Hochschule Neu-Ulm
University of Applied Sciences

Master Thesis

in the master program

Business Intelligence and Business Analytics

at University of Applied Sciences Neu-Ulm

Developing An Intrusion Detection System Using Machine Learning Methods

Halime Köroğlu

Matriculation Number: 311458

Supervised by

Professor Dr. Achim Dehnert

May 2024 - September 2024

ACKNOWLEDGEMENTS

First of all, I would like to sincerely express my gratitude to my supervisor, Professor Dr. Achim Dehnert for his valuable guidance, encouragement, and insightful feedback throughout the my thesis's process.

I would also like to thank Professors whose valuable knowledge and experience I have benefited from throughout my master education.

Lastly, I am thankful to my family, who have continuously supported and motivated me during this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

IDS:	Intrusion detection systems
IG:	Information Gain
RFE:	Recursive Feature Elimination
RF:	Random Forest
DT:	Decision Tree
XGBoost:	Extreme Gradient Boosting
LightGBM:	Light Gradient Boosting
LG:	Logistic Regression
DOS:	Denial of Service
DDOS:	Distributed Denial of Service
NIDS:	Network-Based Intrusion Detection System
HIDS:	Host Based Intrusion Detection System
CFS:	Correlation-based feature selection
MDI:	Mean Decrease Impurity
ELM:	Extreme learning machine method
CANN:	Cluster center and nearest neighbor
U2R:	User to Root
CIC:	Canadian Institute for Cybersecurity

ABSTRACT

The development of technology and the spread of the internet have provided great benefits and offered innovative solutions in many areas. However, these developments have also brought some security risks. In particular, the increase in cyber attacks and the possibility of data being seized by malicious people have created serious threats for individuals and institutions. This has increased the need for more advanced security measures to protect networks and systems. In this context, intrusion detection systems (IDS) play a critical role in providing an effective defense mechanism against cyber attacks by monitoring network traffic and detecting abnormal behavior. This study aims to develop an effective and high-performance machine learning-based IDS by combining the benefits of pre-processing, feature selection, class balancing and ensemble learning approaches. In the study, the most significant features within the dataset were identified by using Recursive Feature Elimination (RFE), Spearman's Correlation Analysis, and Information Gain (IG) methods. This approach aimed to enhance the efficiency of attack detection process and improve performance by eliminating unnecessary and insignificant features from the dataset.To assess the impact of the feature selection techniques on the performance of the intrusion detection models, experiments were conducted on newly generated sub-datasets using the features selected by each method. Logistic Regression (LG), Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting (LightGBM) classifiers were utilized to compare the results obtained from the new sub-datasets with those obtained from the original dataset. The results showed that RFE improves the performance of all models (LR, DT, RF, XGBoost, and LightGBM) in terms of accuracy, F1-score, and error rate. In the next step of the study, a new ensemble learning model was developed that integrates the advantages of individual classifiers in order to increase accuracy and intrusion detection performance. This new model was constructed using the top three classifiers with the highest performance, selected based on the results of initial phase of the study. To evaluate the model's performance, the dataset comprising 40 features, determined through the RFE method, was employed. Subsequently, the performance results of the new model were compared with those of the individual models. The results of the performance assessment demonstrated that the new model outperformed in intrusion detection compared to the other individual models.

Keywords: intrusion detection system, class balancing, machine learning, feature selection, ensemble learning

INTRODUCTION

As technology advances, the internet has begun to be used in virtually all areas of our lives and has gradually come to be an integral part of our daily lives. In recent research, it is estimated that currently, 5.44 billion people are using the internet globally, which accounts for 67.1% of the total global population (Wearesocial, 2024). As more people use the internet, the amount of data shared online has increased significantly. This growth has raised important concerns, especially regarding information security, data privacy, and data integrity. Along with the rise in internet use and development of digital technology, the diversity and complexity of cyber threats have also increased, and this has brought about serious security risks for institutions and individuals. Intrusion Detection Systems (IDSs) have become very important to minimize these security risks and create an effective defense mechanism against potential cyber-attacks.

IDSs are crucial security mechanisms that apply various approaches to monitor and analyze activities across network resources (Tama & Rhee, 2017). These are typically categorized into two different ways. The first one is host-based and network-based IDSs according to the location of the system. The other one is signature based and anomaly based IDSs according to the intrusion detection approach (Ajiya et al., 2021).

The main objective of IDSs is to concentrate on a particular computer network. They analyze and interpret the traffic within that network to identify any suspicious activities, helping to determine if these activities indicate a potential network attack (Ajiya et al., 2021). The suspicious activities and abnormal traffic occurring in computer networks are referred to as anomalies. The fundamental purpose of the anomaly detection approach is to identify this abnormal traffic and detect the attack.

An IDS is considered effective and successful to the extent that it achieves high classification accuracy and maintains a low false alarm rate (Kasongo & Sun, 2019). At the same time, IDS should be responsive to new types of attacks that emerge alongside rapid changes in the internet environment (Liu & Lang, 2019). Machine learning-based techniques are extensively utilized to develop IDS systems that will meet these requirements.

The algorithms used in these techniques benefits from both real-time and historical data to identify abnormal record that may signify potential intrusion attacks. These algorithms enhance their capability to recognize and comprehend both new and emerging threats through training on various datasets. Machine learning enhances IDS by providing rapid and accurate threat identification, minimizing false positives, and adapting to the dynamic nature of swiftly evolving threats (Jayalaxmi et al.,2022). It enables security system to effectively protect networks and information from unauthorized access and harmful activities (Kafi and Akter, 2023).

On the other hand, conducting effective analysis for IDS systems that work with large and diverse data sets can be challenging due to high dimensionality and the presence of irrelevant data. Feature selection techniques stand out as an important method to overcome these kinds of challenges and enhance system performance. By identifying and retaining the most relevant features in the dataset, these techniques enable models to function more quickly and accurately. Thus, utilizing different feature selection methods can greatly improve the effectiveness and performance of IDSs (Lyu et al., 2023; Liu & Yu, 2005).

This study aims to develop an effective and high-performance machine learning-based intrusion detection system by combining benefits of pre-processing, feature selection, class balancing, and an ensemble learning approach.

In the study, Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting (LightGBM) are used as machine learning algorithms. The most significant features within the dataset are identified by using Recursive Feature Elimination (RFE), Spearman's Correlation Analysis, and Information Gain methods (IG). The class imbalancing problem in the data set is also adressed with the Near Miss undersampling method. The performances of the established models are evaluated using accuracy, precision, recall, f1-score metrics.

The objectives of the study, problem statements, research questions and study outline are also presented in the subsections under this section.

1.1. Problem Statement

Cybersecurity, in this fast-growing digital world of today, is a prerequisite for the safety of networks and information systems. Indeed, it is continuously threatened by the growing complexity and diversity of current cyber-attacks. Due to this fact, most of the intrusion detection systems currently used suffer a lot from a number of serious issues that result from their limited flexibility, high rates of false-positives, and low accuracy of detection. High false-positive rates result in normal network traffic being mistakenly classified as threats and trigger unnecessary alerts, while low detection accuracy may cause real attacks to be overlooked, hence security breaches (Khan, 2022; Talukder, 2023).

Machine learning-based IDS systems emerge as a viable solution to these challenges. However, the existence of irrelevant and unnecessary features within large-scale datasets can considerably reduce model performance, prolong processing times, and elevate computational costs. Therefore, the selection of meaningful and relevant features from the dataset plays a critical role in enhancing model accuracy, optimizing processing efficiency, and lowering computational cost (Hota, 2014; Khammassi & Krichen, 2017; Zhou et al., 2020).

On the other hand, large datasets often include inaccurate, incomplete, noisy data, or data on different scales. This can negatively impact the model's performance and its ability to generalize effectively (Zhou et al., 2020). Pre-processing is, therefore, important for ML-based Intrusion Detection Systems because it ensures that the data is consistent, clean, and well scaled, enabling the system to accurately detect and effectively respond to potential threats.

Furthermore, one of the other important issues arising during the development of machine learning-based IDSs is the class imbalance problem. Indeed, class imbalance problems can prevent machine learning models from effectively identifying an attack type. When trained on imbalanced datasets, these models often prioritize accurately classifying the majority class (e.g., normal traffic), which can impairs their ability to detect attacks accurately. As a result, the model might show high overall accuracy, yet the rate of detecting attacks (true positive rate) could be significantly low. This scenario can lead to the IDS failing in its primary role of detecting attacks (Alfrhan et al., 2020; Barua,et al.,

2014). Moreover, systems that rely on a single classifier or model may fall short of achieving the desired performance in attack detection (Zhou et al., 2020).

To tackle the challenges mentioned above, there is a need to develop more flexible, advanced, and adaptable systems and approaches. Therefore, this study aims to develop an effective and high-performance ML based intrusion detection system by integrating the benefits of feature selection, class balancing, pre-processing, and an ensemble learning approach that combines the strengths of multiple models. Thus, this study seeks to provide a fresh perspective and valuable insights for both practical applications and academic research, contributing significantly to the existing literature in this field.

1.2. Objective

The main objective of this thesis is to develop an effective and high-performance ML based intrusion detection system by integrating the benefits of feature selection, class balancing, pre-processing, and an ensemble learning approach that combines the strengths of multiple models.

Another objective, which runs parallel to the main one, is to conduct a comparative analysis of feature selection methods and investigate their impact on the performance of machine learning models. This therefore targets finding the most effective approach toward enhancing the accuracy and efficiency of IDSs to enhance general performances of the IDS in network attack detection.

1.3. Research Questions

The Research Questions (RQ) which will be answered in this thesis are as follows:

RQ1: How do feature selection techniques effect the performance of intrusion detection models?

RQ2: Which technique produces the superior outcomes among RFE, Spearman's correlation analysis, and IG techniques?

RQ3: How does the proposed model that integrates the benefits of efficient preprocessing, class balancing, feature selection, and an ensemble learning approach perform compared to individual models in the network intrusion detection?

1.4. Structure of Thesis

The thesis consists of 5 chapters.

The first chapter emphasizes the importance of intrusion detection systems (IDS) in developing an effective defense mechanism against the increasingly complex and diverse network attacks that have emerged in parallel with advancing technology and digitalization. It also explains the effects of machine learning and feature selection methods on the effectiveness and performance of IDSs and presents the thesis's objectives, structure, problem statement, and research questions.

The second chapter explains the concept of information security and gives detailed information about types of cyber security attacks, intrusion detection systems, machine learning techniques, ensemble learning, and feature selection methods. This chapter then presents existing studies in the literature that are related to the thesis topic and formulates a research gap drawing from a comprehensive review of relevant literature.

The third chapter presents the research methodology, which includes an overview of data collection, data preprocessing, data balancing, feature selection, data splitting and scaling, model building, training and evaluation, the proposed system, and relevant performance measures.

The fourth chapter presents a synthesis of the study's generalizable findings along with the experimental results and discusses the results.

The final chapter summarizes the study's key conclusions and significance, answers the research questions, presents the limitations of the study, and provides recommendations for future research.

2. LITERATURE REVIEW

2.1. Information Security and Cyber Attacks

Information, one of the most important concepts of the developing and changing world, has been widely researched by thinkers since ancient times. Today, although information continues to be researched, the security of information emerges as another research topic and problem.

Information security refers to the process of safeguarding information against unauthorized access, utilization, modification, disclosure, destruction, or any form of damage. Confidentiality, Integrity and Availability are considered as three basic components of information security worldwide. These components are also known as the CIA triad. The CIA triad is a design model that guides the creation of security policies. Components such as risk approaches, information to be protected, and processes are first evaluated within the framework of this model and tried to be protected (ISO/IEC 27001, 2022; Harman et al., 2012; Kim & Solomon, 2018).

Figure 2. 3 Three Basic Components of Information Security

Confidentiality refers the protection of information from unauthorized access (Kim & Solomon, 2018; ISO/IEC 27001,2022).

Integrity ensures that the information is complete, accurate, consistent and correct (Kim & Solomon, 2018; ISO/IEC 27001, 2022).

Availability ensures that information can be accessible to authorized persons when needed (Kim & Solomon, 2018; ISO/IEC 27001, 2022).

In order to say that information is secure, all three of these elements should be provided. The proliferation of information systems and especially Internet technologies has brought with it an increased risk of exposure to information security breaches. For this reason, the security of Internet-connected devices against various threats, the integrity of the system and ensuring that they are constantly accessible have become very important issues (Ren et al., 2016).

Ensuring continuity depends on the measures taken against attacks being up-to-date. In order for the measures to be up-to-date, new attacks and their methods must be followed, learned, and added to existing systems.

Any action taken to prevent the confidentiality, integrity, reliability or availability of a resource constitutes the definition of an attack. In this context, a cyber attack can be defined as the interruption of the functions of computer-based systems, the weakening of their effectiveness or the unauthorized monitoring of the online network.

There are many types of cyber attacks that aim at applications, systems, and infrastructure. Some of the most common attacks are explained in the following section.

2.1.1. Types of Cyber Attacks

The main purpose of cyber attacks, which have increased in recent years, is generally to access the target system and access information or to cause permanent damage to the system. Sometimes these attacks, which manipulate the information accessed and make the system unreliable, are carried out with different methods. In attacks targeting system resources, called active attacks, the main purpose is to corrupt data. In another type of attack called passive attacks, the purpose is only to access and use information. Common cyber attacks are explained below.

2.1.1.1. Dos and DDos

As the Internet and networks have expanded, Denial of Service (DoS) attacks have emerged as the most prevalent type of attack in network security. When DoS attack occurs, the system's ability to respond to a request for service is compromised. An attacker-controlled host computer that has malware on it starts a DOS attack. This kind of cyber attack causes the disruption of the host's service, rendering the system or network resources unavailable for the intended user. While DoS attacks are launched from a single source, Distributed Denial of Service (DDoS) attacks are launched from multiple sources. DDoS attacks are much larger and more complex than DoS attacks, so it is more difficult to defend against them (Biju et al., 2019; Stallings, 2012).

2.1.1.2. Infiltration

An infiltration attack, also known as network infiltration, is a type of attack carried out by exploiting vulnerable software. For example, attackers can infiltrate through ports used by commonly used programs such as Adobe Acrobat Reader or Dropbox. Once infiltration is achieved, attackers can easily gain access to the computer through the compromised ports and then to the local network. Through these ports, they can execute various attacks from the computer. Additionally, attackers can use NMap to perform IP scans on the local network, identify other vulnerabilities within the network, and launch attacks on network devices or servers (Armstrong, 2007; Sharafaldin et al., 2018).

2.1.1.3. Botnet

In recent years, Botnet malware has been widely used for large-scale internet attacks. Bot is derived from the word “Robot”, which is defined as a machine that performs pre-planned tasks. Botnets are large groups of bots that are managed from a single center. Botnets direct bots in a certain order for certain purposes. The computers that are managed by botnets are called zombies or botnet members. Botnets are malicious software. Like worms and viruses, these software also spread by infecting vulnerable computers. The main factors that make these malicious software different from others are that they can update and manage themselves by communicating with the command and control (C&C) center (Stallings, 2012; Kara & Şişeci, 2011).

Like other types of malware today, botnets primarily spread through the following ways:

- Security vulnerabilities in IT products
- Weak or insecure policies
- Social engineering tactics

2.1.1.4. SQL Injection

SQL injection is one of the most popular attacks on web-based databases nowadays. In these attacks, the attacker actively strives to undermine online application security by leveraging SQL language features.

SQL injection poses a significant risk to any web application that processes user input to formulate and execute SQL queries against an database. Malicious actors can manipulate the data submitted into such web applications, injecting harmful SQL code into a query, and executing arbitrary SQL commands. This situation may cause the unauthorized extraction of sensitive customer data from e-commerce platforms or jeopardize the robust security measures designed to safeguard databases and file systems (Hasan et al., 2019; Kemalis & Tzouramanis, 2008).

2.1.1.5. Phishing Attacks

Phishing attacks are a form of cyber attacks where attackers try to trick people into getting confidential information like passwords, account details, or credit card numbers by masquerading as a reliable entity. These kinds of attacks are typically executed through e-mails, messages, or fake websites that are very similar to original ones. Attackers often exploit emotions such as fear, curiosity, urgency, and greed to coerce their victims into opening e-mail attachments or clicking on links. It is one of the most commonly faced and dangerous attack types lately (Basnet et al., 2008; Gupta et al., 2016).

2.1.1.6. Brute Force

Brute force attacks, which have an important place among cybersecurity threats, are attempts to obtain passwords and usernames through trial and error, to find the encryption key of a message or a secret

web page (Maryam et al., 2014). Different methods are used to capture sensitive data in each brute force attack.

Some of the ways to protect against brute force attacks are as follows:

- Using strong passwords to protect against identity theft, data loss, and unauthorized access to accounts
- Using CAPTCHA
- Using a multi-factor authentication system
- Increasing password complexity to make it harder to crack and increase the duration of the password
- Using features that prevent subsequent attempts after a certain number of incorrect entries

2.2. Intrusion Detection Systems (IDS)

With the rapid development of the Internet, networks and devices, information sharing has increased and storage of information in devices and cloud environments has become widespread. In parallel with this development, our information, networks and devices are faced with the threat of being exploited by malicious people at any time. With the increase in threats, ensuring information security has become a great necessity. Various intrusion detection systems (IDS) have been created to meet the need for information security. IDSs are classified in various ways in terms of detection approach and data source. They are classified as network-based, host-based IDS according to the data source, and signature and anomaly-based IDS according to the detection approach (Ajiya et al. 2021). The classification scheme of IDSs is given in Figure 2.2.

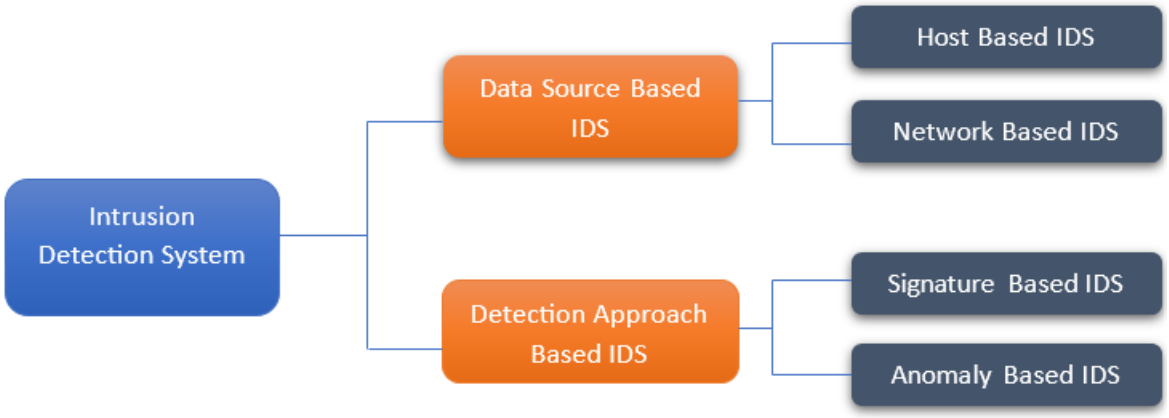

Figure 2.4 Types of Intrusion Detection System

2.2.1. Data Source Based IDS

2.2.1.1. Host Based IDS

Host Based Intrusion Detection System (HIDS) is a system that detect attacks by collecting data from the host computer and other computers, servers and other devices to which it is connected, analyzing potential threat elements and evaluating suspicious activities. HIDS monitors activities on system records in particular and identify anomalies there (Saxena et al., 2017).

One of the main advantages of HIDS is its capability to monitor the internal workings/operations of a host. Another advantage of HIDS is the ability to track encrypted traffic. However, HIDS has some disadvantages; for instance, analyzing intrusion attempts across several computers can be challenging, and attackers can disable HIDS once they have gained access to the system (Rajasekaran & Nirmala 2012).

2.2.1.2 Network Based IDS

Network-Based Intrusion Detection System (NIDS) consists of two different network devices. It has an Ethernet card (NIC) that works in different modes as well as a manageable network device. It analyzes subnets by monitoring all traffic on the network it is connected to (Rajasekaran and Nirmala

2012). It quickly detects known attacks by creating an alarm in detected abnormal situations. However, its success rate in detecting zero-day attacks is low.

2.2.2. Detection Approach Based IDS

2.2.2.1. Signature Based IDS

Signature Based IDS is based on searching for "known patterns (signatures)" of malicious activities. In the signature-based detection method, each attack is recorded by creating a wordlist with a uniquely defined signature. Each new attack detected is stored in this wordlist. Thus, a defense system is created on known and discovered attacks. All this system needs to do is search the list of recognized attack signatures, and once it finds a match, it reports it to the user or institution. It is quick because it only compares what it detects with a predefined rule. However, when a new attack is carried out, it will not be able to protect against these new attacks since it will not match these attacks with any pattern from its own database. Attacks can camouflage themselves by splitting messages. After a new attack is recorded, the data files, attack signatures need to be updated before the network becomes secure (Aleroud & Karabatis, 2017; Ajiya, 2021).

2.2.2.2 Anomaly Based IDS

Anomaly-based IDS monitors network traffic or system operations to detect anomalies that disrupt normal traffic patterns and classifies them based on their types. Unlike signature-based system, anomaly-based system focuses on detecting attacks through heuristic methods. Generally, it aims to monitor system traffic by labeling data as normal or attack-related based on previously trained datasets (Rajasekaran & Nirmala, 2012).

During the training or learning phase, normal and abnormal traffic profiles are created and taught to the model. In the testing phase, the efficiency of the learning process is evaluated, and the models's performance is assessed. Models that successfully pass these two phases can be integrated into systems. While many anomaly-based IDSs use artificial intelligence techniques, data mining and natural language processing techniques are also employed. One of the main advantages of these systems is their ability to detect new types of attacks (Lalduhsaka et al.2022).

2.3. Machine Learning Techniques

In this study, some primary aspects such as accuracy, learnability, scalability, and speed were considered when choosing classifier algorithms. After researching some prior studies that support the evaluation, five machine learning algorithms were taken into consideration in this study; they are: Random Forest, Logistic Regression, Decision Trees, Lightgbm, and XGBoost. Details about the algorithms used in this study are presented in the following section.

2.3.1 Random Forest

Random Forest (RF) is one of the most commonly utilized algorithm in the field of ML. This algorithm makes use of the predictive power of several decision trees. Every decision tree within RF is trained utilizing a randomly selected subset of the dataset. The result is a collection of decision trees. The trees become more diverse as a result of the randomness, which also helps to prevent overfitting. The final forecast is created by integrating the individual guesses from all the trees in the Random Forest, frequently through voting or averaging (Breiman,2001; Khan et al.,2021).

RF is a popular option for many machine learning problems since this ensemble technique increases the overall accuracy, performance and robustness of the model. Figure 2.3 shows the flow chart of the RF algorithm.

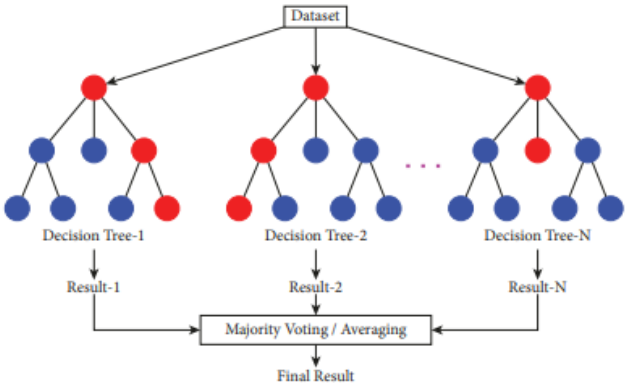

Figure 2.3 Random Forest Sample (Khan et al.,2021)

2.3.2. Logistic Regression

Logistic regression (LR) is a supervised ML algorithm commonly utilized for binary classification. It employs a logistic or sigmoid function to forecast the probability value or binary outcome (Equation 2.1). This function returns a probability value between 0 and 1 (Hosmer et al., 2013; Peng, 2002).

The ouput of logistic regression is established by a decision boundary and a threshold. In the case of binary classification, for instance, if the output ≥ 0.5, it is classified as class A; if not, it is classified as class B, as illustrated in the Equation (2.2) (Hosmer et al., 2013).

$$f(x) = 1 + \frac{1}{1+e^{-x}} \tag{2.1}$$

$$\begin{Bmatrix} A, & if\ f(x) \geq 0.5 \\ B, & otherwise \end{Bmatrix} \tag{2.2}$$

In a two-dimensional space, the resulting curve can be represented as an S-shape, as illustrated in Figure 2.4. It is a common choice in many domains since logistic regression is straightforward, comprehensible, and capable of handling both continuous and categorical variables.

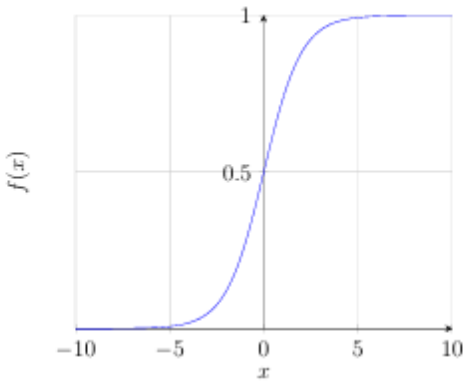

Figure 2.4 LR Sigmoid Function

2.3.3. Decision Tree

The decision tree (DT) is a another commonly utilized ML algorithm for the purposes of classification and estimation. It is frequently preferred due to its simplicity, efficiency, interpretability and ability to handle both categorical and numerical data (Chien & Chen,2008; Mienye & Jere, 2024).

Decision trees are made up of three components; decision nodes, branches and leaves (Han & Kamber., 2000). In decision trees, the process begins at the root node and proceeds by traversing through successive nodes from top to bottom until reaching the leaf.

The process of classifying data with the decision tree method involves two stages. The initial stage, referred to as the learning stage, involves the analysis of previously available, known training data by the classification algorithm to develop a model. This learned/developed model is expressed as a decision tree or classification rules. The subsequent stage involves classification. In this stage, established classification rules or decision tree's accuracy is assessed on the test data. If result meets an acceptable level, these rules are then applied to classify new data (Chien & Chen, 2008; Han & Kamber, 2000).

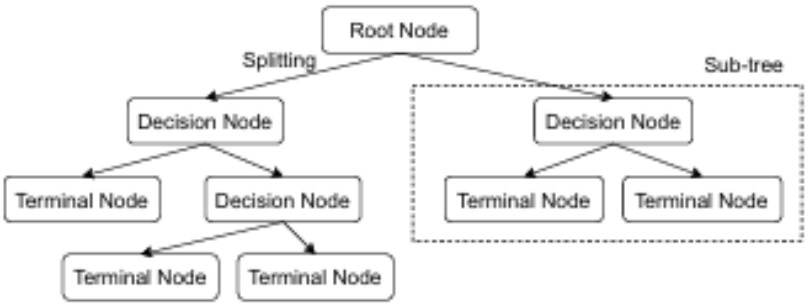

Figure 2.5 Decision Tree Sample (Mishra, et al., 2019)

2.3.4. LightGBM

LightGBM is robust, efficient algorithm which is improved version of gradient boosting algorithm. It has some advantages over other boosting algorithms in terms of computational speed, memory

consumption and prediction rate (Gong & Liu, 2022; Aksoy & Genc, 2023). According to the results obtained from experiments with different data sets, LightGBM has 20 times faster training time than GBDT algorithms (Ke et al., 2017).

LightGBM is capable of working effectively with large datasets. Unlike other boosting algorithms, LightGBM employs the leaf oriented approach as illustrated in Figure 2.6 in the training phase of decision trees, and since the division process is continued from the leaves that reduce the loss, it reduces the error and increases the learning speed (Aksoy & Genc, 2023; Rufo et al., 2021).

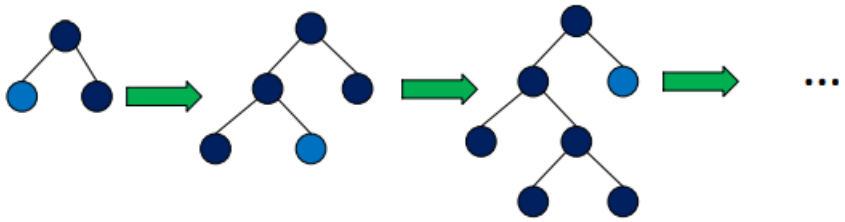

Figure 2.6 Leaf-wise tree growth in LightGBM (Ke et al., 2017)

2.3.5. XGBoost

XGBoost is an effective and scalable version of the decision tree-based gradient boosting algorithm. The most important factor behind the success of the XGBoot algorithm is that it is scalable for all scenarios in which it will be used (Chen & Guestrin, 2016).

XGBoost algorithm is based on the logic of bringing together relatively weak decision trees, also called weak learners, and creating a much stronger tree through ensemble learning. As in other boosting algorithms, it strengthens the weak algorithm by training it sequentially and iteratively. Each tree created in the algorithm is added to reduce the loss function, and since each added tree is fed with the model of the previous tree, it reduces this loss rate. With this structure, various optimizations have been made to improve performance and the XGBoost algorithm has emerged (Chen & Guestrin, 2016; Li et al., 2019; Dhaliwal et al., 2018).

XGBoost algorithm is an optimized type of traditional gradient boosting algorithms. One of the biggest advantages of the XGBoost algorithm over traditional gradient boosting algorithms is that it has a smoother structure to prevent overfitting. In addition, GPU support, less resource usage, high performance values, fast training process and the ability to run on large data sets are among its other important features (Chen & Guestrin, 2016; Dhaliwal et al., 2018).

2.3.6. Ensemble Learning Approach

Ensemble learning is a ML method that employs various algorithms simultaneously to achieve a more successful prediction result (Polikar, 2006; Kuncheva & Whitaker, 2003). An ensemble consists of a set of learners (decision tree, neural network, etc.) called base learners or weak learners and is created in two steps. These are selecting the base learners and then combining the prediction results of these learners (Zhou, 2012). To receive the best prediction results, the base learners in the ensemble should be selected appropriately for the problem and should be diverse. The base learners that make different errors at different data points are different from each other, i.e. diverse (Fawagreh et al., 2014). Which base learners can be used together and transformed into strong learners with high predictive power is an important topic studied in the area of ML and has resulted in the emergence of ensemble learning methods such as Bagging, Boosting, Stacking, Voiting (Rokach, 2010; Zhou, 2012).

The benefit of ensemble learning compared to single base learner is the ability to integrate prediction outcomes from multiple base learners to enhance accuracy, efficiency, generalizability and robustness (Jaw & Wang, 2021; Gao et al., 2019).

2.3.6.1 Bagging

In this method, base learners are trained on different subsets randomly chosen from the training set. Since each selected subset is replaced, some samples may occur more than once in the training set. The purpose of diversifying the training sets is to increase the overall prediction accuracy. Finally, all the predictions from the learners are combined by averaging for regression problems and by weighted voting for classification problems (Breiman, 1996; Tama ve Rhee, 2017; Sutton, 2005). Figure 2.7 illustrates the process steps of the Bagging method.

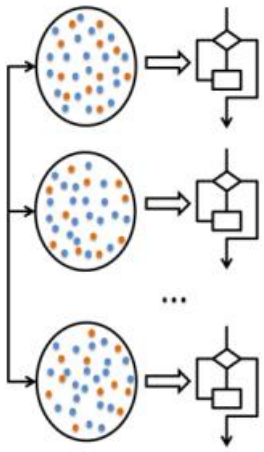

Figure 2.7 Process of Bagging method (Torabi et al.,2021)

2.3.6.2 Boosting

In this method, the base learner is trained using randomly selected data from the dataset allocated for training. The resulting model is then tested, and incorrectly classified examples are identified. These misclassified examples are prioritized in the selection of training data for the next learner. This selection is updated with each training iteration (Sidharth & Kavitha, 2021). By focusing on the mistakes made, the accuracy of predictions is improved. The goal of boosting is to integrate multiple weak learners in order to form a robust learner (Zhou, 2012; Kearns, 1988). In the bagging method, the probability of selecting each example in the training dataset remains the same in every iteration, whereas in boosting, the selection probabilities of the data samples are updated in each iteration. Figure 2.8 illustrates the process steps of the Boosting method.

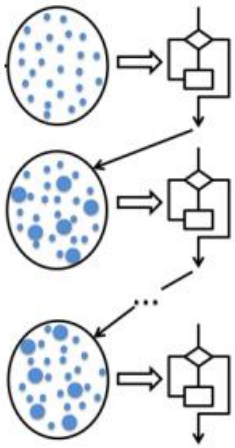

Figure 2.8 Process of Boosting method (Torabi et al.,2021)

2.3.6.3. Stacking

Unlike bagging, and boosting ensemble learning methods, stacking uses a distinct model called a meta-learner to integrate the outcomes of base models (Jain, et al., 2023). In order to achieve high accuracy, predictions from various classifiers are provided as input to the meta-classifier. In this approach, after predictions are obtained from different types of classifiers created using the training dataset, results of predictions are integrated in the meta learner or classifier to create a collective model and produce the final results (Wolpert, 1992). Figure 2.9 illustrates the process steps of the Stacking method.

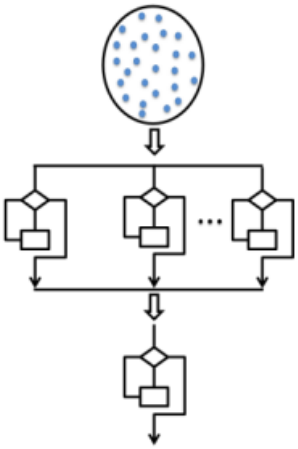

Figure 2.9 Process of Stacking method (Torabi et al.,2021)

2.3.6.4. Voiting

This method combines different machine learning classification models and aggregates the prediction values obtained from each model used. It uses these prediction values to predict new observation's class through voiting. In this process, the vioiting can be conducted in two forms: hard and soft. In hard voting, the class that receives the highest number of votes is determined by looking at the classification results of different models. Here, each model casts a vote for a one class, and the class receiving the highest number of votes is deemed the last prediction. In soft voiting, a more sensitive selection is made by considering the probability distributions estimated by the models. Every individual model provides a probability estimation for every class available within the dataset. Ultimately, the class that receives the highest average probability from these calculations is chosen as the definitive prediction or final estimate (Ampomah, et al., 2020; Raihan Al Masud, 2019).

Figure 2.10 illustrates an example of hard voting where the majority vote is used.

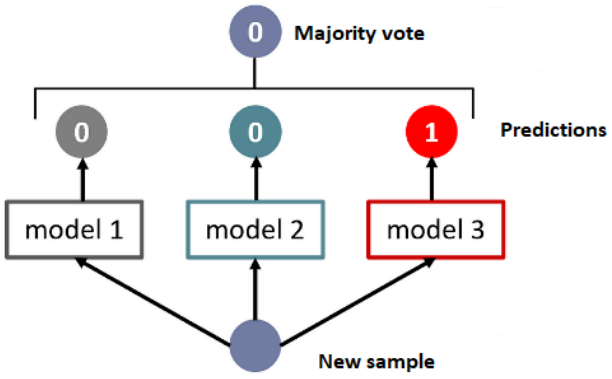

Figure 2.10 Example of hard voting

2.4. Feature Selection

In ML, feature selection (FS) plays a critical role in enhancing model performance during preparing the dataset. Through feature selection, it becomes possible to identify features that do not contribute to the model's success, unnecessarily expand the dataset, or negatively impact overall model efficacy. By removing features that have a minimal influence, FS facilitates more effective classification. Additionally, feature selection reduces the time and computational resources required for both model training and prediction. Therefore, employing an appropriate feature selection method to identify the

most significant and relevant features in the dataset can lead to enhanced model performance and accelerate the training and prediction processes (Mishra et al.,2018; Jaw & Wang, 2012).

There are several methods in the literature regarding feature selection. This study specifically utilizes Correlation Based Feature Selection, Recursive Feature Elimination Feature Selection, and Information Gain Feature Selection techniques. A comprehensive overview of these techniques will be presented in the following section.

2.4.1. Correlation Based Feature Selection

Correlation-based feature selection (CFS) uses correlation analysis. Correlation is defined as the statistical relationship between two variables. In machine learning, correlation is used to check how much two or more features are related to each other. If any of the two features is highly correlated or both carry the same information, then one of them is redundant and is selected from among them.

CFS employs a search algorithm as well as a function that measures the information values of sub-groups of features. It also takes into account the intercorrelation values between them when estimating the class label of each feature. A good subset or group of features comprises those that exhibit a strong correlation with the class (target variable) while maintaining a low correlation among themselves (Hall,1999; Zhou et al., 2019).

Equation 2.3 illustrates function of feature subgroup that is used in CFS.

$$M_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1) + \overline{r_{ff}}}} \tag{2.3}$$

In Equation 2.3, , M_s is result of evaluation for a feature subgroup s that comprises k features, $\overline{r_{cf}}$ is the average correlation between the external variable or category class and the features, and $\overline{r_{ff}}$ is the average intercorrelation of the features with each other (Zhou et al., 2019, p.3).

2.4.2. Recursive Feature Elimination

In this technique, the dataset is evaluated in subsets using a chosen supervised learning method. The least important feature of each subset is identified and removed. This recursive process continues until the desired number of features remains. The fundamental logic behind this technique is to identify the top n most meaningful features by dividing all the features in the dataset into subsets. Within these subsets, the features are scored based on their importance. The least important features are then eliminated from the subsets. This recursive process continues until only the top n most important features remain. A supervised learning algorithmis is employed to assess the significance of each feature (Scikit Learn, 2022). RFE is particularly advantageous for datasets with a large number of features, as it enables the identification and emphasis of the most relevant information pertinent to the task at hand.

2.4.3. Information Gain

Information gain (IG) is a feature selection technique that relies on the concept of entropy. Entropy measures the uncertainty in the system and takes a value between 0 and 1. A high entropy value means that the system in question contains more information. The aim of IG method is to gain information about feature Y by observing feature X and to measure the decrease in the entropy value of feature Y. In this method, the entropy values of the class variables are calculated (Equation 2.4). Then, the entropy values are calculated for each feature in the data set (Equation 2.5). Finally, IG is determined by calculating the difference between the obtained entropy values (Equation 2.6). The higher the result, the more successful the relevant feature is in representing the dataset. Features with low IG values and are insufficient in the representation of the data set are eliminated (Kaynar et al., 2018; Kurniabudi et al., 2020).

IG is a symmetric evaluation criterion. A disadvantage of this approach is that it produces biased outcomes that favor features with high cardinality, despite not providing additional information (Budak, 2018). This causes over-fitting.

$$H(Y) = -\sum_{y \epsilon Y} p(y)\ log_2\big(p(y)\big) \tag{2.4}$$

$$H\ (Y\backslash X) = -\sum_{x \epsilon X} p(x) \sum_{y \epsilon Y} p(y\backslash x)\ log_2\big(p(y\backslash x)\big) \tag{2.5}$$

$$Information\ Gain = \mathrm{H(Y)} - \mathrm{H(Y\backslash X)} \tag{2.6}$$

2.5. Related Work

In the literature, it is seen that numerous studies have been conducted on intrusion detection systems (IDS) developed for detecting cyber attacks. Many of these studies propose various methods and techniques to enhance the performance of IDS. Some of these studies are presented below.

Nimbalkar and Kshirsagar (2021) proposed a feature selection method for intrusion detection systems (IDS) that incorporates Information Gain (IG) and Gain Ratio (GR). In their research, they identified 50% of the most significant features necessary for detecting DoS and DDoS attacks. The researchers evaluated their models using the KDDCUP'99 and BOT-IOT datasets. For the BOT-IOT dataset, they selected 16 features, while for the KDD CUP'99 dataset, they chose 19 features. They then utilized the JRip classifier to train their model in order to achieve optimal results. Impressive accuracy rates of 99.99% were achieved in their analysis for both datasets.

Kasongo and Sun (2020) implemented a filter-based feature reduction method in their study, benefiting the XGBoost algorithm for IDS. After employing this method to reduce the feature set, they applied several algorithms, including ANN, SVM, KNN, DT, and LR. Their models were trained and evaluated on the UNSWNB15 dataset, focusing on both binary and multiclass classification scenarios. In the context of binary classification, test accuracy for models like DTs increased from 88.13% to 90.85% as a result of integrating XGBoost for feature selection.

Khammassi and Krichen (2017) employed a wrapper method that integrates a genetic algorithm for feature selection and the LG algorithm as the learning method to identify the optimal set of features for network intrusion detection systems. Their implementation of this wrapper method proved to be quite effective. The results demonstrated that it was capable of accurately detecting intrusions using only 20 features from the UNSW-NB15 dataset and 18 features from the KDD99 dataset.

Sharafaldin et al. (2018) employed a feature reduction technique, namely Mean Decrease Impurity (MDI), in their studies. With the help of this approach, they selected the best features for each of the 15 traffic types and reduced data volume when training and testing. After feature selection, they applied seven machine learning algorithms and tested the performance of these algorithms on CICIDS 2017. Result of their studies, they achieved accuracy values ranging from 77% to 98%, recall values between 4% and 98%, and F1 scores varying from 4% to 94%.

Hota and Shrivas (2014) proposed a model in which different methods of feature selection are used to remove the unnecessary features from the dataset. The findings of the study indicated that the C4.5 algorithm, when utilized in conjunction with Information Gain, exhibits superior performance, attaining a remarkable accuracy rate of 99.68% on the test set after the selection of merely 17 features.

Awad and Alabdallah (2019) introduced a weighted extreme learning machine method (ELM) to deal with the the class imbalance problem in IDS. Their study showed that the weighted ELM method can efficiently deal with the unbalanced classification and improve the prediction accuracy and overall performance. An important contribution of their study is the creation of an innovative strategy for dealing with imbalanced data, a common challenge in intrusion detection.

Yueai and Junjie (2009) developed a two-step approach that incorporates a load balancing model for the implementation of an IDS, consisting of both online and offline phases. In the online phase, the system gathered packets from the network to detect potential intrusions. Meanwhile, the offline phase utilized a training dataset to create an offline model. They employed the SMOTE technique for oversampling and conducted classifications using the AdaBoost and Random Forest algorithms. However, their experimental results revealed that the combination of SMOTE and AdaBoost did not yield satisfactory effectiveness.

Parsai et al. (2016) proposed a hybrid method that integrates SMOTE and cluster center and nearest neighbor (CANN). In their study, they used the leave one out (LOO) strategy to identify significant features in the dataset. Their findings revealed that this newly proposed method significantly boosted the accuracy of detecting R2L (Remote to Local) and U2R (User to Root) attack types, achieving enhancements of 50% and 94%, respectively, in comparison to standard benchmarks.

In their research, Gao et al. (2019) presented a novel model for adaptive ensemble learning, which focuses on adjusting the learning data ratio and utilizes a multi-tree algorithm with multiple decision trees. In their research, they chose various base classifiers such as RF, DT, DNN, and KNN, along with developing an adaptive voting algorithm aimed at enhancing detection efficiency. Evaluating their findings on the NSL-KDD dataset, they found that the multi-tree algorithm achieved an accuracy of 84.2%, while the final adaptive voting ensemble improved this accuracy to 85.2%.

To improve the effectiveness of learners, Paulauskas and Auskalnis (2017) proposed an ensemble approach that incorporated algorithms like C5.0, Naive Bayes, J48, and PART. This strategy was built on the idea of combining multiple weaker learners to create a more robust model. Their research revealed that this ensemble model significantly enhanced performance and accuracy in the context of an Intrusion Detection System (IDS).

Govindarajan (2014) suggested a hybrid classification ensemble model that combines the strengths of RBF and SVM methods. They tested and evaluated their models' performances on the NSL-KDD dataset, a widely used dataset in intrusion detection. Their results showed that their ensemble model performed better than individual models, with an impressive accuracy rate of 98.46%.

As a result of the literature review, it was seen that many studies were conducted on intrusion detection systems (IDS) developed for the detection of cyber-attacks and various techniques and methods were applied. However, it was noticed that there were not enough studies focusing on machine learning algorithms, efficient preprocessing, class balancing/undersampling, feature selection and ensemble learning methods in the same study and examining in detail the effect of their combined use on intrusion detection performance.

Therefore, this study differs from other studies in that it focuses on the development of more effective and high-performance IDS systems by presenting an approach that combines machine learning, class balancing/undersampling, feature selection and ensemble methods.

Thus, this study aims to make a significant contribution to the literature by presenting a robust and well-structured approach that enables more effective handling of advanced and complex network intrusions, and allowing for their more accurate and reliable detection, and by developing different perspectives and insights. Additionally, this study is expected to provide a solid foundation for the future development of more secure and intelligent intrusion detection systems.

3. METHODOLOGY

3.1. Data Collection Procedure and Dataset Description

This study utilized the CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity, which encompasses various scenarios of network attacks. Due to its large scale and detailed structure, CICIDS2017 is a key resource for designing and evaluating new models and algorithms aimed at mitigating network intrusions. The dataset includes eight distinct files, covering five days of both normal and attack-related network traffic data, provided by the Canadian Institute for Cybersecurity. In total, the dataset contains 2,830,743 records, each described by 79 different features (Sharafaldin et al., 2018).

Table 3.1 presents the distribution of attack categories in the dataset.

Table 3.1 Distribution of attack categories in the CICIDS2017 dataset

Attack categories	Count	Percentage (%)
Bening	2273097	80.3004
Dos Hulk	231073	8.1630
PortScan	158930	5.6144
DDoS	128027	4.5227
Dos GoldenEye	10293	0.3636
FTP-Patator	7938	0.2804
SSH-Patator	5897	0.2083
DOS slowloris	5796	0.2048
DOS Slowhttptest	5499	0.1943
Bot	1966	0.0695
Web Attack Brute Force	1507	0.0532
Web Attack XSS	652	0.0230
Infiltration	36	0.0013
Web Attack Sql Injection	21	0.0007
Heartbleed	11	0.0004
Total	2830743	100

In the dataset, while 80.3% (2,273,097) of the total 2,830,743 instances are benign (normal), the remaining 19.7% are attack. Among the attack records, Dos Hulk attack (231,073 records,

8.1630%) has the highest frequency, while Heartbleed attack (11 records, 0.004%) has the lowest frequency. Visualizations of the distribution of attack categories are presented in Figure 3.1.

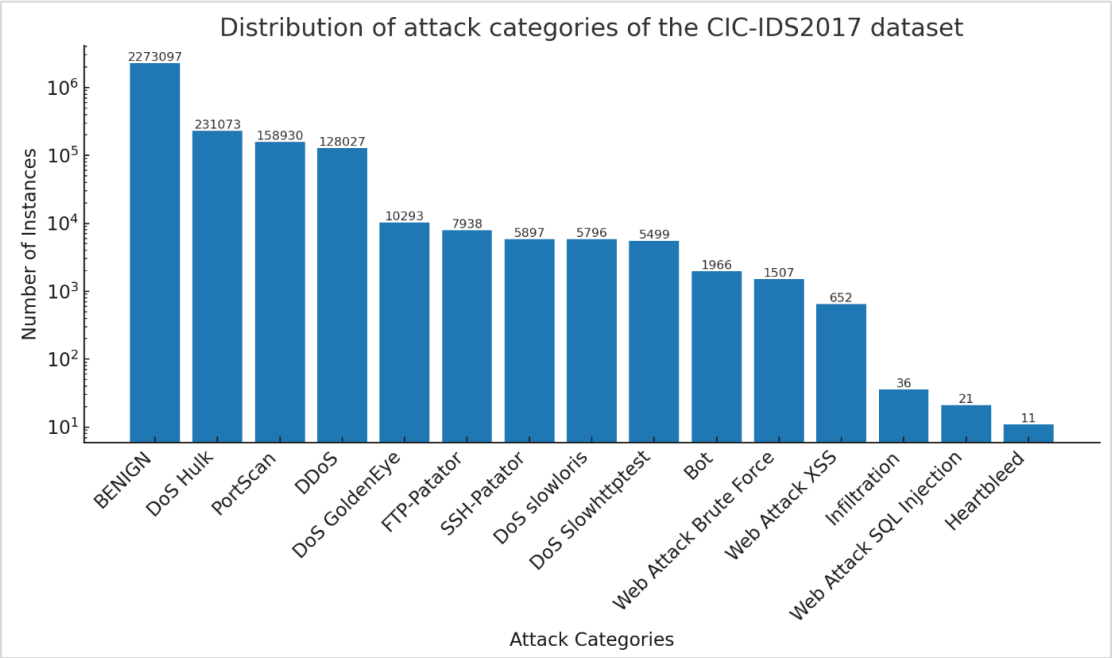

Figure 3.1 Distrubution of attack categories of the CIC-IDS2017 dataset

3.2. Data Preprocessing

Data preprocessing constitutes a essential and time-intensive phase in the realm of data mining. Typically, real-world data is derived from diverse sources and may exhibit characteristics such as noise, redundancy, incompleteness, and inconsistency (Cheng et al., 2018). Hence, it is imperative to transform unprocessed, orginal data into a format that is conducive to analysis and knowledge extraction.

Properly preparing the data significantly influences the performance of classification models; by implementing appropriate techniques, technical challenges associated with data preparation can be addressed, thereby enhancing performance levels. This section outlines the specific steps involved in data preparation, including data integration, data cleaning, data encoding. These steps are explained in detail in the following section.

3.2.1. Data Integration

The CICIDS2017 dataset comprises 2,830,743 instances distributed across eight files, with each instance characterized by 79 features. This dataset includes 14 distinct types of attacks, with attack traffic making up approximately 19.7% of the total incidents ((Sharafaldin et al., 2018). In our study, initially these eight files were merged into one consolidated file and then data cleaning steps were performed.

3.2.2. Data Cleaning

In the real world, the diversity of the platforms results in the raw data containing irregular, inconsistent, and duplicate instances, which could adversely affect the accuracy of classification In order to overcome this problem, it is crucial to eliminate these instances from the dataset at the beginning of our study. High-quality and reliable data are fundamental for producing accurate analyses, which in turn support well-founded and informed decision-making. Data cleaning is an important aspect of data preprocessing that enhances a dataset's usefulness. It guarantees that the data is devoid of inconsistencies and mistakes that might lead to technical problems with the model.

In our study, during the data cleaning process, the steps given below were applied to ensure the quality as well as reliability of the data set:

- Infinite values were replaced with the maximum value within the respective column
- Missing values were replaced with the mean of the respective column
- Features consisting entirely of zero values were excluded from the dataset
- Repeated features were removed from the dataset

3.2.3. Data Encoding

The process of data encoding is essential for transforming categorical, non-numerical variables into numerical variables that can be utilized by ML algorithms. Since our dataset contains some categorical variables, these variables were transformed into numerical variables by using the LabelEncoder function in Python. “Benign” categorical values were transformed to “0“ numerical values, and “Attack” categorical values were transformed to “1” numerical values since we are dealing with a binary classification issue.

The model would struggle to perform effectively if the labels were not encoded, as it would find it difficult to interpret them. Consequently, encoding the data allows the model to comprehend the labels as numerical values, enhancing its ability to process and learn from the data.

Table 3.2 shows the labels and their corresponding values.

Table 3.2 Labels and their corresponding values

Label	Value
Benign	0
Attack	1

3.3. Data Balancing

Class imbalance denotes a circumstance within a classification issue wherein the quantity of instances belonging to one class considerably exceeds that of instances in other classes.

Imbalance between classes is known to cause underfitting or overfitting problems. Classification techniques generally work with the assumption that data sets are balanced. However, in reality, many data sets can be quite imbalanced. Classifier algorithms learn the majority class features better, but they may be inadequate in learning the minority class features. Data balancing techniques make imbalanced data sets more balanced, which has a positive effect on the learning rate and increases model performance (Domingues et al., 2018; Jadhav et al., 2022).

It can be seen that in our dataset the number of attack records is considerably lower than that of normal records (see Figure 1). This observation is reasonable, as attacks do not typically happen as often as normal records. Nevertheless, the proportion of attacks to normal records presents a significant challenge that can substantially impact model performance.

Figure 3.2 shows the binary frequency distribution of the dataset before undersampling after preprocessing.

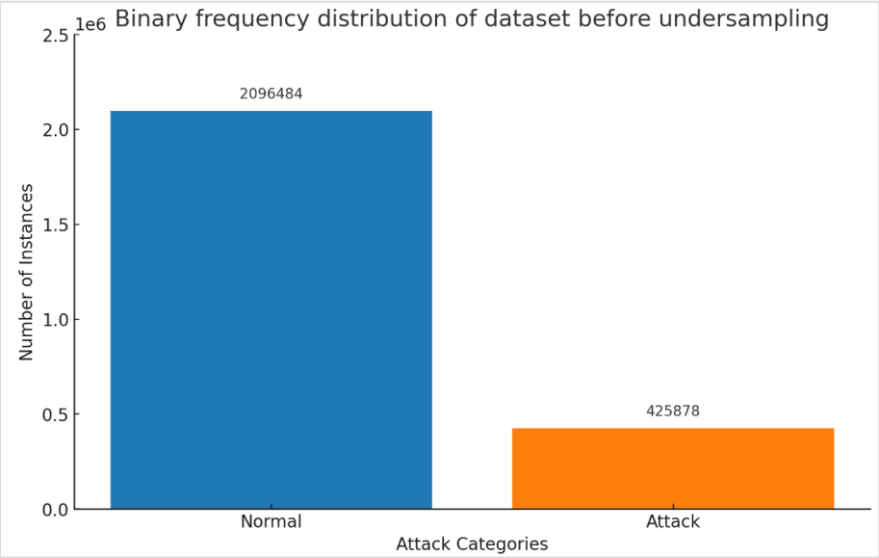

Figure 3.2 Binary frequecy distribution of dataset before undersampling

The IDS model might focus on identifying more common traffic patterns instead of rare attacks, resulting in a high overall accuracy but a low detection rate for minority attacks. This situation is known as the accuracy paradox, emphasizing that the accuracy metric may not truly represent the model's effectiveness and performance (Elmasry et al.,2019).

In this study, in order to develop a balanced dataset comprising normal and attack records from the original datasets, Near Miss undersampling technique were utilized.

Near Miss undersampling method strategically eliminates most instances from the majority class by considering their proximity to instances in the minority class. The objective is to develop a more balanced dataset while retaining the most valuable samples from the majority class (Mani and Zhang, 2003).

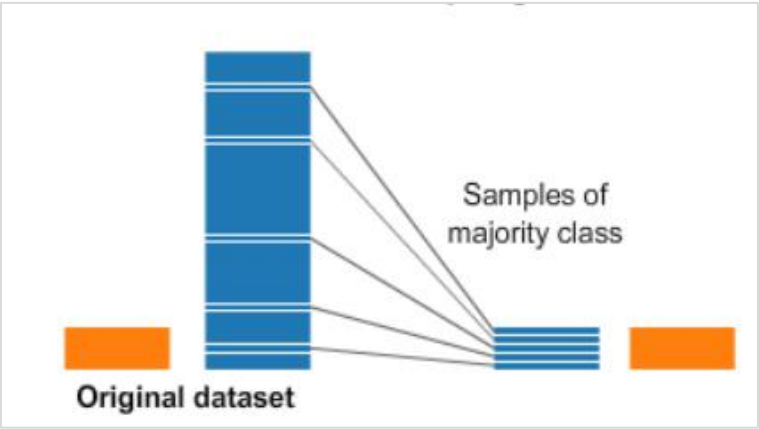

Figure 3.3 Visualisation of Undersamplig Technique

After applying Near Miss undersampling technique, our dataset achieved a balanced distribution, and its size was reduced from 2,522,362 to 851,756. As a result, out of a total of 851,756 records, 425,878 belong to the attack class, and 425,878 belong to the normal class. The balanced dataset distribution is shown in Figure 3.4.

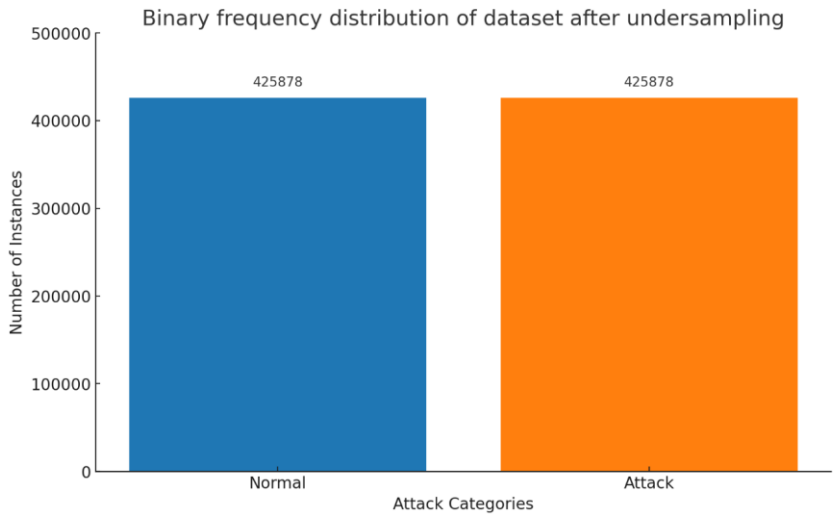

Figure 3.4 Binary frequecy distribution of dataset after undersampling

Attack and normal class numbers and distribution rates before and after Near Miss undersampling are presented in Table 3.3. below.

Table 3.3. Attack and normal class numbers and distribution rates before and after undersampling

Class	Before Undersampling		After Undersampling	
	Number of Instance	Percentage	Number of Instance	Percentage
Normal	2,096,484	83.12	425,878	50
Attack	425,878	16.88	425,878	50
Total	2,522,362	100	851,756	100

3.4. Feature selection

The features within a dataset play a critical role in influencing classification performance (Zhou et al., 2019). Having too few features may result in poor class separation, while too many can introduce challenges like longer training times and reduced accuracy due to noisy or irrelevant data. Therefore, it is essential to identify an optimal subset of features that adequately represents the original dataset, reducing training time, enhancing data quality, and improving model performance.

In this study, key features to be used for the training and testing phases of models were selected through Recursive Feature Elimination (RFE), Information Gain (IG), and Spearman's Correlation Analysis.

3.4.1. Recursive Feature Elimination

The RFE method begins by constructing a model using the complete set of features and assigns an importance score to each one. It then eliminates the least important features from the dataset and builds a new model with the updated set of features, recalculating their importance. This process is repeated iteratively until the desired number of features, as specified in the RFE parameters, is achieved.

The number of features to be selected in RFE method is a setting parameter. In our study, optimal number of features that gave the best success rates was determined by trial and error approach. In this regard, accuracy and attack detection performance were assessed for feature sets of 30, 40, 50, and 60. As a result of the evaluation, it was seen that 40 features that were selected using RFE increased the accuracy and attack detection success.

The performance results of the models developed using this dataset of 40 features are comprehensively discussed in the Results and Discussion chapter of the present study.

Table 3.4 40 Features selected as a result of the RFE Method.

Selected Features			
1	Destination Port	21	Bwd IAT Std
2	Flow Duration	22	Bwd IAT Min
3	Total Fwd Packets	23	Fwd Header Length
4	Total Backward Packets	24	Fwd Packets/s
5	Total Length of Fwd Pack	25	Bwd Packets/s
6	Total Length of Bwd Pack	26	Min Packet Length
7	Fwd Packet Length Std	27	Max Packet Length
8	Bwd Packet Length Min	28	Packet Length Mean
9	Bwd Packet Length Mean	29	Packet Length Std
10	Flow Bytes/s	30	FIN Flag Count
11	Flow Packets/s	31	PSH Flag Count
12	Flow IAT Mean	32	URG Flag Count
13	Flow IAT Std	33	Down/Up Ratio
14	Flow IAT Max	34	Average Packet Size
15	Flow IAT Min	35	Init_Win_bytes_forward
16	Fwd IAT Total	36	Init_Win_bytes_backward
17	Fwd IAT Mean	37	Act_data_pkt_fwd
18	Fwd IAT Std	38	Min_seg_size_forward
19	Fwd IAT Max	39	Active Mean
20	Fwd IAT Min	40	Idle Min

3.4.2. Spearman's Correlation Analysis

Another method used in this study to determine significant, key features is Spearman's correlation analysis. This method utilizes Spearman's correlation coefficient. This coefficient is used to determine both the strength and the direction of the relationship that exists between two distinct variables (Dubey, et al., 2021).

In this study, significant features were determined by calculating the Spearman's correlation between each feature and the dependent or target variable. The features whose absolute Spearman correlation with the dependent or target variable is greater than the specified threshold value were selected.

Here the threshold value is a setting parameter. Within the scope of this study, threshold values of 0.3, 0.5, 0.7 were tested, and among these, it was seen that the 0.3 threshold value gave the best results in terms of accuracy and attack detection success. After Spearman's correlation analysis, which took

into account the threshold value of 0.3, 43 features were selected from a total of 78 features. These 43 features selected as a result of Spearman's correlation analysis are shown in Table 3.5.

The performance results of the models developed using this dataset of 43 features are comprehensively discussed in the Results and Discussion chapter of the present study.

Table 3.5 43 Features selected as a result of the Spearman's correlation

Selected Features					
1	Flow Duration	15	Total Fwd Packets	29	Total Backward Packets
2	Total Length of Bwd Packe	16	Fwd Packet Length Min	30	Fwd IAT Mean
3	Fwd Packet Length Std	17	Bwd Packet Length Max	31	Bwd IAT Mean
4	Bwd Packet Length Min	18	Bwd Packet Length Mean	32	Flow IAT Mean
5	Bwd Packet Length Std	19	Flow Packets/s	33	Fwd IAT Total
6	Flow IAT Std	20	Flow IAT Max	34	Bwd IAT Total
7	Fwd IAT Std	21	Fwd IAT Max	35	Fwd Header Length
8	Bwd IAT Std	22	Bwd IAT Max	36	Min Packet Length
9	Bwd Header Length	23	Fwd Packets/s	37	Packet Length Variance
10	Max Packet Length	24	Packet Length Std	38	Subflow Fwd Packets
11	PSH Flag Count	25	Avg Bwd Segment Size	39	Init_Win_bytes_forward
12	Subflow Bwd Packets	26	Subflow Bwd Bytes	40	Active Mean
13	Init_Win_bytes_backward	27	act_data_pkt_fwd	41	Idle Mean
14	Active Max	28	Active Min	42	Idle Max
				43	Idle Min

3.4.3. Information Gain

In this study, another feature selection method which was performed to select relevant and important features is Information Gain method (IG). This method measures the decrease in entropy, which represents the level of uncertainty, when the dataset is divided according to a specific features. Features that result in higher Information Gain are considered more important as they provide more information about the target variable. To implement this method, each feature in the dataset is evaluated for its Information Gain with respect to the target or dependent variable. Features that have higher Information Gain values are retained, as they are deemed to have a greater impact on the classification process. Conversely, features with lower Information Gain values are discarded to reduce dimensionality and enhance efficiency of the model (Kaynar et al., 2018; Kurniabudi et al., 2020).

In this study, the number of features was considered as the setting parameter in order to select the most important and valid features using IG feature selection method. The number of features of 30, 40, 50, and 60 were tested, respectively, and the results were evaluated. As a result of the evaluations, it was seen that 60 features that were selected using IG method increased the accuracy and attack detection success. Table 3.6 displays the 60 features selected as a result of the IG method.

The performance results of the models developed using this dataset of 60 features are comprehensively discussed in the Results and Discussion chapter of the present study.

Table 3.6 60 Features selected as a result of Information Gain

1	Destination Port	21	Fwd IAT Min	41	PSH Flag Count
2	Flow Duration	22	Bwd IAT Total	42	URG Flag Count
3	Total Fwd Packets	23	Bwd IAT Mean	43	Down/Up Ratio
4	Total Backward Packets	24	Bwd IAT Std	44	Average Packet Size
5	Total Length of Bwd Packets	25	Fwd IAT Max	45	Avg Fwd Segment Size
6	Fwd Packet Length Max	26	Fwd IAT Min	46	Avg Bwd Segment Size
7	Total Length of Fwd Packets	27	Bwd IAT Total	47	Subflow Fwd Packets
8	Fwd Packet Length Min	28	Bwd IAT Mean	48	Subflow Fwd Bytes
9	Fwd Packet Length Mean	29	Bwd IAT Std	49	Subflow Bwd Packets
10	Bwd Packet Length Std	30	Bwd IAT Max	50	Subflow Bwd Bytes
11	Flow Bytes/s	31	Bwd IAT Min	51	Init_Win_bytes_forward
12	Flow Packets/s	32	Fwd Header Length	52	Init_Win_bytes_backward
13	Flow IAT Mean	33	Fwd Packets/s	53	act_data_pkt_fwd
14	Flow IAT Std	34	Bwd Packets/s	54	min_seg_size_forward
15	Flow IAT Max	35	Min Packet Length	55	Active Mean
16	Flow IAT Min	36	Max Packet Length	56	Active Max
17	Fwd IAT Total	37	Packet Length Mean	57	Active Min
18	Fwd IAT Mean	38	Packet Length Std	58	Idle Mean
19	Fwd IAT Std	39	Packet Length Variance	59	Idle Max
20	Fwd IAT Max	40	FIN Flag Count	60	Idle Min

3.5. Data Splitting

In this study, the dataset was split into two distinct subsets to conducting a comprehensive evaluation of the model. Specifically, 80% of the data was reserved for the training set, utilized for training the machine learning models. The remaining 20% of the data was allocated as the testing set, enabling an

impartial assessment of the model's efficacy and performance on the data that has not been unseen or unexamined before.

3.6. Data Scaling/Normalization

In datasets, there can be very large differences between the values of some data points, and this situation can cause the data with larger values to become disproportionately dominant, making accurate comparisons difficult. Data scaling is the process of aligning data values from different interval onto a specific interval. This allows the data to be compared more objectively.

One common method of data scaling is the minimum-maximum scaling method. This method aims to transform the data into a specific interval (usually [0,1]) by using the minimum and maximum values of the data (G. Ketepalli & P. Bulla, 2023). This way, all data points are normalized within a defined interval, and differences in the scales of various data points (features) are eliminated. This scaling process can increase the learning rate as well as reduce the learning time (Alasadi & Bhaya, 2017; Singh & Singh, 2020).

In this study, by using the minimum-maximum scaling method, the original data was scaled to the new data interval [0,1] with a linear transformation and made comparable. The training and testing data sets were separately scaled to ensure we would be observing real-life performance of the model during the testing phase. To accurately assess the model's real-world performance during testing, the training and testing datasets were scaled separately. This process aimed to improve both the learning rate and overall performance, while also minimizing the time needed for modeling. The equation of the minimum maximum scaling method applied in the study is given below (Yadav, 2021).

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

X': The feature's value after being transformed using Min-Max scaling and adjusted to a standardized range

X: The orginal, initial value of the feature in the dataset before it is scaled/normalized.

X_Min and **X_Max**: These show the features' minimum and maximum values.

3.7. Model Building and Training

One of the key stages that helps to reach the final result for the relevant problem is the building of the model correctly. Building the model with the correct parameter values allows the intended optimal final estimate results to be obtained.

In our study, after determining the best feature selection method, individual models were built using a total of 5 classifiers, including Random Forest, Logistic Regression, Decision Tree, LightGBM, and XGBoost classifiers, and then three basic classifiers that showed the highest performance among these models were determined. These are: XGBoost, Decision Tree, and LightGBM classifiers. Then, an ensemble model was built using these classifiers.

The majority voting approach, which integrates the prediction of multiple models to make a final decision, was used in the building of the ensemble model. During the building of models, a random seed parameter was set to 42 for reproducibility. The models were trained using scaled training data. Then, trained models' performances were assessed on the scaled testing data to determine and assess their generalization ability.

3.8. Proposed Model

Intrusion Detection Systems (IDSs) should meet the evolving requirements in advancing technology. Machine learning techniques, commonly favored in numerous studies, are also applied within this area. These techniques are employed in IDSs to achieve efficient classification using previously unseen data. Typically, IDSs need to manage extensive datasets that include multiple redundant features, resulting in decreased accuracy and prolonged processing times (Khammassi & Krichen, 2017; Thomas, 2018).

This study explored different techniques for feature selection and ensemble learning to develop a high-accuracy Intrusion Detection System for binary classification. The diagram in Figure 3.5 illustrates the proposed model designed for IDS. This model basically consists of 5 parts, which are: data preprocessing, data balancing, feature selection, model training and model evaluation.

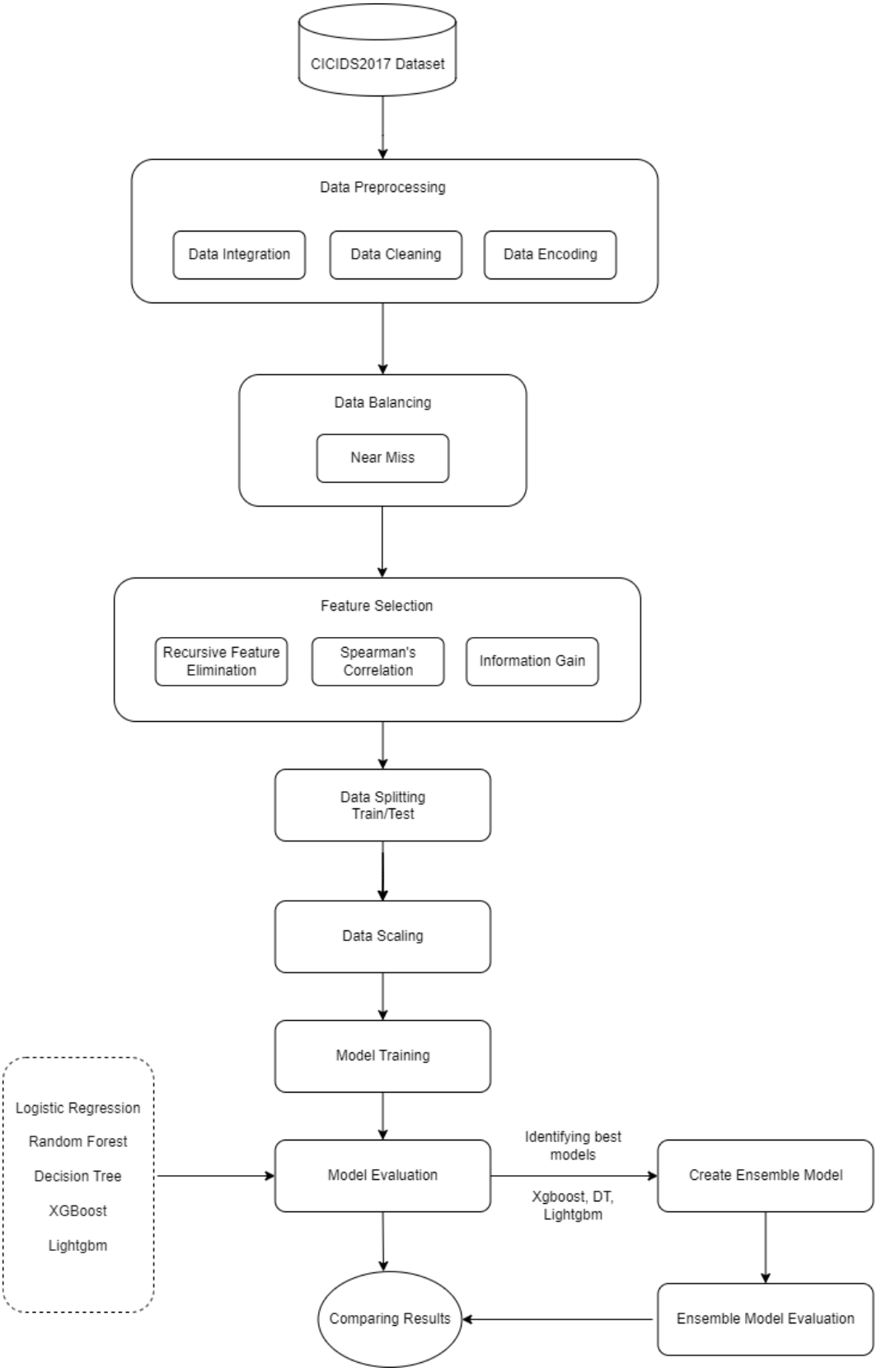

Figure 3.5 Proposed Model

In the data processing, initially, the dataset containing 14 different attack types across 8 separate files were merged into a single file. Subsequently, steps such as detecting/cleaning inconsistencies, handling missing values, removing repeated features from the dataset, and data encoding were applied to this merged dataset.

In the second part, the dataset was balanced using the Near Miss undersampling method. Thus, potential issues such as biased predictions due to data imbalance and overfitting were prevented, and challenges that might arise during the training and testing phases due to the large dataset size were also eliminated.

In the third part, the most significant features within the dataset were identified through the application of feature selection methods. The selected feature selection methods were implemented individually, and their results were analyzed. Results demonstrated that RFE is the most effective approach for feature selection, and utilizing a dataset with 40 features can enhance performance.

In the fourth part, the models were trained using the featured data obtained from the best-performing feature selection, and their performances were evaluated on the test data. Based on the evaluation, the top three classifiers with the highest performance were identified, and an ensemble model was developed using these classifiers.

In the final part, the performance of the developed ensemble model was evaluated and compared its results with the results of other models.

3.9. Model Evaluation

In this study, performance of developed intrusion detection models tested and evaluated on the testing set. Accuracy, precision, recall, f1-score metrics and confusion matrix were used to evaluate the performance of the models. Detailed information on the performance metrics used is given below.

Accuracy: This metric is used to measure the overall accuracy of the model's predictions. It illustrates the ratio of correctly predicted cases to the total instances within the dataset (Bhuyan, et al., 2014).

The equation of Accuracy is shown below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

Precision: It is a metric that measures how well the model can identify the positive cases (attacks) among those that it expected to be positive. It is calculated by using the ratio of true positives cases to the total of true positives and false positives cases (Bhuyan, et al., 2014; Sharafaldin, 2018).

The equation of Precision is shown below:

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

Recall: The model's ability to recognize each positive case is measured through recall, which is also referred to as sensitivity or the true positive rate. It demonstrated the proportion of true positive cases to the total of true positive and false negative cases (Bhuyan, et al., 2014; Sharafaldin,2018).

The equation of Recall is shown below:

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

F1 Score: This score shows the harmonic mean of Precision and Recall. It merges the two metrics to produce an overall outcome that encapsulates the performance of the model. A model that has a high F1-score indicates strong Recall and Precision, accurately identifying and classifying positive cases while reducing the occurrence of false positive and false negative cases (Bhuyan, et al., 2014; Sharafaldin, 2018).

The equation of F1 score is shown below:

$$f1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{3.5}$$

Confusion Matrix : This matrix displays the number of accurate and inaccurate predictions generated by the model based on the actual situations in the dataset. It provides insight into the model's accuracy. The dimension of the confusion matrix is determined by the number of classes in the classification (Bhuyan et al.,2014). In this study, since the network includes normal traffic data and traffic data containing attacks, a binary confusion matrix was used for evaluation. For a binary matrix, there are four possible outcomes. Figure 3.6 shows the confusion matrix used in this study.

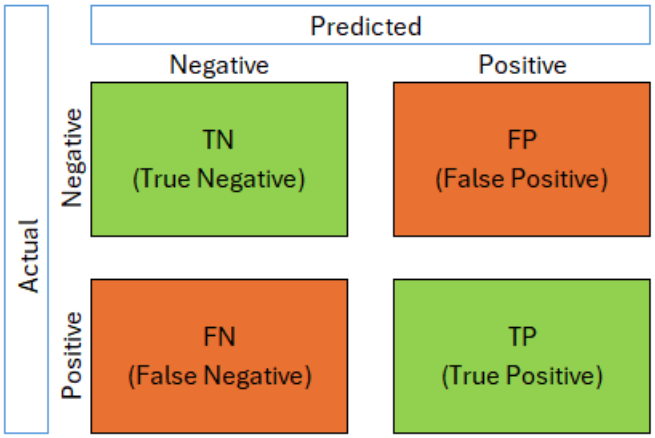

Figure 3.6 Confusion Matrix

True positive (TP): The situation where the data in the attack class is predicted as an attack by the established model.

False positive (FP): The situation where the data in the normal class is predicted as an attack class by the established model.

True negative (TN): The situation where the data in the normal class is predicted as normal class by the established model.

False Negative (FN): The situation where the data in the attack class is predicted normal class by the established model.

The values derived from the this matrix determines the performance criteria of the models. These values are utilized to calculate metrics such as accuracy, precision, Recall and F1-score.

3.10. Technical Environment

Our code was developed using Python programming language. In the programming phase, various libraries, and frameworks such as NumPy, Scikit-Learn, Matplotlib, Seaborn, Pandas within Python were utilized. Our code was executed on the cloud platform using a Jupyter notebook-based runtime environment, which was Collaboratory provided by Google. During executing phase of our code, NVIDIA Tesla T4 GPU was used.

4. RESULT AND DISCUSSION

In this chapter of the study, performance results of the established models are presented and discussed.

In this study, established models are tested on a testing set, and their performances are assessed using several performance measures, such as accuracy, precision, recall, f1-score, and error rate.

Furthermore, in order to improve the intrusion detection process and achieve better performance results, the features in the dataset to be used in model predictions were determined by applying RFE, Spearman's correlation analysis, and IG methods.

In the first part of the study, multiple experiments were conducted by using different thresholds values and feature numbers in order to determine the feature selection techniques and features numbers to be employed in the study. As a result of the experiments it was seen that selecting 40 features using RFE, 43 features using Spearman's correlation, and 60 features using IG improved the accuracy and f1-score while reducing the error rate.

Then, these feature selection results obtained using RFE, Spearman's correlation, and IG methods were compared with each other and also with the results obtained using the original dataset containing all the features (see Table.4.1).

Table 4.1 Performance results of models with and without feature selection

	Without FS			RFE			Spearman			IG		
Model	**Accuracy**	**F1 Score**	**Error Rate**	**Accuracy**	**F1 Score**	**Error Rate**	**Accuracy**	**F1 Score**	**Error Rate**	**Accuracy**	**F1 Score**	**Error Rate**
LG	0.9275	0.9279	0.0725	0.9278	0.9282	0.0722	0.9263	0.9265	0.0738	0.9192	0.9147	0.0808
RF	0.8880	0.8744	0.1120	0.8892	0.8759	0.1108	0.8845	0.8705	0.1155	0.8880	0.8745	0.1120
DT	0.9792	0.9792	0.0208	0.9877	0.9877	0.0123	0.9887	0.9887	0.0113	0.9392	0.9357	0.0608
XGBoost	0.9876	0.9876	0.0124	0.9975	0.9975	0.0025	0.9960	0.9960	0.0040	0.9975	0.9975	0.0025
LightGBM	0.9971	0.9970	0.0030	0.9977	0.9977	0.0023	0.9957	0.9957	0.0043	0.9971	0.9971	0.0030

When comparing the results obtained by using feature selection with the results obtained by using the original dataset (dataset without using feature selection), it is seen that RFE enhances the performance of all models regarding accuracy, f1-score, and error rate.

On the other hand, it is observed that Spearman's correlation improves the performance of the DT and XGBoost but decreases the performance of LG, RF, and LightGBM. Similarly, it is seen that IG results in a performance decline for LG and DT, while improving the performance of XGBoost and not having a significant effect on the performance of RF and LightGBM.

When the results of RFE, Spearman's correlation, and IG methods are compared with each other, it is seen that RFE generally stands out as the most effective feature selection method and offers the best accuracy, f1-score, and also lowest error rate for the most models. Although Spearman's correlation provides the best results for the DT model compared to RFE, it lags behind RFE in other models.

While IG produces similar results to RFE for the XGBoost model regarding accuracy, f1-score, and error rate, it results in lower accuracy, f1-score, and a higher error rate than RFE for other models.

On the other hand, IG outperforms Spearman's correlation in RF, XGBoost, and LightGBM models, where it provides higher accuracy, f1-scores, and lower error rates. However, when it comes to LG and DT, IG underperforms compared to Spearman's correlation, leading to lower accuracy, a lower f1-score, and a higher error rate.

Given the above information, it can be said that RFE enhances accuracy, f1 scores, and reduces error rates, making it the most effective feature selection method across a most of models.

Spearman's correlation also performs well, particularly in models like DT and XGBoost, but is slightly less effective than RFE. IG, while maintaining performance for ensemble methods like XGBoost and LightGBM, is less effective for DT and LG models, resulting in notable performance drops and higher error rates.

Therefore, it can be said that RFE generally emerges as the most robust and reliable method for feature selection, offering performance improvements across all models and maintaining the lowest error rates, followed by Spearman and Information Gain.

Given the information above, it has been concluded that when RFE is applied with appropriate parameters in machine learning-based IDSs, it can significantly enhance performance. Therefore, the ensemble model evaluations in the second phase of the study were conducted using the dataset composed of features selected through the RFE method.

Construction of the Proposed Ensemble Learning Model

In this study, the data set containing 40 features obtained from RFE feature selection method was used for the proposed new ensemble learning model.

In the study, the performance results of each individual model (RF, LG, DT, XGboost LightGBM) were evaluated regarding accuracy, recall, precision, f-score and error rates and the proposed new ensemble learning model was constructed with the 3 models having the best performance results.

Table 4.2 shows the results of model performances reached using the RFE method.

Table 4.2 Results of model performances reached using the RFE method

Model	Accuracy	Precision	Recall	F1 Score	Error Rate
Logistic Regression	0.9278	0.9214	0.9350	0.9282	0.0722
Random Forest	0.8892	**0.9931**	0.7834	0.8759	0.1108
Decision Tree	**0.9877**	0.9851	**0.9903**	**0.9877**	**0.0123**
XGboost	**0.9975**	**0.9963**	**0.9988**	**0.9975**	**0.0025**
LightGBM	**0.9977**	**0.9967**	**0.9986**	**0.9977**	**0.0023**

When the results in Table 4.2 are examined, it is seen that LightGBM and XGBoost have the highest accuracy values, with 0.9977 and 0.9975, respectively. The DT model follows these two models with an accuracy value of 0.9877. These results show that the models' general performances are quite good and their ability to make correct classifications is high.

Similarly, LightGBM and XGBoost have the highest precision values with 0.9967 and 0.9963, respectively. RF model comes next with a precision value of 0.9831. These results outline clearly that these models are quite successful in accurately identifying true positives (attacks).

In terms of recall, the highest scores belong to XGBoost, LightGBM, and DT models with values of 0.9988, 0.9986, and 0.9903, respectively. This result indicates that these models have very low false negative rates in detection of attacks.

Similarly, XGBoost and LightGBM have the highest f1 scores here again, with 0.9975 and 0.9977, respectively. DT follows these models with an f1-score of 0.9877. These results clearly suggest that these three models maintain a pretty good balance between precision and recall when compared to others.

Furthermore, LightGBM has the lowest error rate at 0.0023. Then, XGBoost follows these models with error rate of 0.0025. DT also shows a reasonable error rate of 0.0123. This means that these models make very few errors in classifying both attacks and normal records compared to the other models.

On the other side, LG and RF models represent relatively lower performances compared to these three models. In particular, RF demonstrates the lowest accuracy at 0.8892 and the highest error rate at 0.1108, with lower performance across all metrics except for precision. Similarly, LG also underperformes relative to the other three models-LightGBM, XGBoost, and DT-across all evaluation metrics.

The graphical representation of the results is also given in Figure 4.1.

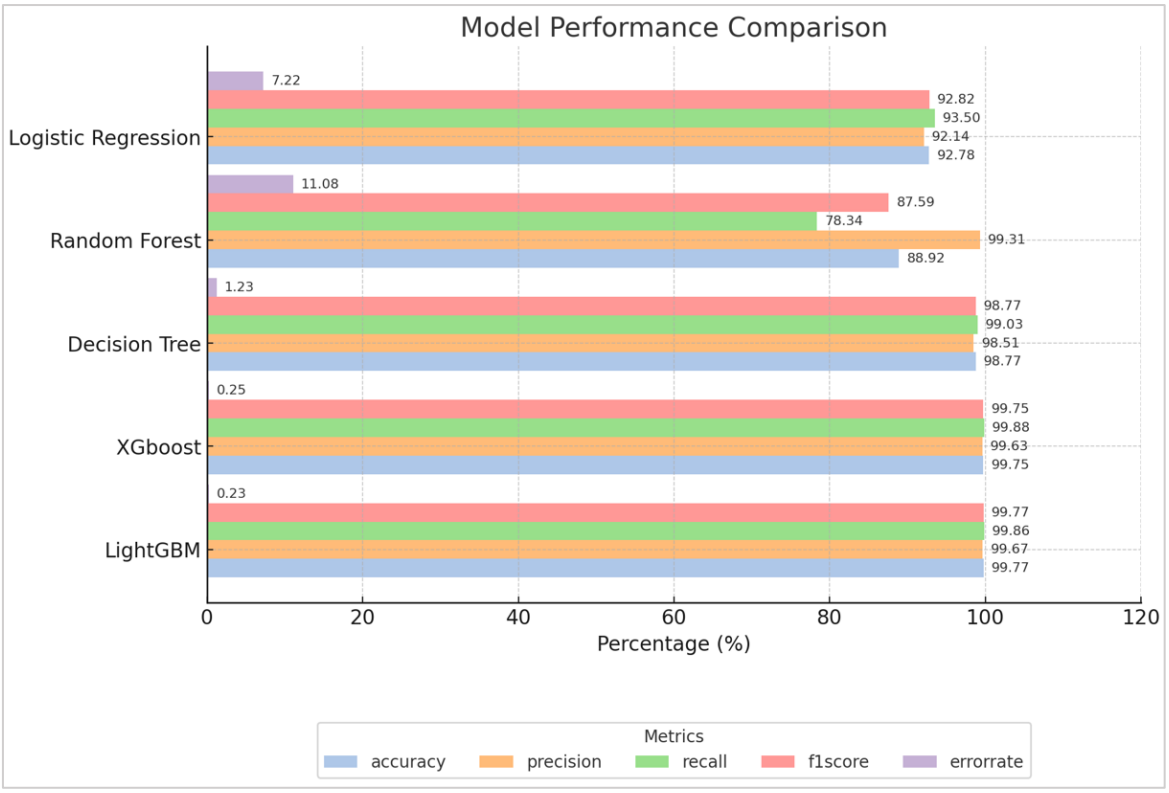

Figure 4.1 Visualization of model performances results reached using the RFE method

Based on these results, the proposed new model was constructed using LightGBM, XGBoost, and DT, as these three models provide the most favorable performance outcomes and overall success.

Then, the results of the performance of the proposed new ensemble learning model constructed using these three selected models and RFE feature selection were compared with the results of other individual models.

Table 4.3 presents the performance results of models

Table 4.3 Result of performance results of Ensemble model and indivual models

Model	Accuracy	Precision	Recall	F1 Score	Error Rate
Logistic Regression	0.9278	0.9214	0.9350	0.9282	0.0722
Random Forest	0.8892	0.9931	0.7834	0.8759	0.1108
Decision Tree	0.9877	0.9851	0.9903	0.9877	0.0123
XGboost	0.9975	0.9963	**0.9988**	0.9975	0.0025
LightGBM	0.9977	0.9967	0.9986	0.9977	0.0023
Ensemble Model	**0.9978**	**0.9968**	0.9987	**0.9978**	**0.0022**

When we look at the results of the performance metrics in Table 4.3, it is seen that the ensemble model that we proposed outperforms other individual models in intrusion detection.

When the results are examined in detail, it is seen that our proposed model exhibits the highest accuracy value of 0.9978 compared to other models. This result indicates that the model correctly classifies nearly all instances, both normal and attack, and its overall performance is quite high. In particular, it outperforms even high-performing models such as XGBoost (0.9975) and LightGBM (0.9977), which shows that the ensemble model provides the best performance in classification tasks and is superior to other models.

Moreover, the model has the highest precision value of 0.9968. This means that compared to the individual models, the model performs better in minimizing false positives. This ability is notably important in sensitive fields like security, where false alarms may lead to taking many unnecessary actions and wasting resources.

Additionally, following XGBoost (0.9988), the proposed model achieves the highest recall value of 0.9987. This high recall indeed shows that the model successfully detects almost all attacks, with minimal risk of overlooking any real incidents. Such performance underscores the model's reliability, particularly in environments where accurate attack detection is critical.

Similarly, in terms of f1 score, the proposed model outperforms other models. With a high f1 score of 0.9978, the model exhibits a balanced performance in both detecting true positives (recall) and

preventing false alarms (precision). This indicates that the model is very successful in both detecting attacks and minimizing false positives.

The model's error rate is also very low at 0.22%. This means that model mostly makes accurate predictions with infrequent mistakes. Moreover, this relatively small error rate enhances further the model's reliability. Notably, it outperforms both XGBoost (0.25%) and LightGBM (0.23%) in terms of error rate, suggesting that the ensemble model delivers more consistent and dependable results.

Consequently, it can be stated that the proposed new ensemble learning model is quite effective and successful in both detecting attacks correctly and minimizing false alarms with its high accuracy, precision, recall, f1 score, and low error rate. This high performance of the model makes it a more reliable and effective option in attack detection compared to other individual models (LG, RF, DT, XGBoost, LightGBM).

Figure 4.2 also provides a graphical representation of the results mentioned above.

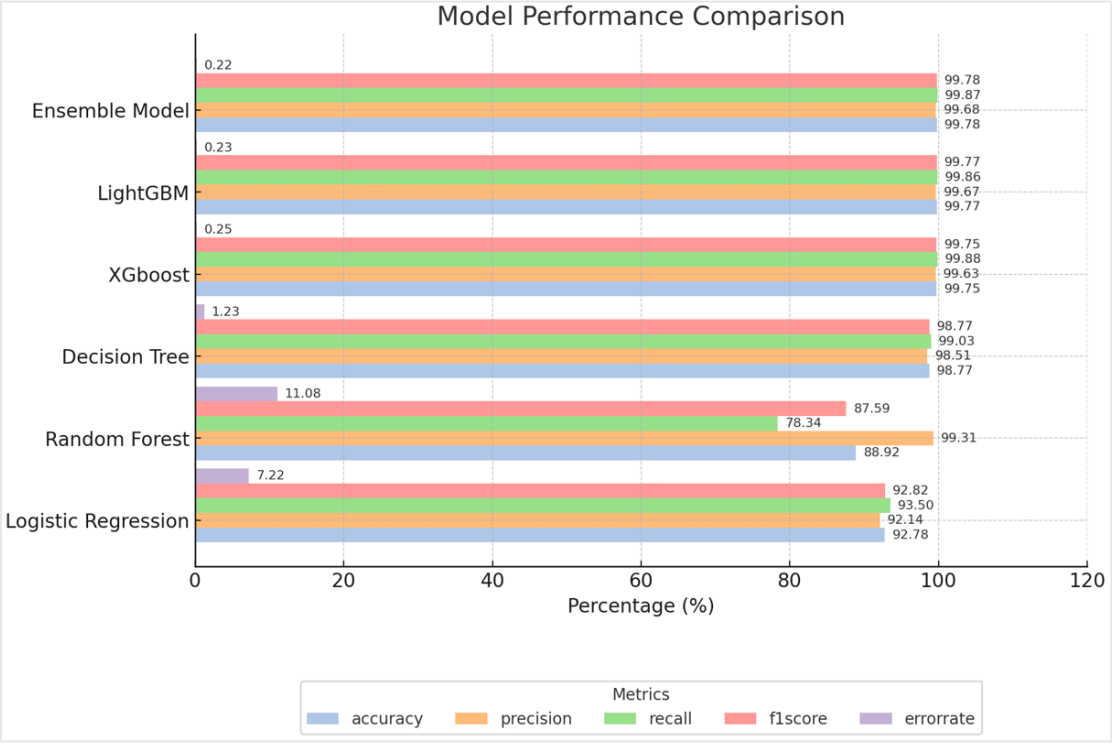

Figure 4.2 Result of performance results of Ensemble model and indivual models

Furthermore, the confusion matrix of the model we proposed is also presented and discussed below.

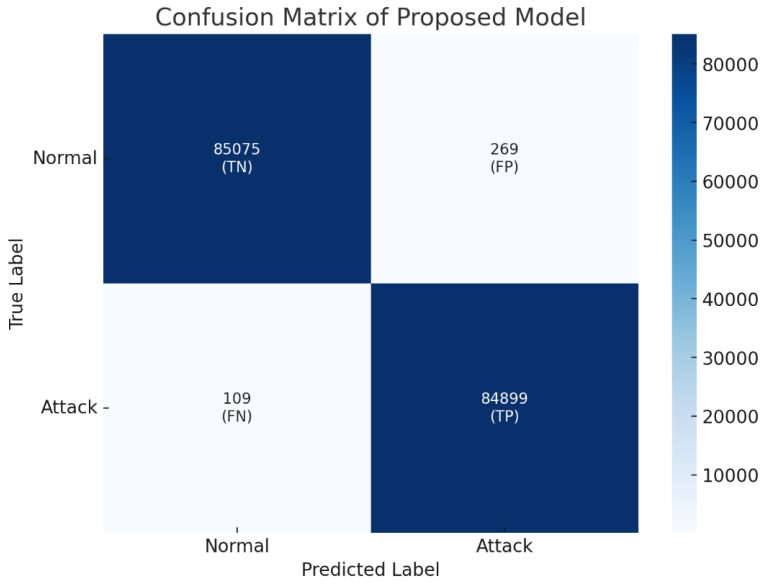

Figure 4.3. Confusion matrix of proposed model

When examining the confusion matrix of the model we proposed, it is seen that it reveals an outstanding performance across key metrics. The True Positive (TP) rate is impressively high at 99.87%, demonstrating the model's strong ability to correctly identify actual attacks, with 84,899 out of 85,008 attack cases correctly classified. Equally notable is the True Negative (TN) rate, which stands robustly at 99.72%, indicating the model's effectiveness in accurately recognizing non-intrusive, normal cases, with 85,075 out of 85,344 normal cases correctly classified. The False Positive (FP) rate is remarkably low at just 0.315%, meaning the model seldom raises false alarms, with only 269 normal cases mistakenly identified as attacks. Similarly, the False Negative (FN) rate is extremely low and equal to only 0.128%, reflecting that a limited number of actual attacks—exactly 109 cases—did not get detected. These results emphasize the model's robust and balanced performance in accurately differentiating between positive cases (attacks) and negative cases (non-attacks). This makes it a highly effective option for reliable intrusion detection.

5. CONCLUSION

The rapid advancement of the internet has resulted in a substantial rise in number and complexity of cyber attacks. This situation has made intrusion detection systems (IDS) more crucial than ever for protecting network infrastructures. However, traditional IDSs have fallen short in the face of the growing complexity and diversity of cyber attacks. This situation has highlighted the need for more advanced technologies to ensure network security.

Machine learning algorithms present significant potentials to overcome the limitations of traditional IDS and enhance their capabilities. By processing large datasets, these algorithms can help detect anomalies and identify previously unknown attack types, thereby making IDSs more effective and adaptive. By leveraging machine learning techniques, IDSs can better respond to the constantly evolving landscape of cyber threats, making them a key component in modern cybersecurity strategies.

This study aims to develop an effective and high-performance Machine Learning-based IDS by combining the advantages of various techniques such as feature selection, pre-processing, class balancing, and ensemble learning.

The implementation phase of this study utilized the CICIDS2017 dataset. In the initial part of the study, pre-processing steps, including data integration, data cleaning, and data encoding, were conducted. During this process, missing, redundant, and inconsistent features were removed, categorical variables were converted into numerical values, thereby improving data quality and ensuring the dataset was optimized for analysis and model training.

Following that, the dataset was balanced by using the Near Miss undersampling method. Thus, potential issues such as biased predictions due to data imbalance and overfitting were prevented, and challenges that might arise during the training and testing phases due to the large dataset size were also eliminated.

The most significant features within the dataset were identified through the application of feature selection methods, including Recursive Feature Elimination, Spearman's Correlation Analysis, and Information Gain. This approach aimed to improve the attack detection process and performance by identifying the most important features and eliminating unnecessary and insignificant features from the dataset.

To assess the impact of the analyzed feature selection techniques on the performance of the intrusion detection models, experiments were performed on newly generated datasets using the features identified by each method. LG, DT, RF, XGboost, LightGBM classifiers were used to compare the results obtained using the new data sets with the results obtained using the original data set.

In the results obtained, it was seen that RFE enhances the performance of all models (LR, DT, RF, XGboost and LightGBM), regarding accuracy, f1-score, and error rate compared to results obtained using the full dataset.

In contrast, it was observed that Spearman's correlation improves the performance of DT and XGBoost but decreases the performance of LR, RF, and LightGBM. Similarly, it was seen that IG results in a performance decline for LR and DT, while improving the performance of XGBoost and not having significant impact on the performance of RF and LightGBM model.

In addition, the performance outcomes of RFE, Spearman's correlation, and IG techniques were compared with each other. Result of the comparison revealed that RFE generally outperforms other feature selection methods, with the best accuracy, f-score, and lowest error rate. Furthermore, it was observed that Spearman's correlation yields the best results for DT model compared to RFE, while it falls behind RFE in other models.

Moreover, it was seen that while IG produces similar results to RFE for the XGBoost model regarding accuracy, f1-score, and error rate; it results in lower accuracy, f1-score, and a higher error rate than RFE for other models.

On the other hand, it was observed that IG outperforms Spearman's correlation in RF, XGBoost, and LightGBM models, providing higher accuracy, f1 scores, and lower error rates. However, it was seen

that IG underperforms compared to Spearman's correlation analysis in LG and DT models, resulting in a substantial drop in accuracy and f1 scores, along with a notably higher error rate.

Given all of this information, it can be said that RFE generally comes out as the most effective method for feature selection, giving performance improvements across all models while keeping their error rates at the lowest.

Thus, it can stated that enhancing the performance of machine learning-based intrusion detection systems can be accomplished by employing the RFE technique with appropriately configured parameters.

In the next stage of the study, a new ensemble learning model was constructed that integrates the advantages of individual classifiers in order to increase accuracy and intrusion detection performance. This new model was constructed using classifiers chosen according to the performance results from the initial phase of the study. In order to assess the model's performance, the dataset that consists of 40 features determined as a result of the RFE given above was used.

When evaluating the performance results, it was observed that XGBoost, LightGBM, and DT classifiers demonstrate better performance and success compared to other classifiers in intrusion detection. Therefore the proposed ensemble learning model was constructed using these three classifiers and its performance results were compared with the performance results of individual models.

Performance evaluation showed that our proposed ensemble learning model outperformed the other individual models in intrusion detection. Based on the information obtained from the analysis, it can be stated that our proposed model is highly successful in both accurately detecting attacks and minimizing false alarms, thanks to its high accuracy, precision, recall, f1 score, and low error rate.

Given all of this information, it can be said that our proposed model possesses significant potential for use in intrusion detection systems (IDS) aimed at safeguarding network infrastructure. By combining different machine learning techniques, feature selection methods, and ensemble learning strategies, our model capitalizes on the strengths of each approach, providing a flexible and robust

solution for network security professionals. Moreover, its enhanced accuracy, coupled with reduced rates of false negatives and false positives, positions it as a reliable option for detecting and addressing potential threats. This ultimately can helps minimize the risk of security breaches and mitigates the operational impact of false alerts.

5.1. Research Questions

This section of the thesis includes the answers to the research questions formulated in the introduction section of the thesis.

RQ1: How do feature selection techniques affect the performance of intrusion detection models?

RQ1 is addressed in Chapter 4. Based on the findings and results in this chapter, it can be said that feature selection techniques can have varying effects on different models' performances. Findings and results demonstrate that RFE techniques improve the performance of all intrusion detection models regarding accuracy, f1-score, and error rate compared to results obtained using the original dataset.

Additionally, it can be stated that Spearman's correlation boosts the performance of Decision Tree and XGBoost models but leads to performance declines in Logistic Regression, Random Forest, and LightGBM models.

Similarly, it can be said that Information Gain improves the performance of XGBoost but decreases the performance of Logistic Regression and Decision Tree, while having a minimal impact on Random Forest and LightGBM models.

RQ2: Among RFE, Spearman's Correlation Analysis, and IG techniques, which technique produces the superior outcomes?

The RQ2 is detailed in Chapter 4. Based on the findings and results in this chapter, it can be said that RFE generally stands out as the most effective feature selection method and offers the highest accuracy, f1-score, and lowest error rate for the most models.

Spearman's correlation also performs well, particularly in models like DT and XGBoost, but is slightly less effective than RFE. IG, while maintaining performance for XGBoost, LightGBM, and Random Forest, is less effective for DT and LG models, resulting in notable performance drops and higher error rates.

Overall, it can be said that RFE provides more consistent performance gains across different models and produces superior outcomes than Spearman's correlation analysis and IG.

RQ3: How does the proposed model that integrates the benefits of efficient preprocessing, class balancing, feature selection, and an ensemble learning approach perform compared to individual models in the network intrusion detection?

RQ3 is addressed in Chapter 4. Based on the findings and results in this chapter, it can be said that the proposed model outperforms all individual models (LG, RF, DT, XGBoost, and LightGBM) in network intrusion detection.

The proposed model attains the highest accuracy, precision, f1-score, and lowest error rate. This demonstrates that the model has a strong ability to make correct predictions overall and superior effectiveness in minimizing false positives (false alarms) and capturing true positives (attacks). Although XGBoost achieves a slightly higher recall value compared to the proposed model, the proposed model delivers the best overall results across the other metrics, proving to be the most effective model for network intrusion detection according to the other individual models.

5.2. Limitations

A limitation of our research is that it requires a strong working environment that can effectively manage large datasets. Systems with inadequate GPU capabilities might face challenges while executing the models. To mitigate this issue, it is recommended to employ high-performance computing systems with ample GPU resources. Another option is to use cloud computing services that provide scalable resources, which can facilitate the seamless running of models on extensive datasets.

5.3. Recommendations

While the results obtained in our current research are promising, several recommendations are provided to help future research achieve more advanced performance in this area. These recommendations are presented below:

- It is recommended that future research focus on developing hybrid models that combine different machine learning and deep learning techniques. This approach can balance the strengths and weaknesses of the models, providing a more comprehensive and effective intrusion detection system.

- It is recommended to use hybrid methods in the feature selection process. Hybrid feature selection techniques combine the advantages of both filtering and wrapping methods, helping the model select the most relevant features and thereby achieve higher performance.

- It is recommended to test the developed IDS models with real-time datasets instead of pre-existing datasets. This would allow for overcoming various practical challenges and assessing how the model performs under real-world conditions.

REFERENCES

Ajiya, A., Boukari, S., Bello, A., Mustapha, A. & Muhammad, A. (2021). A Survey of Intrusion Detection Techniques on Software Defined Networking (SDN*).International Journal of Innovative Science and Research Technology*, 6(8), 521-530.

Aksoy, N. & Genc, İ. (2023). Predictive models development using gradient boosting based methods for solar power plants. *Journal of Computational Science*, 67, 2023.
Available at: https://doi.org/10.1016/j.jocs.2023.101958.

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences,* 12(16), 4102-4107.

Alfrhan, A. , Alhusaın, H. R & Khan, R., U. (2020). SMOTE: Class Imbalance Problem In Intrusion Detection System. *2020 International Conference on Computing and Information Technology (ICCIT-1441)*, 1-5. Doi: 10.1109/ICCIT-144147971.2020.9213728

Aleroud, A., & Karabatis, G. (2017). Contextual information fusion for intrusion detection: a survey and taxonomy. *Knowledge and Information Systems*, 52(3), 563-619. Available at: https://doi.org/10.1007/s10115-017-1027-3

Ampomah, E., Qin, Z. & Nyame, G. (2020). Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information*. 11(6), 332.
Available at: https://doi.org/10.3390/info11060332

Awad, M &Alabdallah, A. (2019). Addressing imbalanced classes problem of intrusion detection system using weighted Extreme Learning Machine," *International Journal of Computer Networks and Communications*, 11(5), 39-58. Doi: 10.5121/ijcnc.2019.11503

Basnet R., Mukkamala S. & Sung A. H. (2008). Detection of Phishing Attacks: A Machine Learning Approach. *Studies in Fuzziness and Soft Computing, 226, 373-383*. Available at: https://doi.org/10.1007/978-3-540-77465-5_19

Bhuyan, M.H, Bhattacharyya, D.K, & Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools, *IEEE Communications Surveys & Tutorials*, 16(1),303-336.
Doi: 10.1109/SURV.2013.052213.00046

Biju, J. M., Gopal, N., Prakash, A. J. (2019). Cyber attacks and its different types. *International Research Journal of Engineering and Technology,* 6(3), 4849-4852.

Breiman L., (2001). Random forests,machine learning. *Kluwer Academic Publishers*, 45(1), 5-32. Available at: https://doi.org/10.1023/A:1010933404324

Breiman L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140.
Available at: https://doi.org/10.1007/BF00058655

Budak, H. (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(10). Doi:10.19113/sdufbed.01653

Chien, C. F., &Chen, L. F. (2008). Data Mining to Improve Personnel Selection and Enhance Human Capital: A Case Study in HighTechnology Industry, *Expert Systems with Applications,* 34 (1), 280-290. Available at: https://doi.org/10.1016/j.eswa.2006.09.003

Chen, T., Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. Available at: https://doi.org/10.1145/2939672.2939785

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

Domingues, I., A., Abreu, J. P., Duarte, P. H. & Santos, J. (2018). Evaluation of Oversampling Data Balancing Techniques in the Context of Ordinal Classification. *2018 International Joint Conference on Neural Networks (IJCNN),* 1-8. Available at: https://doi.org/10.1109/IJCNN.2018.8489599

Dubey,G. P., Bhujade,R. K. (2021). Optimal feature selection for machine learning based intrusion detection system by exploiting attribute dependence, *Materials Today: Proceedings*, 47(1), 6325-6331. Available at: https://doi.org/10.1016/j.matpr.2021.04.643

Elmasry, W. Akbulut, A. & Zaim, A.H (2019). Empirical study on multiclass classification-based network intrusion detection. *Computational Intelligence.* 35, 919–954. Available at: https://doi.org/10.1111/coin.12220

Fawagreh, K., Gaber, M. M., Elyan, E. (2014). Random forests: from early developments to recent advancements. *Systems Science & Control Engineering*, 2 (1), 602-609. Available at: https://doi.org/10.1080/21642583.2014.956265

Gao, X., Shan, Hu, C., Niu, Z. & Liu, Z. (2019). An Adaptive Ensemble Machine Learning Model for Intrusion Detection. *IEEE Access*, 7. Doi: 10.1109/ACCESS.2019.2923640

Gong, D & Liu. Y. (2022). A Mechine Learning Approach for Botnet Detection Using LightGBM. *2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL & ICCEA)*, 829-833. Available at: https://doi.org/10.1109/CVIDLICCEA56201.2022.9824033

Govindarajan, M. (2014). Hybrid Intrusion Detection Using Ensemble of Classification Methods *International Journal of Computer Network and Information Security,* 6(2), 45-53. Available at: https://doi.org/10.5815/ijcnis.2014.02.07

Gupta, S., Singhal A. & Kapoor, A. (2016). A literature survey on social engineering attacks: Phishing attack. *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 537-540. doi: 10.1109/CCAA.2016.7813778.

Hall, M. (1999). Correlation-based Feature Selection for Machine Learning. The University of Waikato, PhD Thesis, Hamilton.

Han, J., & Kamber, M. (2000*). Data Mining Concepts and Techniques* (1st ed), Morgan Kaufmann.

Harman, L. B., Flite, C. A., & Bond, K. (2012). Electronic health records: privacy, confidentiality, and security. *AMA Kournal of Ethics*, 14(9), 712-719. Doi: 10.1001/virtualmentor.2012.14.9.stas1-1209.

Hasan, M., Balbahaith, Z., & Tarique, M. (2019). Detection of SQL Injection Attacks: A Machine learning approach. *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA),* 1-6. Available at: https://doi.org/10.1109/ICECTA48151.2019.8959617

Hosmer, D. W., Stanley L. & Rodney X., S. (2013). *Applied Logistic Regression.* John Wiley & Sons.

Hota, H.S., Shrivas, A.K. (2014). Decision Tree Techniques Applied on NSL-KDD Data and Its Comparison with Various Feature Selection Techniques. In: Kumar K., Mohapatra, M, Konar,D., Chakraborty, A. (eds) *Advanced Computing, Networking and Informatics*, 27, 205–211.
Available at:https://doi.org/10.1007/978-3-319-07353-8_24

International Organization for Standardization (2022), *Information security, cybersecurity and privacy protection — Information security management systems — Requirements* (ISO Standard No. 27001:2022) Available at: https://www.iso.org/standard/27001

Jain N., Singh, M. P., Chaurasia, K. & Ravindran, G. (2023). Intrusion Detection System Using Ensemble Techinque. *2023 3rd International Conference on Innovative Sustainable Computational Technologies*, 1-5, doi: 10.1109/CISCT57197.2023.10351427.

Jadhav, A., M. Mostafa, S., Elmannai, H., & Karim, F. K. (2022). An Empirical Assessment of Performance of Data Balancing Techniques in Classification Task. *Applied Sciences*, 12(8).
https://doi.org/10.3390/app12083928

Jaw E. & Wang, X. (2021). Feature Selection and Ensemble-Based Intrusion Detection System: An Efficient and Comprehensive Approach. *Symmetry*. 13(10), 1764.
Available at: **https://doi.org/10.3390/info11060332**

Jayalaxmi P., Saha R., Kumar G., Conti M & Kim T. H. (2022). Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. *IEEE Access*. 99, 1-1, 2022.
Doi: 10.1109/ACCESS.2022.3220622

Kafi M.A. & Akter N. (2023). Securing financial information in the digital realm: case studies in cybersecurity for accounting data protection. *American Journal of Trade and Policy,* 10(1):15–26. https://doi.org/10.18034/ajtp.v10i1.659

Khan, M. Y. & Qayoom, A., Nizami, M., Siddiqui, M. S., Wasi, S. & Syed, K. R. R. (2021). Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques, *Complexity*. 2021, 1-18.
Available at: https://doi.org/10.1155/2021/2553199

Khammassi, C. & Krichen, S. (2017). A GA-LR wrapper approach for feature selection in network intrusion detection, *Computers & Security*, 70, 255–277. Doi: 10.1016/j.cose.2017.06.005

Kara, M & Necati Ş. (2011). Fighting with Botnets in the World and Turkey, Conference of Academic Computing.

Kasongo S. M., Sun Y., (2019). A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection System, *IEEE Access*, 7, 38597 38607. Doi: 10.1109/ACCESS.2019.2905633.

Kasongo, S.M. & Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, 7(2020), 1–20. Doi: 10.1186/s40537-020-00379-6

Kaynar, O., Arslan, H., Görmez, Y., IŞIK, Y. E. (2018). Intrusion Detection with Machine Learning and Feature Selection Methods. *Bilişim Teknolojileri Dergisi*, 11 (2): 175-185.
Available at: https://doi.org/10.17671/gazibtd.368583

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Conference: Advances in Neural Information Processing Systems*, 3149-3157. Aveilable at: https://dl.acm.org/doi/pdf/10.5555/3294996.3295074

Kearns, M. (1988). Thoughts on hypothesis boosting, Machine Learning Class Project. Available at: https://www.cis.upenn.edu/~mkearns/papers/boostnote.pdf.
Kemalis K., & Tzouramanis T. (2008). SQL-IDS: A specification-based approach for SQL İnjection detection, *ACM symposium on Applied computing,* 2153-2158.

Ketepalli. G, & Bulla, P. (2023). Data Preparation and Pre-processing of Intrusion Detection Datasets using Machine Learning. *2023 International Conference on Inventive Computation Technologies (ICICT)*, 257-262. Doi: 10.1109/ICICT57646.2023.10134025.

Khan A., Rehman M., Rutvij H., Jhaveri, R., Raut, T, Saba S.A. (2022). Deep learning for intrusion detection and security of Internet of things (IoT): current analysis, challenges, and possible solutions. *Security and Communication Network*s, 2022(4), 1-13. Doi: 10.1155/2022/4016073

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2), 181–207.
Available at https://doi.org/10.1023/A:1022859003006

Kim, D., Solomon, M. G. (2018). *Fundamentals of Information Systems Security* (3rd ed). Jones&Bartlett.

Kurniabudi, K., Stiawan, D., Darmawijoyo Dr., Mohd I. (2020). CICIDS-2017 dataset feature analysis with ınformation gain for anomaly detection. *IEEE Access*. 1-1. 10.1109/ACCESS.2020.3009843.

Lalduhsaka, R., Bora, N. & Khan, A. (2022). Anomaly-Based Intrusion Detection Using Machine Learning: An Ensemble Approach. *International Journal of Information Security and Privacy*, 16, 1-15.
Available at 10.4018/IJISP.311466.

Li, W., Yin, Y., Quan, X., & Zhang, H. (2019). Gene expression value prediction based on XGBoost algorithm. *Frontiers in Genetics*, 10, 1077, 1-7.
Available at https://doi.org/10.3389/fgene.2019.01077

Liu, H. & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge & Data Engineering,* 17(4), 491–502.
Doi: 10.1109/TKDE.2005.66

Liu H., & Lang B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences*, 2019, 9(20), 4396. https://doi.org/10.3390/app9204396
Doi: 10.3390/app9204396

Lyu, Y., Feng, Y., & Sakurai, K. (2023). A Survey on Feature Selection Techniques Based on Filtering Methods for Cyber Attack Detection. *Information,* 14(3), 191. Available at https://doi.org/10.3390/info14030191

Mienye, D. & Jere, N. (2024). A Survey of Decision Trees: Concepts, Algorithms, and Applications. *IEEE Access*. 4, 86716 - 86727. Doi: 10.1109/ACCESS.2024.3416838

Najafabadi, M., Khoshgoftaar, T., Kemp, Clifford & Seliya, N. & Zuech, R. (2014). Machine Learning for Detecting Brute Force Attacks at the Network Level. *2014 IEEE International Conference on Bioinformatics and Bioengineering.* 379-385. 10.1109/BIBE.2014.73.

Neil A. (2007). Network infiltration with client-side attacks, *Network Security*, 2007(9), 8-10.
Available at: https://doi.org/10.1016/S1353-4858(07)70081-8.

Nimbalkar, P. & Kshirsagar, D. (2021). Feature selection for intrusion detection system in internet of things (IOT). *ICT Express*, 2021;7(2), 177–81.
Available at: https://doi.org/10.1016/j.icte.2021.04.012

Parsaei, M. R., Rostami, S. M. & Javidan, R. (2016). A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 7(6), 2016. Available at : http://dx.doi.org/10.14569/IJACSA.2016.070603

Paulauskas, N. & Auskalnis, J. (2017). Analysis of data pre-processing influence on intrusion detection using NSL-KDD dataset. *2017 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–5. Doi: 10.1109/eStream.2017.7950325

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine,*6(3), 21–45. Doi: 10.1109/MCAS.2006.1688199

Peng, J. (2002). An introduction to logistic regression analysis and reporting. *The Journal of Educational Research*. *96*(1), 3-14. Available at: https://doi.org/10.1080/00220670209598786

Rajasekaran, K. & Nirmala, K. (2012). Classification and importance of intrusion detection system (IJCSIS) *International Journal of Computer Science and Information Security*, 10(8), 44-46. Available at: https://www.researchgate.net/publication/340655192_Classification_and_Importance_of_Intrusion_Detection_System

Raihan-Al-Masud, M & Mustafa, H. A. (2019). Network Intrusion Detection System Using Voting Ensemble Machine Learning. *2019 IEEE International Conference on Telecommunications and Photonics (ICTP),* 1-4. doi: 10.1109/ICTP48844.2019.9041736.

Raschka S., & Mirjalili V. (2017). *Python Machine Learning* (2nd ed). Puckt Publishing.

Ren, S. Q., Tan, B. H. M., Sundaram, S., Wang, T., Ng, Y., Chang, V., & Aung, K.M.M. (2016). Secure searching on cloud storage enhanced by homomorphic indexing. *Future Generation Computer Systems*, 65, 102–110. Available at : https://doi.org/10.1016/j.future.2016.03.013

Rufo, D. D., Debelee, T. G., Ibenthal, A., & Negera, W. G. (2021). Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM). *Diagnostics*, 11(9), 1714.
Available at: https://doi.org/10.3390/diagnostics11091714

Sharafaldin, I. , Lashkari, A. H. & Ali Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *4th International Conference on Information Systems Security and Privacy (ICISSP),* 1, 108-116. Doi. 10.5220/0006639801080116

Stallings, W. Brown, L. M. D. Bauer, & Bhattacharjee, A.K. (2012). *Computer security: principles and practice.* Pearson Education.

Saxena, A. K., Sinha, S., & Shukla, P. (2017). General study of intrusion detection system and survey of agent based intrusion detection system. *International Conference on Computing, Communication and Automation*, 421-471. https://doi.org/10.1109/CCAA.2017.8229866

Scikit Learn. (2022). Sklearn Feature Selection RFE. Retrieved in 2024 from: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

Sidharth, V. & Kavitha. C, R. (2021). Network Intrusion Detection System Using Stacking and Boosting Ensemble Methods. *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*. 357-363, doi: 10.1109/ICIRCA51532.2021.9545022.

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*. https://doi.org/10.1016/j.asoc.2019.105524

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review* 33(1-2), 1-39.
Doi: 10.1007/s10462-009-9124-7.

Sutton, C. D. (2005). Classification and Regression Trees, Bagging, and Boosting. İçinde: Handbook of Statistics. Elsevier Masson SAS, pp. 303–329. doi: 10.1016/S0169 7161(04)24011-1. https://doi.org/10.1016/S0169-7161(04)24011-1

Tama, B. A., & Rhee, K. H. (2017). Performance evaluation of intrusion detection system using classifier ensembles, *International Journal of Internet Protocol Technology*, 10(1), 22-29.
Doi:10.1504/IJIPT.2017.10003843

Talukder M.A., Hasan K.F., Islam M.M., et al.(2023). A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications*, 72(103),405.
Available at : https://doi.org/10.1016/j.jisa.2022.103405

Thomas, R. & Pavithran, D. (2018). A Survey of Intrusion Detection Models based on NSL-KDD Data Set. *2018 Fifth HCT Information Technology Trends (ITT).*286-291. doi : 10.1109/CTIT.2018.8649498.

Wearesocial (2024). The Global State of Digital in April 2024.
Available at: https://wearesocial.com/uk/blog/2024/01/digital-2024/

Wolpert D.H. (1992). Stacked Generalization. *Neural Networks,* 5(2), 241-59.
Available at : https://doi.org/10.1016/S0893-6080(05)80023-1

Yadav, S.M. (2021). A Survey on Network Intrusion Detection Using Deep Generative Networks for Cyber-Physical Systems. Artificial Intelligence Paradigms for Smart Cyber-Physical Systems, 137-159 Doi:10.4018/978-1-7998-5101-1.ch007

Yueai, Z. & Junjie, C. (2009). Application of unbalanced data approach to networkintrusion detection. *First International Workshop on Database Technology and Applications, DBTA,* 140-143.
Doi: 10.1109/DBTA.2009.116

Zhang, J.P. & Mani, I. (2003). KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. *Proceeding of International Conference on Machine Learning (ICML 2003),* Workshop on Learning from Imbalanced Data Sets, Washington DC, 21 August 2003.

Zhou, Z. H. (2012). *Ensemble Methods: Foundations and Algorithms* (1st ed). Chapman and Hall/CRC.

Zhou, Y., Cheng, G., Jiang, S. & Dai, M. (2019). An Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier, 14(4), 1-12. Doi: 10.48550/arXiv.1904.01352

Zhou, Y., Cheng, G., Jiang, S. & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, 174.
https://doi.org/10.1016/j.comnet.2020.107247.