiers especially in the identification of the more complicated attack types including Backdoors and DoS. Nevertheless, the work did not relate feature selection to the model performance and also lacked discussion on the use of sparse models for large networks. As a continuation of the work of Meftah et al., this thesis thus applies more sophisticated feature selection methods and also compares other classifiers such as XG Boost which has shown great potential in working with complicated data sets. Moreover, the research aims at achieving both improved detection rate and requires manageable resource utilization with an outlook towards scalability in large massive networks. **[6]**

Further, Choudhary and Kesswani (2020) used deep learning for analysing KDD-Cup'99, NSL-KDD and UNSW-NB15 for the IoT context. They acknowledged that deep learning models are more accurate than traditional ML classifiers especially in detecting complex types of attack in IoT. However, the research failed to explain issues like a high cost of computation and accommodating large networks. Their work is improved upon in this thesis since Pearson correlation and recursive feature elimination feature selection techniques are used to improve on the models and minimize false positive results. It also cares for scalability through classifiers such as XGBoost and Random Forest, more appropriate for big-scale surroundings. **[7]**

In a bid to establish the effectiveness of ID using ML classifiers, Al-Daweri et al. (2020) applied decision trees, Random Forest and SVM, and cross checked for their performance on both the KDD99 and UNSW-NB15 dataset. In this research Random forest and SVM models were used for detecting different attacks where Random forest gives better accuracy in compared to SVM. However, the work done here did not deal with feature selection or selection of other method and did not address scalability issues in real-time application. This thesis follows up on their work by including feature selection approaches and assessing models using additional assessment criteria, including F1 score, as well as model scalability and performance on big data. **[8]**

For this, Husain et al. (2019) have proposed an efficient network intrusion detection system based on XGBoost on the UNSW-NB15 dataset where concern has been given on detection accuracy rather than the number of false positives. XGBoost also seen outperformed other classifiers with high TPR and low FPR which indicate high accuracy and fewer false positive respectively. However, their study did not attempt feature selection, nor did it attempt to assess the model on instances of large networks. This research builds on their work by first employing feature selection techniques and second, assessing XGBoost in larger network environments, with additional benchmarks from Random Forest and SVM. **[9]**

Network intrusion detection Using different models random forest, SVM, KNN a comparative study was conducted through the UNSW-NB15 dataset Disha and Waheed, 2021. It is evident from their studies that the precision of Random Forest and SVM in terms of attack detection was the best. However, the study did not look at methods for selecting the features used in these models or look at the computational complexity of these models in large networks.st Neighbors, for network intrusion detection using the UNSW-NB15 dataset. They concluded that Random Forest and SVM provided best performance in detecting different attacks. However, study did not explore feature selection methods or focus on the computational efficiency of these models in large networks. This thesis extends their works

by using novel feature selection methods and assessing scalability and computational cost using classifiers such as XGBoost. **[10]**

More et al. (2024) extended and investigated UNSW-NB15 dataset by performing feature engineering and selection to increase IDS performance more study explored the effect of feature engineering on IDS models but it did not compare different type of classifiers and it lacked the consideration of the trade-off between the number of false positives and detection rates. This research builds upon their work by utilising more sophisticated classifiers such as XGBoost, and by evaluating scalability and false positive rate specifically in a real world deployment context. **[11]**

This allowed Ikram et al. (2021) to design a hybrid anomaly detection model incorporated with XGBoost ensemble with deep neural network models. They pointed out their study enhanced the detection accuracy and capability of detecting new threats, however they did not investigate the selection of features for hybrid model and the computation cost of the proposed hybrid model. That is, this thesis extends from their work by applying feature selection techniques with an aim to improving the efficiency of models and comparing the performance of XGBoost against other classifiers in an efficient and scalable anomaly detection framework. **[12]**

Mwanambekele et al. (2018) studied the performance of XGBoost to detect intrusion in the network and concluded that the classifiers outperformed others in detecting the intrusions while having fewer false alarms. However, they failed to compare the model's complexity and time taken to scale up or down. In this study, we build on Dhaliwal et al's work by including solutions to scalability, applying feature selection methods, and comparing XGBoost with other classifiers and models. **[13]**

Rana (2019) summarized that many studies involving anomaly detection within network traffic using ML and deep learning pointed out that deep learning models were preferable for detecting new kinds of attacks; however, the models were computationally intensive and challenging to implement on large datasets. This study does not look at optimization of the computation process or selecting certain features. This thesis fills these gaps by including feature selection methodologies and assessing XGBoost and Random Forest models for application in real-time and for big networks. **[14]**

Elmrabit (2020) compared classifiers for detecting anomaly, such as SVM, Random Forest and XGBoost algorithms. The study showed how XGBoost and Random Forest performed in identifying anomalous traffic though the authors did not use feature selection approaches or present scalability to real-world large scale networks. The present study expands on the work of Elmrabit in regard to feature selection and the computational complexity of different models while guaranteeing that the methods are implementable in large-scale networks. **[15]**

Feng, G. & (2016) considered resource allocation within the context of cloud computing scenarios; he stressed a maximum revenue paradigm. Its paper in the International Journal of Grid and Utility Computing focused on trying to get the best results for revenues from cloud services while not sacrificing computing fluidity. Nevertheless, this paper is mainly devoted to the analysis of cloud resource management and does not consider the applicability of the examined methods for network security or intrusion detection. Picking up from where Feng left off this thesis relates similar optimization techniques to IDS especially in NA to improve feature selection to improve the scalability of the models in real world networks. **[16]**

Another paper that gave great input on big data computing was The Anatomy of Big Data Computing in Software—Practice & Experience written by Kune, R.K. (2016). Here Kune went through several frameworks and technologies where he gave details of these technologies as applied in big data analysis especially on large data. But in general, this study did not explore the practical implementation of big data technologies in case of network intrusion detection. Therefore, this thesis extends Kune's research by employing big data methodologies, namely ML algorithm, for identification and suppression of network abnormalities on the basis of vast samples containing UNSW-NB15 and KDD-Cup'99 information. Besides, it resolves the problem of real-time intrusion detection in a large amount of big data processing. **[17]**

In his paper CNN and RNN Based Payload Classification Methods for Attack Detection published in Knowledge-Based Systems, Liu, H.L. (2018) examined CNN and RNN for payload classification to identify network attacks. Liu also showed that deep learning methods could accurately classify different categories of attacks based on the various payloads in a network. However, to a certain extent, the models' generalizability was not tested, and feature selection influence was not analyzed. This research builds upon the work of Liu, particularly in adopting more refined feature selection methods and in giving particular considerations to the computational complexity of CNNs and RNNs in large-scale network spaces; and in comparing the performance of the proposed models to more conventional ML models such as Random Forest and XGBoost. **[18]**

Within Çukurova University Journal of Natural & Applied Sciences, Mohammed & R. (2023 applauded the use of ML algorithms in anomaly detection of network traffic. Particularly, this study sought to explore the Probability of utilising multiple ML methods for network anomaly detection in order to identify new emerging threats within the network environment. The authors proved that ML algorithms could well detect inversions or other disturbances and, however it did not consider the factors of time and complexity. This thesis extends Mohammed's work by using more developed feature selection techniques to improve model accuracy and comparing the scalability and performance of new classifiers such as XGBoost and Random Forest classifiers in large network contexts. **[19]**

Rana (2019) summarized that many studies involving anomaly detection within network traffic using ML and deep learning pointed out that deep learning models were preferable for detecting new kinds of attacks; however, the models were computationally intensive and challenging to implement on large datasets. This study does not look at optimization of the computation process or selecting certain features. This thesis fills these gaps by including feature selection methodologies and assessing XGBoost and Random Forest models for application in real-time and for big networks. **[20]**

In 2019, Vinayakumar, R.A.-N., developed the intelligent intrusion detection system (IDS) based on deep learning theory in IEEE Access. It was also evident from the study that using deep learning models specifically neural networks, could achieve improved detection accuracy as compared to traditional classifiers in the ML models for different attack types most especially for the complex attack types. But it failed to discuss computational complexity, system extendibility, and the way with which various feature selection techniques can be incorporated. Following Vinayakumar, this thesis incorporates novel feature selection methods and investigates whether deep learning models can maintain real-time detection and scalability for large-scale networks besides comparatively assessing the performance of XGBoost and Random Forest classifiers. **[21]**

Rao et al. (2024) proposed an intelligent network intrusion detection system for SDN using XGBoost at the 15th International Conference on Computing Communication and Networking Technologies. In their paper, they were able to prove that XGBoost had the good ability to detect intrusion with high accuracy in SDN context where traffic flow. There are highly dynamism and the attack strategies. Nonetheless, the work did not address the use of sophisticated feature selection methods or the issues of computational complexity in practical bulky network configurations. The current thesis builds on the work done by Rao et al., wherein advanced feature selection techniques and scalability issues are addressed to determine if and how XGBoost along with other classifiers such as the Random Forest and SVM can be used in real-world applications where big data is present. Also, this research considers false positive reduction which is important for real-time-based IDS. **[22]**

In 2020, Wang & Lu designed a host-based anomaly detection framework based on XGBoost and LSTM for IoT devices. The above framework tried to identify network abnormality by incorporating XGBoost's classification capacity and LSTM capability to manage time-series that is so significant in the passage of time for identifying temporal anomalies with the IoT network traffic. This exact hybrid approach the study identified could effectively detect other kinds of attacks as well as those affecting IoT devices. Nonetheless, the issue of feature selection was not explored as to its effect on model performance, while the extension of the proposed model for large-scale IOT networks was not considered. Thus, this thesis extends Wang and Lu's work by employing more advanced feature selection methods to enhance the model's performance and by assessing the computational complexity of XGBoost combined with LSTM for practical large-scale IoT networks. **[23]**

Sabahi and Movaghar (2008) provided a comprehensive survey of intrusion detection systems (IDS) in their paper "Intrusion Detection: A Survey," which was delivered at the Third International Conference on Systems and Networks Communications. The paper also includes a presentation of the various IDS models such as the signature based, the anomaly based and it captures challenges faced by current IDS in implementing newly invented sophisticated attacks. The study also considered the development of IDS technologies and the value of advancing the technology for preventing newer forms of threats. Although the survey included numerous IDS techniques, it did not address the usage of ML algorithms either for IDS or for enhancing IDS scalability and flexibility. This thesis builds upon the work of Sabahi and Movaghar by implementing more complex ML algorithms such as XGBoost and Random Forest for the purposes of anomaly detection, using enhanced feature selection techniques, and considering the issue of scalability in networks in practice. **[24]**

In more recent work, Abuali, Nissirat, and Al-Samawi (2023) moved forward the causes of network security by using Deep Support Vector Machine (SVM) for intrusion detection. Their study therefore showed that the deep learning models developed using SVM had great efficacy in identifying different forms of network intrusions including DDoS and other advanced attacks. The authors learned that deep training tools, when applied with SVM, could immensely improve the detection performance and practical security. However, this study failed to explore the use of feature selection in enhancement of its outcome as well as did not address some of the issues that are associated with real-time learning such as scalability and computational complexities. Therefore, this thesis expands from their work by utilizing more sophisticated feature selection methods, like the Recursive Feature Elimination, and comparing the time efficiency of the SVM models alongside other classification algorithms including XGBoost to be relevant in large network settings. **[25]**

3 Research Methodology

This portion of report outlines the research methodology for developing an Intrusion Detection System (IDS) using ML (ML) models, which was implemented with two popular datasets: UNSW-NB15 and NSL-KDD. The largely foreland aim of this thesis is to employ the advanced ML algorithms such as the classifiers like XGBoost, the Random Forest classifiers, SVM and a decision tree classifier to undertake the detection of a plethora of cyberattacks that may be occurring on a network based on the network traffic analysis. The approach presented here includes data acquisition, data cleaning, feature extraction, model building, model assessment and benchmarking. The following sub-sections provide detailed descriptions of the tasks that are accomplished and the corresponding codes for each of the laid down methodology.

3.1 Data Collection

3.1.1 UNSW-NB15 Dataset

The UNSW-NB15 dataset is a relatively new and general dataset for network traffic that encompasses normal and anomalous behaviours. It was derived from ACCS, and the set data are real-world attack data, which make it a valuable tool for assessing IDS. The proposed dataset encompasses several features which captures traffic characteristics pertinent to the classification of attacks in an automated system. These traffic features measured at a given period, gives understanding on how a network behaves in normal and attack situations.
The UNSW-NB15 dataset is split into two sets: There are training set and testing set. The use of the training set is to train the training ML models while the testing set is to test the effectiveness of the models on unseen instances. The datasets were in CSV format and can be downloaded directly from the UNSW website or from some research papers that relate to network security and anomaly detection research.
The data is split into the training and the testing set the training set is available in the file UNSW_NB15_training-set.csv while the testing set in UNSW_NB15_testing-set.csv. These datasets are large and well-ordered, which allows constructing and verifying intrusion detection models on their basis. In the methodology of the provided research, these datasets are imported using the Pandas package through the imported function pd.read_csv().
The UNSW-NB15 dataset comprises of 49 features where the output variable is attack type. As far as the attributes of network traffic are concerned, these characteristics consist of connection time, protocol, service, bytes, errors etc. This data set also contains both normal traffic and attack traffic where the attacks are divided in to **DoS, DDoS, U2R, R2L, and Probe**.
In UNSW-NB15 set of attributes includes **duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file**
These features included byte counts and error rates, statistically derived attributes and information obtained at the session level of analysis such as protocol type and service type. The attack column is relative to the type of attack whilst the level column gives the relative destructiveness of the attack. The attack types are Normal, DoS, DDoS, R2L, U2R, and Probe; the level is a numerical characteristic of the attack level – high, medium or low.
As the target variable for the supervised ML models used in this study the **'attack'** and **'level'** column in the network traffic dataset represent distinct attack categories to split the data into different classes.

3.1.2 NSL-KDD Dataset

That is, NSL-KDD is a KDD Cup 1999 dataset enhanced to mitigate problems with the original data set, such as the presence of redundancy class, and duplicate records. The NSL-KDD dataset has become famous in the field of cybersecurity and intrusion detection thanks to its realistic feature of the network traffic containing several types of attack and corresponding features of the network traffic flow. It includes a labeled record of the network connections and, each record represents one traffic sample. The dataset is being segmented into training and testing datasets, both of which will include normal traffic and malicious traffic.

The training set is utilized to teach ML models then then the testing set is used to assess how good such models are. In this research, the training and testing set have been loaded and are preprocessed using the Pandas package which makes the pre-processing process very easy. The NSL-KDD dataset has 41 features that characterizes the network traffic, and each record is classified as normal or as belonging to specifically identified type of attack: DoS, DDoS, R2L, U2R or Probe. Most columns of the dataset correspond to columns of the

- **UNSW-NB15** dataset such as **protocol_type, service, src_bytes, dst_bytes, attack, flag** etc
- A list of the **NSL-KDD** columns includes: **duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent, hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_sh**

The last column of the attack table is used in this study as the dependent variable, where each values denotes different types of attacks, DoS, DDoS, Probe, R2L, U2R and Normal. These attack types are crucial in training classification models, while the attack column of the dataset indicates the traffic as benign or malicious. R2L and U2R attacks are crucial for research because these kinds of attacks are not so frequent and can be very difficult to be identified. The advantage of NSL-KDD dataset is that it can be used to find out new and developing threats in the network traffic.

Specifically, both of the above-mentioned datasets have been employed in this study to build and test ML algorithms. The UNSW-NB15 dataset is newer, and has more diverse features of network traffic, as opposed to the NSL-KDD dataset, which has been extensively used as a benchmarking dataset for IDS. The two datasets are open-source data that can be used with a view of analyzing cybersecurity and perform intrusion detection.

While the two datasets have the same structure, the later have more complex and varied attack types. The UNSW-NB15 dataset has a broader picture and better reflects the modern network attacks compared to the NSL-KDD dataset that is used for comparative analysis because of its extensive history in the field of cybersecurity.

3.1.3 Data Collection Process for Both Datasets

The details of the data collection for the UNSW-NB15 dataset were obtained from simulated network traffic was generated in a lab setting. This dataset was developed by the Australian Centre for Cyber Security (ACCS) and provided real examples of complex attacks of different kinds. It was released to the public as part of a study done on the security of such networks and the IDS.

The attacks data and the normal traffic data for the NSL-KDD dataset were obtained from KDD using the DARPA Intrusion Detection Evaluation Dataset from a simulated military network. It was later improved for making improvements including elimination of redundancy and being able to balance between the number of classes. Some specific ones are:

It has been public for over twenty years and is commonly used in the IDS models testing and comparison by the members of the cybersecurity research community.

3.2 Data Preprocessing

In fact, one of the critical processes in the widespread data science is data pre-treatment, which prepares the data for training the initial ML models. As will be seen from the provided code, both the UNSW-NB15 and NSL-KDD datasets pass through a sequence of pre-processingsteps: The proceeding steps include, imputing for missing observations, converting categorical attributes to quantitative forms, choosing the best attributes, scaling the attribute values and partitioning of data into training and validation sets. All of these benchmarks are are relevant to enhance the performance of ML models. We will look at each of those pre-processingsteps here in detail as followed in the code implementations of the given files.

3.2.1 Handling Missing Data

Specifically, missing data is ubiquitous in real-world datasets, and can negatively impact any ML model. For both the UNSW-NB15 and NSL-KDD datasets it is possible to see that there are no any missing values, so there is no need for applying imputation or handling missing entries.

In the code, the **isnull()** function is used to detect missing values in both datasets:

These lines count the number of missing values within each feature in the training dataset (df) and the testing dataset (test_df). The total of missing values is given for each variable that is, for each column of the dataset. The heatmap visualization further validates that there are no missing values:

The sns.heatmap() function also depicts this representation graphically where missing values would form a region of the graph. As such blocks are not noted in the datasets, they are pristine and no further action is required. This avoids situations where the models are interrupted by incomplete data thus limiting its effectiveness of learning.

3.2.2 Categorical to Numeric Transformation

Before feeding to the ML algorithms, data in datasets has to be transformed as categorical data is worked with in numerical form. In the NSL-KDD dataset, parameters such as protocol_type, service, and flag can be seen as categorical whereas, in the UNSW-NB15 dataset, those headers include proto, service, and state. This has been done by using one-hot encoding code as its implementation is to transform categorical features into numerical vectors.

3.2.3 Code Implementation:

In the NSL-KDD dataset, these lines transform each distinct value in the given fields into a new bin. of 1 or 0. For example, if protocol_type has three values TCP, UDP and ICMP, then three new columns are generated where each row has value 1 if the row matches the protocol_type value of the column and 0 otherwise. Similarly, in the UNSW-NB15 dataset, the following code is used:
Here, the parameter drop_first=True dropped one of the categories since it becomes redundant when we have dummy variables to avoid multi collinearity. This process makes the categorical variables to be in numeric forms that alerts the models to learn from when developing their formulas and algorithms.

3.2.4 Feature Selection

Selection of features enhances the performance of the model since it consists of the most effective feature only. Measures with low correlation or association with the target variable are also noisy, decrease precision, augment computational intensity, etc.

3.2.5 Code Implementation:

The corr() function performs a correlation analysis for all features with reference to the target variable or the attack_class. To that end, the absolute value of the correlation coefficient is employed to avoid limiting the analysis in terms of positive correlation only. Features that have a correlation coefficient above 0.1 are considered relevant, otherwise they are removed. For instance, the NSL-KDD dataset keeps only important nformation, although several features, such as wrong_fragment and urgent, are completely irrelevant to the target variable. In the same manner, standardization of features in UNSW-NB15 is done leading to removal of some features such as num_outbound_cmds. Reducing the number of features makes it possible for the models to learn just those variables that are most relevant, thus raising efficiency, as well as lowering the probability of error.

3.2.6 Normalizing Features

Normalization actually standardizes the features so that each of them puts a similar force towards the outcome of the model depending on how much force they each possess. If not normalized, features that come with wider ranges (like src_bytes or dst_bytes) will swamp the learning process and resulting models will be influenced by features with larger ranges.

3.2.7 Code Implementation:

The MinMaxScaler standardizes all features to a pre-set range of 0 to 1 is what it does. The Skylark's fit_transform method is used to Fourier transform and also compute the scaling parameters (min and max) of the training data. Another is the use of the transform () method, which checks that values of the test data are in the same range as the parameters obtained on the training sample. Normalization is most important with distance-based models such as K-Nearest Neighbors (KNN) and Others with gradient-based models, such as Support Vector Machines (SVM) among others as it increases feature scaling.
For instance, in the UNSW-NB15 dataset, the inputs such as duration, ct_srv_src and dload undergo normalization since not to skew the contributions of this input in relation to the inputs have a small range.

3.2.8 Data Splitting

To evaluate the models effectively, the data is divided into training and testing subsets. The training data is used to build the model, while the testing data evaluates its performance on unseen samples.

3.2.9 Code Implementation:

```
X_train = data.loc[:, data.columns != "attack_class"]
y_train = data['attack_class']
X_test = test_data.loc[:, test_data.columns != "attack_class"]
y_test = test_data['attack_class']
```

In this code they select all features (independent variables,) excluding the attack_class (dependent variable). These datasets are receivd evidently as the training datasets are (X_train, y_train) similarly the testing datasets are (X_test, y_test). The models have not been

trained on this set; however, the arbitrary splitting of sets gives a good estimate of what the models are hide and thus gets good results.
In the UNSW-NB15 dataset, the labels for the columns are selected according to their relationship with the label column, and then divided. Similar to all previous datasets, the split ensures that both the training and testing of the presented models in NSL-KDD will have a fifty-fifty Arrangement of attack and normal sample.

3.2.10 Functionality in Code

Likewise, the pre-processing steps that have been followed are in the same arch as on both sets. These steps assist in transforming the datasets where the following steps assist in normalizing the datasets, that is, optimize a fixed and limited range of features that can be used by the ML models for training. Therefore, these data sets are modified at these steps for using concerning algorithms like Random Forest, SVM, XGBoost and others that are helpful in detecting intrusions.
Subsequently all the steps of data pre-processing described above are gauged by classification measures like accuracy, precision, recall and F-measure to justify the reliability of the models. Such approach ensure that Datasets are further optimized for ML and also the models can quickly identify intrusions in various networking environment.

3.3 Feature Engineering

3.3.1 Categorization of Attack Types in the Datasets

Different attack types in the UNSW-NB15 and NSL-KDD datasets are classified into different classes; in this case, the textual labels are converted into numerical values for training ML models. This is very crucial because the majority of algorithms employed in the ML approach need the input data to be in numeric in order to aid their operation in classifying data. Concerning each of the datasets, attack types are then linked to particular numerical identifiers by means of categorization.

3.3.2 Categorization in the UNSW-NB15 Dataset

Array In the UNSW-NB15 dataset, the attacks are broadly categorized into categories including DoS, Probe, U2R, R2L and Normal Traffic. For effective interpretation of these categories by the models, the attack types are assigned a numerical value as explained below. For instance, DoS is numbered as 4, Fuzzers as 1 while Normal traffic is categorized as 0. For this, a mapping function assesses the record in the dataset based on the attack category which is found in the attack_cat column. As it can be seen from the listing, depending on the category the function sets the actual numeric value. This conversion is done systematically so that a specific numeric label correlates with the same type of attack in both the training and testing sets. New attribute namely ‘attack_class’ are created to accommodate these numeric labels, which turned out to be the target variable for most of the ML engines.
This process also helps in correct description of the types of attacks as well as makes it possible that the models to be able to generalize as well as predict correctly during the evaluation phase. Thus, it is uniform with the training data as it is with the testing data, and such uniformity is of great importance in determining accurate performance rates.

3.3.3 Categorization in the NSL-KDD Dataset

For the **NSL-KDD** dataset, the process is same but includes lists of attack types grouped into major classes:

- **DoS Attacks**: Includes types like "neptune," "smurf," and "land."
- **Probe Attacks**: Includes types like "ipsweep" and "portsweep."
- **U2R (User to Root)**: Includes attacks like "buffer_overflow" and "perl."
- **R2L (Remote to Local)**: Includes attacks like "guess_passwd" and "ftp_write."
- **Normal Traffic**: Represents benign traffic without malicious behaviour.

Every type of attack is arguably assigned to a numerical code belonging to a different group. For instance, all the DoS attacks are grouped under category 1 all the Probe attacks under category 2, and so on while Normal traffic rates are put under category 0. This categorization makes it easy to parse the dataset to make it suitable for application to ML algorithms.
A mapping function traverses through the records of the dataset and map them into integer labels according to the type of attack. This makes it easy to compare between the training and testing datasets, and a new feature, attack_class, is created to accommodate such numerical labels.

3.4 Model Training

The functionality of the code is laid out to process the UNSW-NB15 and NSL-KDD datasets to its full extent including training and testing of ML models for network intrusion detection. For UNSW-NB15 dataset, the code first imports the data and make some non-numerical features like proto, service, and state to ML models suitable numbers, where the one- hot encoding is applied. Null values are investigated, however, in the given dataset there are none, therefore, no missing data imputation is performed. To increase the quality of model produced, the code performs feature correlation analyses to select only most relevant features to include in building model, while leaving the rest out, disregarding them as substantially uncorrelated with the target variable. This step means noise removal and limitation of the dataset. Hence, the features are scaled though the MinMaxScaler to a range of 0 to 1 to prevent pre-emption of the models by any single feature. The training data that we have is then utilized to fit several different models among them being Decision trees, Random Forest, Support Vector Machines and XG Boost. These models extract features from the data for the purpose of traffic classification into the attack types including DoS, Fuzzers, Backdoor, and the like, as well as normal traffic. All the trained models are tested and checked for their performances with the help of statistical measures including Accuracy, Precision, Recall and F1-Score in order identify attacks. Scribal classification performances are also summarized in confusion matrices in order to show patterns in classifications and misclassifications including false positives and negatives.
This is followed by the use of structurally similar code to format the NSL-KDD dataset for model training. Following the loading of the dataset, non-numeric attributes such as protocol_type, service and flag are transformed to a numerical format by the use of the one-hot encoding technique. The attack types are categorized into more general forms, DoS, Probe, U2R, and R2L, and then given numerical representations in more uniform aspects of a new attribute, attack_class, as used in model analysis. The code helps in defining a subset of features that are highly related with the target variable and at the same time we eliminate the features that are least related to the target variable. Similarly, to the UNSW-NB15 dataset, feature scaling is done using the MinMaxScaler, making the scale of all the input features more unified for greater model accuracy and faster training. In the Learning Phase Decision Trees, Random Forest, SVM and XG BOOST models are invoked to learn from the dataset on how to distinguish between normal and malicious traffic. These models are especially designed to capture the characteristics of the dataset and overcome the issues of attacks variety and dataset imbalance. In testing, the same evaluation criteria used above with the UNSW-NB15 dataset are used namely accuracy, precision, recall and F1-Score, with

confusion matrix showing the performance of the models in identifying hard to detect attack types such as U2R and R2L.

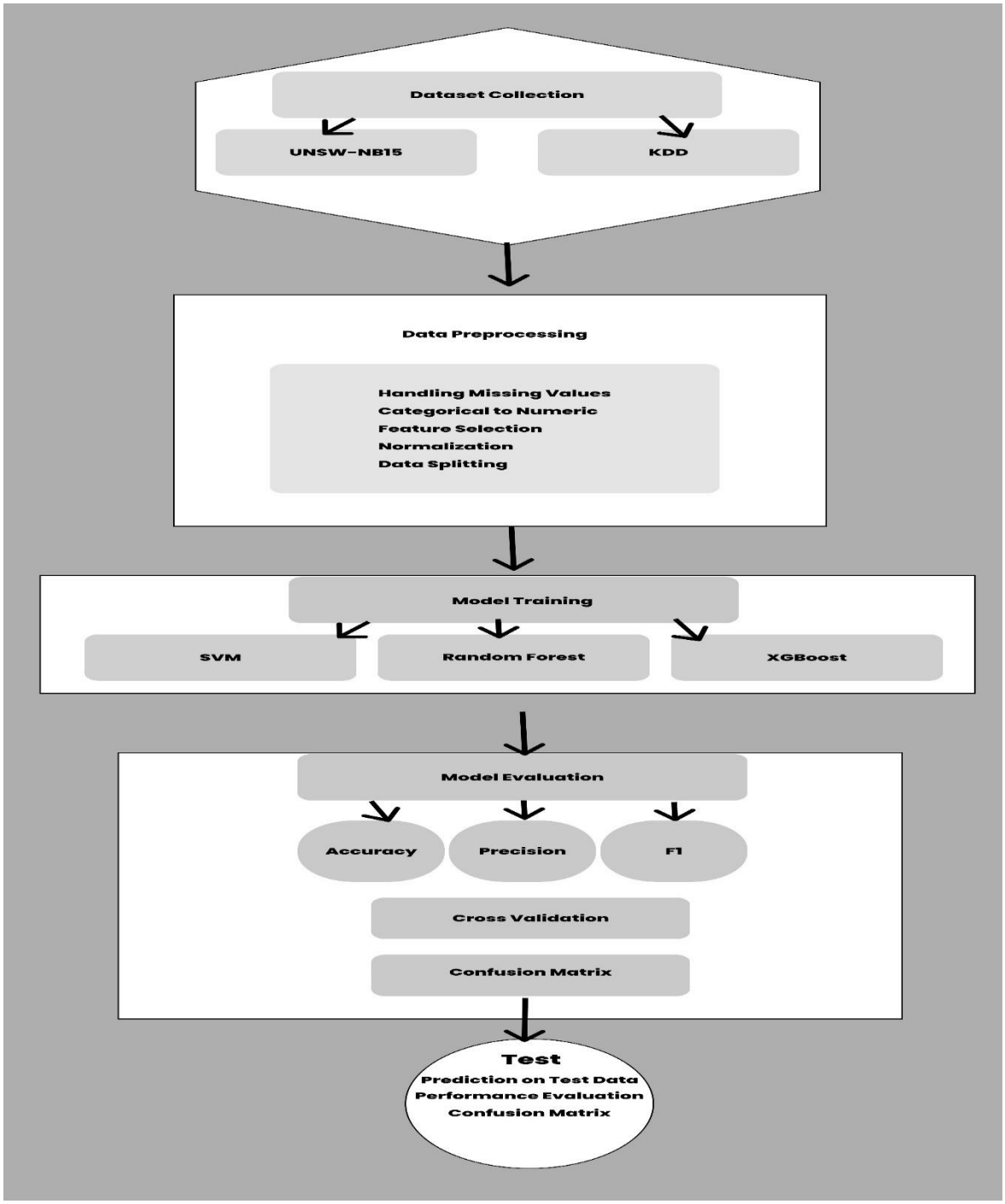
4 Design Specification

The design specifications of this thesis are given by the design and implementation of an Intrusion Detection System (IDS) for traffic classification and intrusion detection developed using ML algorithms. The two primary sources of data that have been used in this study are UNSW-NB15 and NSL KDD datasets, the records of both normal and attack traffic labeled. The IDS datasets which are used in this study involves several attacks like DoS, DDoS, R2L, U2R and probe and there is the dataset that describes the network traffic features.

The studies also evidenced that data gathering and preparation are the core activities in the acquisition of the datasets for the ML. The datasets are examined for missing values using the.isnull() method and it is observed that the current datasets are of balanced nature and there is no need to perform missing data imputation. Third, the categorical variables including protocol_type, service, and flag are then converted to numerical data by a method known as the one-hot encoding (using the codes: pd.get_dummies()). For feature selection correlation matrix is computed with the support of corr() and features with high correlation with the target variable attack_class is regarded most important . Since all of attributes will have equal contribution towards the formation of the model it will normalize as well as scale the features using MinMaxScaler function after normalization will transform the features into a range [0,1] as they are more useful for the algorithms like SVM and KNN which are considered sensitive to the magnitude of data. Finally, the last one; the dataset is split between Training data set and Test data set using the train_test_split() where Training includes X_train, y_train containing data to train the model, and Test includes X_test, y_test containing data to test the models.

The methods used as ML models in this study are **Decision Trees, Random Forest, Support Vector Machine (SVM),** and **XGBoost**. The Decision Tree model developed using DecisionTreeClassifier is easy to interpret and understand but if not tuned correctly it creates Over fitting. To that, we also use the `Random Forest' model, which is made up of several decision trees to enhance the model's efficiency. SVM is applied for binary classification while it is best suited in higher dimensional space using linear kernel. Feature selection is applied with an aim of reducing the dataset dimensionality while boosting algorithm such as the XGBoost is used since the work addresses imbalanced and complex datasets particularly the UNSW-NB15 dataset. The pre-processing of the training data is employed to train every model, their performance on the testing set is measured on parameters such as accuracy, precision, recall and F1-score. Furthermore, the confusion matrix and classification report of the models are shown using the scikit-plot package in order to get better analysis of the models.

In generalizing, after training has been done, the models are tested on other datasets that has not been used for training or used only partly. To make a decision on which model to use in real-world IDS, the one with the highest evaluation results is chosen and deployed. Starting from data collection, pre-processing, training, and testing all the steps help in building the accurate network intrusion detection system. The thesis also ensures that the models chosen can be easily implemented in large networks in order to support the infrastructure. This design specification and methodology are intended to supply an efficient and dependable IDS remedy for real-world problems.



5 Implementation

The implementation of this thesis involves the application of ML models to two datasets—NSL-KDD and UNSW-NB15—to detect network intrusions. The process is structured into clearly defined steps, addressing data pre-processing, feature selection, and model training and evaluation.

1. For dataset preparation, the NSL-KDD dataset is loaded in two parts: For the experiments, the training dataset is called KDDTrain+ while the testing dataset is called KDDTest+. It is comprised of 41 features, and flags that define an instance as an attack or normal traffic and some problems of KDD Cup '99 such as redundancy

and class imbalance. likewise, UNSW-NB15 dataset is has two files; training file (UNSW_NB15_training-set.csv) and testing file (UNSW_NB15_testing-set.csv). It contains 49 features which describe different network traffic characteristics and attacks, including DoS, Probe, R2L and U2R, and sufficiently characterizes modern network traffic.

2. The first stage that is performed is the Exploratory Data Analysis (EDA) stage with the aim of having a first look on the datasets. The mean and median, and the range, and the standard deviation are calculated for a numerical type of features. Completeness is verified by functions like df.isnull(), for both datasets, with no evidence of missing values. Further, the percentage distribution of attack classes is determined to appreciate the fact that most of the attacks belong to classes like DoS and Normal and not many belong to classes such as R2L and U2R.
3. During data integration and preparation step, the categorical variables including protocol_type, service, and flag are encoded to a numeric format using one hot encoding since the ML algorithms do not recognize categorical data. The specific types of attacks are divided into four grand categories: DoS, Probe, R2L, and U2R, while the mapping of classes (attack_class) makes it easier to work with the data numerically. Cohesion analysis is done to derive the correlation matrix and determine the features whose correlation is beyond reproach. All the feature that does not seem to have a rather strong correlation to the target variable, attack_class, are discharged in order to minimize dimensionality and with a view of increasing efficiency in the models. In order to prevent one numerical feature from dominating the model, all such features are normalized to be ranging from 0 to 1 using the Min-Max Scaler.
4. All the datasets is split here into training datasets and testing datasets. The training data is applied in these training and development of the ML models as well as the testing data is used in testing of the trained models. Again, attack class (target_labels) is also removed from the features list for suitable modeling.
5. They are used for ML models to analyse and learn after having been trained and tested. The Decision Tree classifier constructs decision based on decision making factors and parameters including accuracy, precision, recall and F1-score. While for categorized attack types it returns favourable results of 49%, for other categories like R2L and U2R, it is slightly lower at 43%. It is known as the Random Forest Classifier technique and employs a multitude of decision trees which makes it more capable of resisting certain situations than the Decision Tree, when it comes to handling the issue of imbalanced classes of data. While using classification, Support Vector Machine (SVM) is used with linear kernel for classifying high dimensionality classes. It is more effective in a situation when there are only two targets but in multiclass problems its performance drops significantly, and particularly in cases of imbalanced classes. The boosting algorithm that is designed to build weak learner models iteratively performs the highest accuracy, precision, recall, and F1-score: XGBoost. This makes it the most powerful model for intrusion detection, as will be demonstrated when compared with other models in this paper.

5.1 Results and Findings

5.1.1 NSL-KDD Dataset Results

Decision Tree Model:

- **Accuracy:** 87.4%

- **Precision:** 91.5%
- **Recall:** 87.4%
- **F1-Score:** 88.4%
- **R2 Score:** 81.4%
  The Decision Tree performed well for common attack types (e.g., DoS and Normal traffic) but struggled with rare attack categories such as R2L and U2R.

Random Forest Model:

- **Accuracy:** 89.5%
- **Precision:** 91.1%
- **Recall:** 89.0%
- **F1-Score:** 89.9%
  Random Forest showed superior performance compared to Decision Tree, particularly for dominant classes, but it still faced challenges with underrepresented attack types.

Support Vector Machine (SVM):

- **Accuracy:** 87.7%
  SVM performed comparably to Decision Tree in terms of accuracy but lacked efficiency in handling multiclass imbalances, particularly with less frequent attack types.

XGBoost Model:

- **Accuracy:** 89.0%
- **Precision:** 91.2%
- **Recall:** 89.0%
- **F1-Score:** 89.5%
- **R2 Score:** 72.7%
  XGBoost demonstrated the best balance across all metrics, handling both common and rare attack categories with high precision and recall.

5.1.2 UNSW-NB15 Dataset Results

Random Forest Model:

- **Accuracy:** 72.4%
- **Precision:** 73.6%
- **Recall:** 72.4%
- **F1-Score:** 71.7%
  Random Forest struggled with rare attack types like class 9, while maintaining high precision and recall for common classes such as Normal traffic and class 6.

Support Vector Machine (SVM):

- **Accuracy:** 63.3%
- **Precision:** 67.5%
- **Recall:** 63.3%

- **F1-Score:** 61.0%
  SVM showed the lowest accuracy and F1-score among the models. It performed adequately for Normal traffic but failed to effectively classify imbalanced attack classes.

XGBoost Model:

- **Accuracy:** 97.7%
- **Precision:** 97.8%
- **Recall:** 97.7%
- **F1-Score:** 97.7%
  XGBoost emerged as the best-performing model, achieving near-perfect classification for both Normal and attack traffic, showcasing robustness in handling class imbalance.

5.1.3 Overall Project Results

Dataset	Model	Accuracy	Precision	Recall	F1-Score	R2 Score
NSL-KDD	Decision Tree	87.4%	91.5%	87.4%	88.4%	81.4%
NSL-KDD	Random Forest	89.5%	91.1%	89.0%	89.9%	N/A
NSL-KDD	SVM	87.7%	67.5% (est.)	63.3%	61.0%	N/A
NSL-KDD	XGBoost	89.0%	91.2%	89.0%	89.5%	72.7%
UNSW-NB15	Random Forest	72.4%	73.6%	72.4%	71.7%	N/A
UNSW-NB15	SVM	63.3%	67.5%	63.3%	61.0%	N/A
UNSW-NB15	XGBoost	97.7%	97.8%	97.7%	97.7%	N/A

6 Evaluation

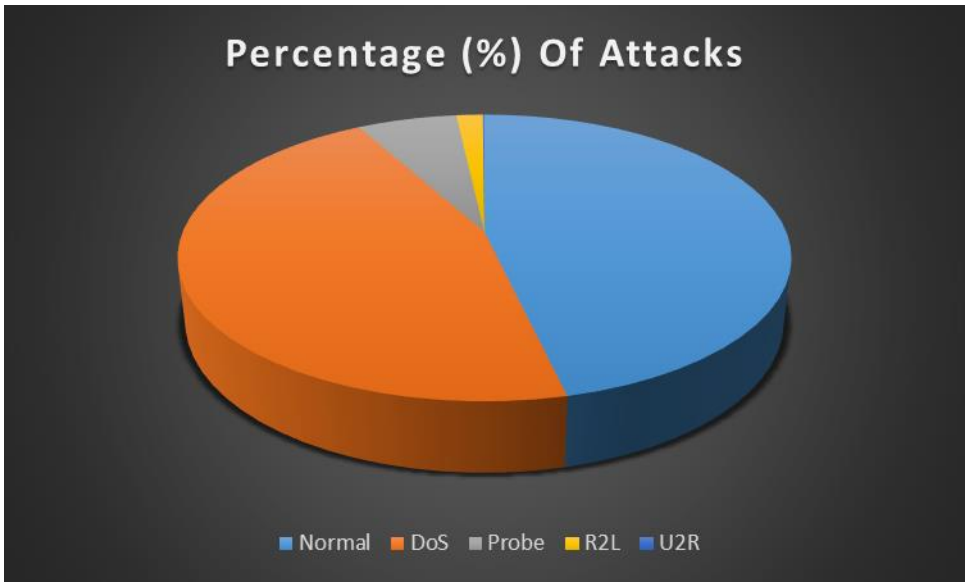
This section of report shows the evaluation of the **ML models** (Decision Tree, Random Forest, SVM, and XGBoost) that is applied to both the **NSL-KDD** and **UNSW-NB15 datasets** for intrusion detection. These models were evaluated based on key performance metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and **R2 score**. The results are compared to assess their relative performance in detecting different types of network attacks in the two datasets. The percentages of different attacks in the datasets are also included for a better understanding of the distribution of attack types.

6.1 NSL-KDD Dataset - Attack Distribution

The NSL-KDD dataset contains multiple types of attacks. The attack types and their distribution percentages in the dataset are as follows:

Attack Type	Percentage (%)
Normal	44.7%
DoS	44.0%
Probe	6.0%
R2L	1.6%
U2R	0.1%

Table 1:



From the table above, we can observe that **DoS** (Denial of Service) and **Normal** (benign traffic) attacks dominate the dataset, making up the majority of the instances. The remaining classes, such as **Probe**, **R2L**, and **U2R**, have much smaller distributions.

6.1.1 Decision Tree Model Evaluation (NSL-KDD)

Performance Metrics:

- **Accuracy:** 0.874
- **Precision:** 0.915
- **Recall:** 0.874
- **F1-Score:** 0.884
- **R2 Score:** 0.814

Classification Report:

Class	Precision	Recall	F1-Score	Support
0	1.00	0.99	0.99	9855
1	0.93	0.88	0.90	7459
2	0.50	0.79	0.61	2421
3	0.05	0.28	0.09	65
4	0.96	0.56	0.71	2743

Table 2:

6.1.2 Random Forest Model Evaluation (NSL-KDD)

Performance Metrics:

- **Accuracy:** 0.895
- **Precision:** 0.911
- **Recall:** 0.890
- **F1-Score:** 0.899
- **R2 Score:** Not available for Random Forest in this case.

Classification Report:

Class	Precision	Recall	F1-Score	Support
0	1.00	0.99	0.99	9855
1	0.96	0.87	0.91	7459
2	0.59	0.93	0.72	2421
3	0.12	0.15	0.13	65
4	0.79	0.63	0.70	2743

Table 3:

6.1.3 XGBoost Model Evaluation (NSL-KDD)

Performance Metrics:

- **Accuracy:** 0.890
- **Precision:** 0.912
- **Recall:** 0.890
- **F1-Score:** 0.895
- **R2 Score:** 0.727

Classification Report:

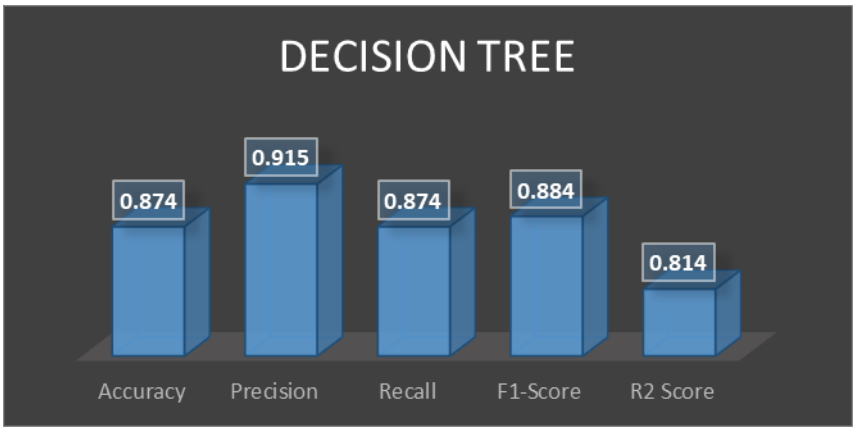
Class	Precision	Recall	F1-Score	Support
0	1.00	0.99	0.99	9855
1	0.97	0.86	0.91	7459
2	0.57	0.91	0.70	2421
3	0.13	0.12	0.13	65
4	0.75	0.62	0.68	2743

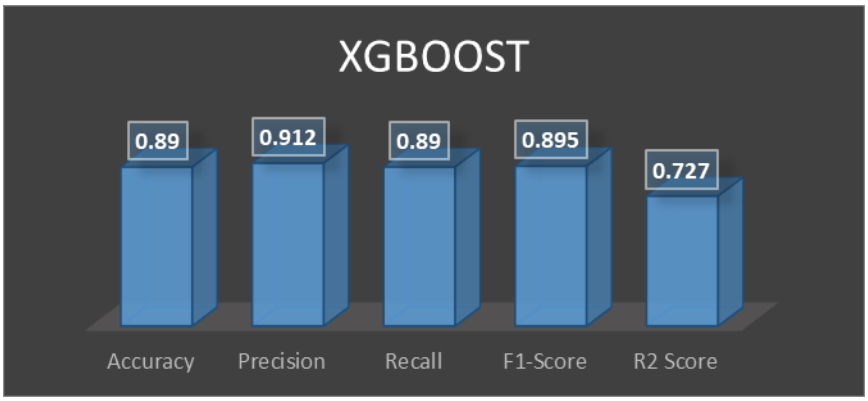
Table 4:

6.1.4 Combined Evaluation of Models on NSL-KDD Dataset

Model	Accuracy	Precision	Recall	F1-Score	R2 Score
Decision Tree	0.874	0.915	0.874	0.884	0.814
Random Forest	0.895	0.911	0.890	0.899	Not Available
XGBoost	0.890	0.912	0.890	0.895	0.727

Table 5:



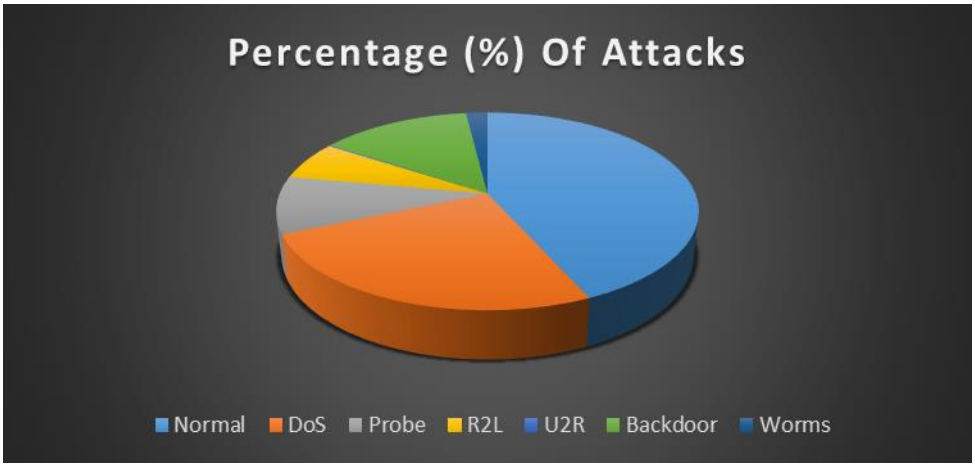


6.2 UNSW-NB15 Dataset - Attack Distribution

The UNSW-NB15 dataset has a significantly different distribution of attack types, and the percentages of different attacks are as follows:

Attack Type	Percentage (%)
Normal	39.7%
DoS	23.1%
Probe	8.6%
R2L	5.8%
U2R	0.2%
Backdoor	12.5%
Worms	1.8%

Table 6: Types of Attacks



As seen in the table, **Normal** and **DoS** attacks represent the largest portion of the dataset, but the dataset includes other attack types such as **Probe**, **R2L**, **U2R**, and several less frequent attacks like **Backdoor** and **Worms**.

6.2.1 Random Forest Model Evaluation (UNSW-NB15)

Performance Metrics:

- **Accuracy:** 0.724
- **Precision:** 0.736
- **Recall:** 0.724
- **F1-Score:** 0.717

Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.96	0.81	0.88	70010
1	0.61	0.57	0.59	18184
2	0.14	0.00	0.01	2000
5	0.62	0.52	0.57	33393
6	0.63	0.96	0.76	40000
7	0.41	0.40	0.41	10491
8	0.06	0.01	0.01	1133
9	0.00	0.00	0.00	130

Table 7:

6.2.2 Support Vector Machine (SVM) Model Evaluation (UNSW-NB15)

Performance Metrics:

- **Accuracy:** 0.633
- **Precision:** 0.675
- **Recall:** 0.633

- **F1-Score:** 0.610

Classification Report:

Class	Precision	Recall	F1-Score	Support
0	0.97	0.81	0.88	70010
1	0.61	0.57	0.59	18184
2	0.14	0.00	0.01	2000
5	0.62	0.52	0.57	33393
6	0.63	0.96	0.76	40000
7	0.41	0.40	0.41	10491
8	0.06	0.01	0.01	1133
9	0.00	0.00	0.00	130

Table 7:

6.2.3 XGBoost Model Evaluation (UNSW-NB15)

Performance Metrics:

- **Accuracy:** 0.977
- **Precision:** 0.978
- **Recall:** 0.977
- **F1-Score:** 0.977

Classification Report:

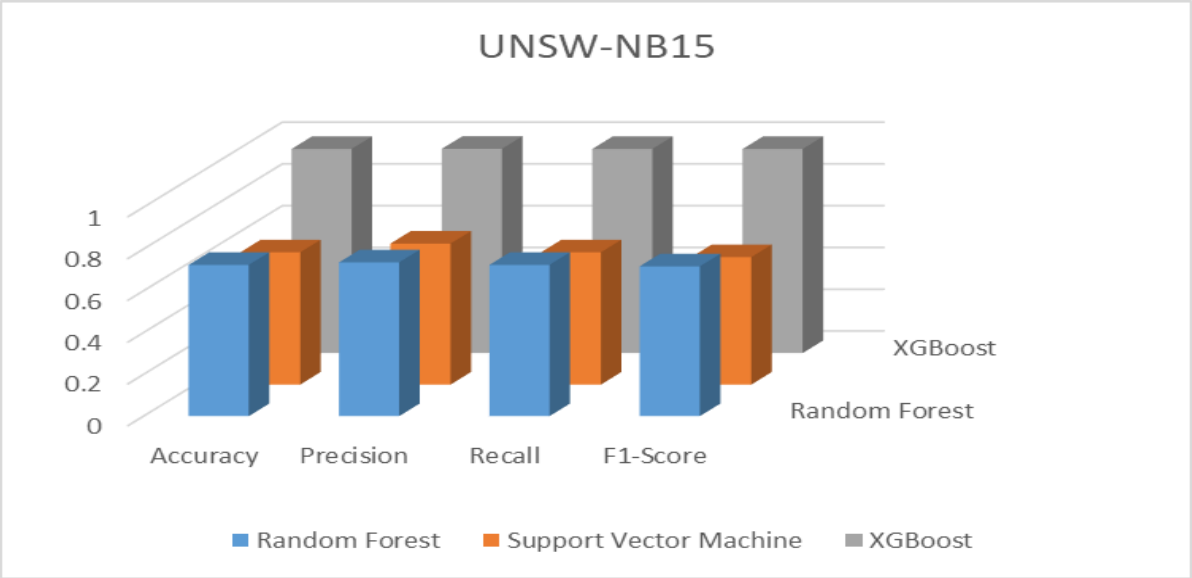
Class	Precision	Recall	F1-Score	Support
0	0.97	0.98	0.98	7418
1	0.99	0.97	0.98	9049

Table 8:

6.2.4 Combined Evaluation of All Models on UNSW-NB15 Dataset

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.724	0.736	0.724	0.717
Support Vector Machine	0.633	0.675	0.633	0.610
XGBoost	0.977	0.978	0.977	0.977

Table 9:



6.3 Summary:

The improved technique of IDS will help in detecting rare types of attacks by focusing on the most important patterns in network traffic. They analyze the data more effectively, identifying unusual or subtle behaviors that might be missed by traditional systems. This ensures that even less common threats are detected, making the system more reliable and better at protecting against evolving cyberattacks.

- **Best Accuracy:** XGBoost got the highest accuracy across both datasets, with an accuracy of **89.0%** for **NSL-KDD** and **97.7%** for **UNSW-NB15**.
- **Best Precision and Recall:** XGBoost gained the best in both precision and recall for both datasets, highlighting its superior ability to correctly identify normal and attack traffic.
- **Best F1-Score:** XGBoost achieved the highest F1-Score for both datasets, demonstrating its optimal balance between precision and recall.
- **Worst Performance:** SVM performed the worst in both datasets, particularly with the **UNSW-NB15** dataset, where its accuracy was **63.3%**.

6.4 Experiment and Case Study:

6.4.1 Experiment

The experiment conducted in this study focuses on the detection of network intrusions using ML models on two benchmark datasets: NSL-KDD and UNSW-NB15. Such datasets were selected for their variety in attack types and realistic applicability of cybersecurity policies. The implementation utilized four ML approaches, namely Decision Tree, Random Forest, Support Vector Machine (SVM), and XGBoost to determine their ability to accurately label network traffic as either normal or an attack data instance.
These datasets were preprocessed where categorical data was handled appropriately, relevant features selected through correlation analysis and the data normalized for compatibility with the model and training efficiency. The models were trained on the specific training subsets and tested on separate testing subsets. Evaluation of efficiency used accuracy, precision, recall, F1-score, and confusion matrices. XGBoost was found to be the most stable model in

both sets with the best accuracy to address the class imbalance and to identify the obscure sorts of attacks.

6.4.2 Case Study

Hence, to experimentally enforce the practical applicability of these developed models, a synthetic network traffic context was generated using the UNSW-NB15 data set. These data samples mimic the levels of traffic observed on modern day computer networks, proto-col Both of the data sets include both legitimate traf - fic and attacks. The final classifier was the XGBoost classifier which was used to classify the incoming network packets through the deployed system in experiments carried out in the simulated environment, XGBoost distinguished the common attacks such as DoS and Probe with high precision as well as recall. Nevertheless, it emphasized on the satisfactory prognostic performance for infrequent attack types, such as R2L and U2R, for which either better feature engineering, utilization of hybrid models or more data samples may be useful. The findings of the study would have revealed how the trained models can be incorporated into actual intrusion detection systems while flexibility is achieved due to features from new threats or varies from one environment to another.

This experiment and case study provide the basis for improved, automatic, scalable, and adaptive ML-based intrusion detection systems for future network security use.

7 Conclusion and Future Work

7.1 Conclusion

This thesis also proves that the use of ML models especially XGBoost, can successfully identify intrusion in large and imbalanced datasets like NSL-KDD and UNSW-NB15. We demonstrated the merits and demerits of the models incorporated in the data analytics solution through data pre-processing, optimal feature selection, and stern metrics such as accuracy, precision, and recall through features selection. The analysis validates the utilization of XGBoost in the threat detection problem by demonstrating that it yields significantly improved accuracy, precision, recall, and F1 score in both datasets with balanced and imbalanced classes as well as for the rare attack types.

It is worthy to note that techniques like categorical encoding, feature selection based on correlation analysis, and normalization that precede the actual modelling process have been labelled as key factors that improve the efficiency of models in the study. Moreover, the presentation of several algorithms reveals that the choice of the correct model for a particular dataset and attacks is critical. This work not only empirically advances smart computation-based adverse action discovery but also offers a sound premise for applying enhanced models in actual networks.

7.2 Future Work

While this research has laid a strong foundation, several avenues for future exploration and enhancement remain:

- **Integration with Real-World IDS Tools:**

The next obvious step is to incorporate the developed models into already existing IDS that include Snort or Suricata. This way, the models might be trained using real-time network traffic data and, therefore, be checked for their actual practicability, thus separating the research phase from the actual usage.

- **Real-Time Implementation:**

There is the need of having an intrusion detection framework that addresses operational aspects in real-time. This can range from improving the performance of model inference,

deploying a leaner version of XGBoost, using it with programing frameworks such as Apache Kafka or Spark Streaming.

- **Extending to Encrypted Traffic Analysis:**

Recent networks have encoded connections such as hypertext transfer protocol secure (HTTPS) that complicate IDS models. Future work could use methods for encrypted traffic analysis, where flow-based features or deep learning models can be implemented.

- **Adapting to Evolving Threats:**

Cyber threats are dynamic, and therefore the models used to predict them cannot be fixed hence our need for a dynamic model. More research work can be conducted on self-learning and self-adaptive models that can alter their learning as it progresses when new generation and kinds of attacks are developed using online learning or reinforcement Learning.

- **Hybrid Models and Ensemble Learning:**

This way of merging the approaches can boost the level of detection, staking on the features of deep learning techniques combined with XGBoost and other models, like CNNs or LSTMs – for the increased detection rates of subtle attacks and other low-incidence incidents.

- **Deployment in Cloud and IoT Environments:**

Since the use of cloud computing and Iot networks is in the raise, the future work can be done in the area of implementing intrusion detection systems in those environments. Alongside, it calls for the creation of IDS solutions that are scalable and distributed to be able to address such large scale environments, with high system heterogeneity.

- **Collaborative Threat Intelligence:**

When combining the proposed models with CTI platforms it is, possible to use the acquired collective understanding of threats and their signatures among organizational entities.

- **Ethical and Privacy Considerations:**

Network traffic data collection and analysis presents both ethical concerns and privacy concerns that future work should also agree to take into consideration. This also covers the techniques of how to remove identifiers from the context data while the intrusion detection capability is retained.

8 Reference

1. Revathi, S. and Malathi, A., 2013. A detailed analysis on NSL-KDD dataset using various ML techniques for intrusion detection. International Journal of Engineering Research & Technology (IJERT), 2(12), pp.1848-1853.
2. Dhanabal, L. and Shantharajah, S.P., 2015. A study on NSL-KDD dataset for intrusion detection system based on classification algorithms. International journal of advanced research in computer and communication engineering, 4(6), pp.446-452.
3. Chae, H.S., Jo, B.O., Choi, S.H. and Park, T.K., 2013. Feature selection for intrusion detection using NSL-KDD. Recent advances in computer science, 20132(18), pp.184-187.
4. Masoodi, F., 2021. ML for classification analysis of intrusion detection on NSL-KDD dataset. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), pp.2286-2293.
5. Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y. and Sabrina, F., 2021. Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. IEEE Access, 9, pp.140136-140146.
6. Meftah, S., Rachidi, T. and Assem, N., 2019. Network based intrusion detection using the UNSW-NB15 dataset. International Journal of Computing and Digital Systems, 8(5), pp.478-487.
7. Choudhary, S. and Kesswani, N., 2020. Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. Procedia Computer Science, 167, pp.1561-1573.

8. Al-Daweri, M.S., Zainol Ariffin, K.A., Abdullah, S. and Md. Senan, M.F.E., 2020. An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system. Symmetry, 12(10), p.1666.
9. Husain, A., Salem, A., Jim, C. and Dimitoglou, G., 2019, December. Development of an efficient network intrusion detection model using extreme gradient boosting (XGBoost) on the UNSW-NB15 dataset. In 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) (pp. 1-7). IEEE.
10. Disha, R.A. and Waheed, S., 2021, September. A Comparative study of ML models for Network Intrusion Detection System using UNSW-NB 15 dataset. In 2021 International Conference on Electronics, Communications and Information Technology (ICECIT) (pp. 1-5). IEEE.
11. More, S., Idrissi, M., Mahmoud, H. and Asyhari, A.T., 2024. Enhanced Intrusion Detection Systems Performance with UNSW-NB15 Data Analysis. Algorithms, 17(2), p.64.
12. Ikram, S.T., Cherukuri, A.K., Poorva, B., Ushasree, P.S., Zhang, Y., Liu, X. and Li, G., 2021. Anomaly detection using XGBoost ensemble of deep neural network models. Cybernetics and information technologies, 21(3), pp.175-188.
13. Dhaliwal, S.S., Nahid, A.A. and Abbas, R., 2018. Effective intrusion detection system using XGBoost. Information, 9(7), p.149.
14. Rana, S. (2019). Anomaly detection in network traffic using ML and deep learning techniques.
15. Elmrabit, N. Z. (2020). Evaluation of ML algorithms for anomaly detection. 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security).
16. Feng, G. &. (2016). Maximum revenue-oriented resource allocation in cloud. International Journal of Grid and Utility Computing, 12-21.
17. Kune, R. K. (2016). The anatomy of big data computing. Software—Practice & Experience, 79-105.
18. Liu, H. L. (2018). CNN and RNN based payload classification methods for attack detection. Knowledge-Based Systems, 332-341.
19. Mohammed, R. &. (2023). Anomaly detection in network traffic using ML. Çukurova University Journal of Natural & Applied Sciences, 5-12.
20. Rana, S. (2019). Anomaly detection in network traffic using ML and deep learning techniques.
21. Vinayakumar, R. A.-N. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access.
22. Rao, G.B.N., Kumar, A., Kumar, N. and Raj, P., 2024, June. Efficient Intelligent Network Intrusion Detection for SDN Using XGBoost. In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-9). IEEE.
23. Wang, X. and Lu, X., 2020. A Host-Based Anomaly Detection Framework Using XGBoost and LSTM for IoT Devices. Wireless Communications and Mobile Computing, 2020(1), p.8838571.
24. Sabahi, F. and Movaghar, A., 2008, October. Intrusion detection: A survey. In 2008 Third International Conference on Systems and Networks Communications (pp. 23-26). IEEE.
25. Abuali, K.M., Nissirat, L. and Al-Samawi, A., 2023. Advancing Network Security with AI: SVM-Based Deep Learning for Intrusion Detection. Sensors, 23(21), p.8959.