IDENTIFICATION OF DDOS ATTACK THROUGH INTRUSION DETECTION MODEL USING ENSEMBLE MACHINE LEARNING

ADEEBA ANIS (SN. 1017140010)

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MILITARY INSTITUTE OF SCIENCE AND TECHNOLOGY
DHAKA, BANGLADESH

January 2024

IDENTIFICATION OF DDOS ATTACK THROUGH INTRUSION DETECTION MODEL USING ENSEMBLE MACHINE LEARNING

M.Sc. Engineering Thesis

By

ADEEBA ANIS (SN. 1017140010)

Approved as to style and content by the Board of Examination on 18 January 2024:

Dr. Md. Shohrab Hossain Professor of Computer Science and Engineering BUET, Dhaka	Chairman (Supervisor) Board of Examination
Tasmiah Tamzid Anannya Lecturer of Computer Science and Engineering MIST, Dhaka	Member (Co-Supervisor) Board of Examination
Dr. Fazlul Hasan Siddiqui Professor of Computer Science and Engineering DUET, Dhaka	Member (External) Board of Examination
Lt Col Muhammad Nazrul Islam Associate Professor of Computer Science and Engineering MIST, Dhaka	Member Examination
Brig Gen Md Towhidul Islam Senior Instructor of Computer Science and Engineering MIST, Dhaka	Head of the Department Member (Ex-officio)

Department of Computer Science and Engineering, MIST, Dhaka.

IDENTIFICATION OF DDOS ATTACK THROUGH INTRUSION DETECTION MODEL USING ENSEMBLE MACHINE LEARNING

DECLARATION

I therefore declare that this thesis is my unique work and authored entirely by myself. I have properly credited all sources of material used in the thesis. This thesis has never been presented for a degree or diploma at any university or institute before (in whole or in part). All sources and support obtained in preparing this thesis have been acknowledged and/or cited in the reference section.

Adeeba Anis

Department of Computer Science and Engineering, MIST, Dhaka.

IDENTIFICATION OF DDOS ATTACK THROUGH INTRUSION DETECTION MODEL USING ENSEMBLE MACHINE LEARNING

ABSTRACT

A distributed denial of service (DDoS) attack targets at hindering authorized individuals from accessing a server or website by flooding it with traffic from many sources. To avoid a DDoS attack from damaging the target system, detection is required. The system becomes unsafe as a result of this attack. The aim of this thesis work is to provide an ensemble machine learning technique to detect DDoS attack. Another objective is to select optimal features of the dataset. In this thesis dataset is collected from Kaggle repository which contains 42 columns and 17171 rows. Firstly, three feature selection techniques—ANOVA, Mutual Information, and Feature Importance have been used to reduce the dataset and increase the performance. Then, optimal features have been selected using domain knowledge. The traditional machine learning methods K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree (DT), and Naive Bayes (NB) are then used with the chosen features. Next, five ensemble models have been created by all the combinations of four traditional models - (KNN, SVM, DT), (KNN, SVM, NB), (KNN, NB, DT), (SVM, NB, DT) and (KNN, SVM, NB, DT). By evaluating accuracy, precision, recall, and F1-score, the experiment's outcome is determined. After all the experiments, the result shows that the ensemble voting classifier by the combinations of KNN, SVM and DT gives the highest accuracy. Among the feature selection techniques, feature importance technique gives the maximum accuracy that is 98.86% and by using the optimal features, highest accuracy to detect the DDoS attack is determined which is 99.4%.

ACKNOWLEDGEMENTS

First and foremost, I would like to state unequivocally that ALLAH is the source of all evaluations. He has given me the ability to carry on with and finish this research work. I thank Allah profusely for all of His blessings.

I would like to thank my supervisor, Dr. Md. Shohrab Hossain, Professor, Department of Computer Science and Engineering (CSE), Bangladesh University of Engineering and Technology (BUET), for the guidance, encouragement, and advice he has provided me throughout my time as his student. I have been extremely lucky to have a supervisor who cared much about my work, and who responded to my questions and queries so promptly. In addition to being an admirable supervisor, he is a man of principles and has immense knowledge of research in general and his subject in particular. My heartfelt gratitude goes to him for his patience in evaluating so many poor drafts, offering fresh approaches, pointing me in the right direction and encouraging me to keep working. I am extremely grateful to my Co-Supervisor, Tasmiah Tamzid Anannya for her support during my thesis work.I would also like to thank the members of my thesis examination board, Brig Gen Md Towhidul Islam, Lt Col Muhammad Nazrul Islam, Professor Dr. Fazlul Hasan Siddiqui for their encouragement and insightful comments.

Most importantly, none of this would have been possible without the sacrifice and patience of my husband. I would like to express my heartfelt gratitude to my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

An intentional attempt to block a computer network or website or server from functioning correctly by overflowing it with incoming data is known as a DDoS attack (Savita and Sharma, 2023). In simple terms, it impedes genuine users' access to the network or website by acting as a virtual traffic jam that clogs it up. Usually, the attacker forms a "botnet" by coordinating a massive number of computer or devices to concurrently send a massive amount of requests or data to the target network or website. The server can no longer handle the volume of traffic, which results in a major slowdown or server breakdown. This attack is frequently used as a form of extortion, threats, or mischief (K. Kumar and Barver, 2021).

1.1 Overview

DDoS attacks involve a number of drawbacks for both the individuals or organizations who are being attacked. The main objective of a DDoS attack is to disable or disrupt the targeted network or website. As it causes delay and lost sales, this can have serious consequences for businesses. Individuals suffer because they cannot access the websites or internet services they require. As a result of disruption of services, the business faces financial loss and reputational harm.

Now, Intruders have additional opportunity to launch malicious attacks as data is generated from many sources, so it is necessary to defend servers against them (Samat, 2022). Since the attacker's traffic could be difficult to distinguish from legitimate traffic, DDoS attacks can be challenging to detect and mitigate. Among the techniques and technologies that may be used to identify and lessen DDoS attacks include network traffic analysis, anomaly detection, rate restriction, blacklisting, and cloud-based DDoS defense.

To sum up, preventing an attack using DDoS is essential to maintaining service avail-

ability, a positive user experience, minimizing monetary losses, safeguarding reputation, preventing secondary attacks, following the legal requirements, and upholding cybersecurity best practices.

Numerous works have been done on the systems related to intrusion detection. By using DT and NB machine learning techniques on the CAIDA's Dataset, Tuan et al. were able to detect DDoS attacks by using SVM, NB, DT, ANN and K-means (Tuan et al., 2020). Polat et al. detected DDoS attack on SDN (Software Defined Network) dataset (Polat et al., 2020). On UNSW-NB 15, Azmi et al. used ANN, NB, and DT to detect a DDoS attack (Azmi et al., 2021). Beulah et al. detected DDoS attack by using ensemble voting technique for SVM and LR on KDD CUP dataset (Beulah and Pitchai Manickam, 2022). However, (Tuan et al., 2020), (Polat et al., 2020) and (Azmi et al., 2021) didn't use any ensemble machine learning technique while (Beulah and Pitchai Manickam, 2022) used ensemble voting machine learning for two machine learning algorithms, SVM and LR and (Das et al., 2019) used ensemble machine learning combining four - NN, SVM, KNN and DT-C4.5. Our work *differs* from the previous works in a way that we have used ensemble voting machine learning technique for three and four machine learning algorithms. We use ensemble machine learning technique for all the combinations of KNN, SVM, DT and NB.

1.2 Motivation

DDoS attack is one of the most frustrating and difficult criminal activity to prevent. Even the largest sites could be taken down easily through this attack by overflowing servers with requests far more than they can serve. When the servers are incapable of serving these dummy requests, they crash and most of the time requires a long time to restore.

"One of the most powerful weapons on the internet" is what Norton calls the DDoS attacks. Denial-of-service attacks have the potential to occur at any time, affect any aspect of a website's functionality or resources, and cause a great deal of disruption to service along with significant financial losses. Data indicates that DDoS attacks, which were formerly a source of amusement, are now being used by hackers as a means of generating

revenue or causing damage (Cook, 2023).

From Q2 to Q3 of 2022, there had been a 10% decline in the number of application layer DDoS attacks, according to Cloudflare (Yoachimik, 2022). In Q2 of 2020, a huge and long-lasting spike in the number of attacks has been witnessed when coronavirus everyone to go online (Kaspersky Lab, 2020). As many people are still working from home, there is still a net increase in the number of attacks.

According to Cloudflare, the third quarter of 2022 witnessed an increase by 67% year-on-year and 24% quarter-on-quarter. Again, online industries reported a record 131% increase quarter-on-quarter rise in the number of attacks in the application layer.

2022 saw a higher amount of DDoS activity than the previous years. Furthermore, the attacks are lasting longer than ever before. In Q2 of 2021, where the average duration of DDoS attack was 30 minutes, in the Q2 of 2022, the average duration was 50 hours.

When Amazon reported their success against the largest DDoS attack ever recorded, the searches for "ddos" spiked in June 2020 (BBC News, 2020). However, in September 2022, Google claimed to have stopped a DDoS attack of 46 million requests per second.

Amazon disclosed that it had to fight off a 2.3 Tbps DDoS attack in the first quarter of 2020 ("AWS Shield Threat Landscape Report – Q1 2020", 2020). This holds significance for multiple reasons. Firstly, it's the biggest known attack in history, surpassing the previous record-holder's throughput by nearly four times (587 GB/s).

Notable is also the fact that attacks exceeding 100 GB/s are still increasing, despite a remarkable 967 percent increase in 2019 over 2018. Attacks between 50 and 100 GB/s also rose 567 percent in the same year. Fig. 1.1 shows the amount of data in DDoS attacks.

Our proposed ensemble model is able to classify ddos attacks based on the top 10 or 11 features. Thus, if it is possible to get values of the same features in real time, it would be possible to predict whether there is a possibility of a ddos attack by assessing those

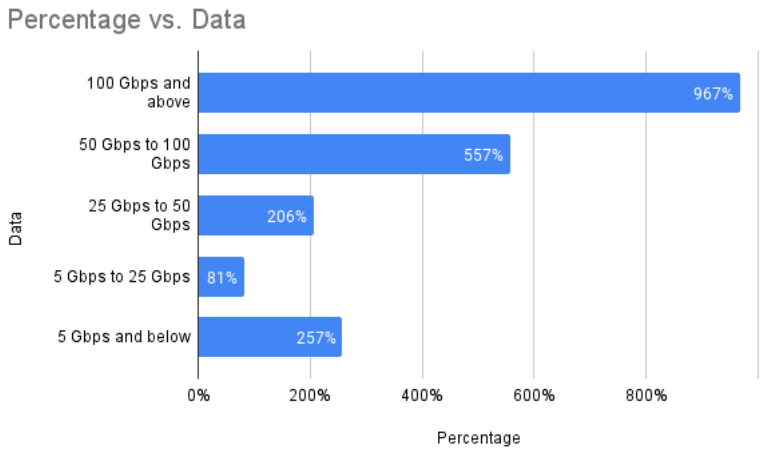

Figure 1.1: Amount of Data in DDoS Attacks

features which can prevent an attack. E.g. if the system observes flag and src_bytes in real time, then it would be seeking 0 for both the attributes. If it sees 0 somehow, it can predict that there might be a ddos attack incoming.

1.3 Objective

The latest trends in machine learning shows that ensemble methods outperform traditional machine learning models' solo performances. Thus, the objective of this work is to implement ensemble models for the detection of DDoS attacks in devices. The following goals have been taken into consideration in order to achieve the objective:

i. To identify the optimal features of DDoS attacks to detect the attack effectively.

ii. To propose a modified intrusion detection model to predict DDoS attacks based on the detected optimal features.

iii. To implement the proposed intrusion detection model and compare the performance with the existing machine learning algorithms.

1.4 Contributions

The fields of machine learning and DDoS attack detection have benefited from this study. Several advancements have been made in the field of general machine learning, even though the application of machine learning to DDoS detection is the main focus of this

thesis.

- All the features on a dataset are not equally important for training and testing a classifier. A new set of features has been provided using domain knowledge to improve the accuracy of DDoS attack detection.
- The combination of base classifiers had not been considered previously in the literature. Ensemble method has been applied on the traditional machine learning models (KNN, SVM, NB & DT) and five ensemble models have been implemented using the four traditional models which successfully offers better performance.

1.5 Organization of The Thesis

The remaining of this thesis is organized as follows. Chapter 2 provides an introduction to the domain of machine learning, containing its applications, different types, traditional models, ensemble learning methods and their kinds followed by literature review, which is followed by the overall gap analysis in section 2.3. Chapter 3 titled proposed approach starts with description of the dataset used in this work followed by different kinds of filter methods. Lastly, the methodology of this thesis has been discussed. The dataset utilized for this research is covered in Chapter 4, along with the methodology. The algorithms' experimental findings are shown in Chapter 5. The algorithm's performance was assessed using separate training and testing data, and it was compared with other classifiers that were already in use. The thesis is concluded and recommendations for more research are provided in Chapter 6.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

In this era, Machine Learning (ML) is very popular. But, previous version of machine learning and the latest versions are not the same. Previously, it was not programmed, but now Artificial Intelligence (AI) can act like human brains. By ensemble machine learning technique, various traditional ML algorithms can be combined

2.1 Machine Learning

A subfield of AI is ML, which focuses on learning from huge amount of data without the intervention of humans ("Machine Learning (ML)", 2023). The "Training" phase in a machine learning project is to train algorithms for discovering relationships and patterns in data (Başţanlar and Özuysal, 2014). This recognition of pattern could be used in prediction of data, classification of information, creating cluster of data points and even generate new content such as text to text, text to speech, caption to image etc. using applications like ChatGPT, Dall–E 2.

2.1.1 Applications of Machine Learning

Machine learning is used in many different kinds of sectors. These include speech recognition, image identification, product suggestions, traffic prediction, self-driving cars, malware and spam email filtering, virtual assistants, online fraud detection, stock market trading, and more ("Applications of Machine Learning", 2023). Fig. 2.1 shows some of the applications of ML.

Image recognition is one of the most widely used applications of machine learning. It is employed to identify things, people, locations, images, etc. One common application of face detection and image recognition is automatic friend tagging suggestion on Facebook.

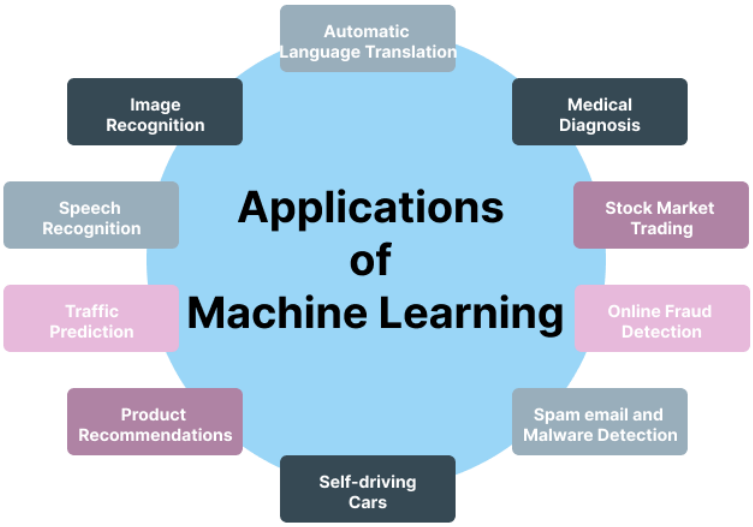

Figure 2.1: Applications of Machine Learning

The option “Search by voice” we get while searching on YouTube or Google is an example of speech recognition, as in this case machine learning is used to recognize human speech and convert the speech into text.

Machine learning is the backbone of recommendation systems, which can be seen in web applications like e-commerce or social media and news organization. The recommendation system in these websites collect data from users first, and then it figures out the necessary pattern from the data to recommend users what they might love, or they might need.

Self-driving cars also use machine learning to get vision, which is also known as computer vision, to navigate safely throughout the roads. Deep Learning (DL), which is machine learning’s subfield, has algorithms such as YOLO for doing this type of job.

In healthcare, machine learning plays a vital role by detecting different types of diseases in less time and more accurately compared to traditional approaches.

The usage of machine learning in spam email and malware detection can be seen in email services, where a new received email is categorized into spam or ham (not spam) based on the email subject.

2.1.2 Types of Machine Learning

Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four categories into which machine learning can be split.

The kind of algorithm that data scientists select is determined on the type of data. Many of the methods and algorithms are not exclusive to any one of the main machine learning types mentioned above. Depending on the data collection and the problem to be solved, they are frequently modified to fit several categories. For instance, convolutional neural networks and recurrent neural networks are examples of deep learning algorithms that are used in supervised, unsupervised, and reinforcement learning tasks, depending on the specific problem and data availability.

2.1.2.1 Supervised Learning

In supervised learning, labelled datasets are used to train models, allowing them to learn about all kinds of data ("Supervised Machine Learning", 2023). The output is forecasted by the model after it has been trained using test data, a subset of the training set.

Let's say there is a dataset including a variety of forms, such as triangles, rectangles, squares, and polygons. Now the model must be trained for every shape as the first step.

- A form is going to be classified as a square if it has four equal sides.
- A form is going to be classified as a triangle if it has three sides.
- A form is going to be classified as a hexagon when it has six equal sides.

After training, the test data is used to evaluate the model. The model's job is to recognize the shape. The model has already been educated on a wide variety of shapes. When it encounters a new shape, it categorizes it based on several sides and forecasts the result. Supervised learning is shown in Fig. 2.2
Supervised learning can be divided into regression and classification as shown in Fig. 2.3. Regression algorithms are used when the output is continuous. For instance, car price,

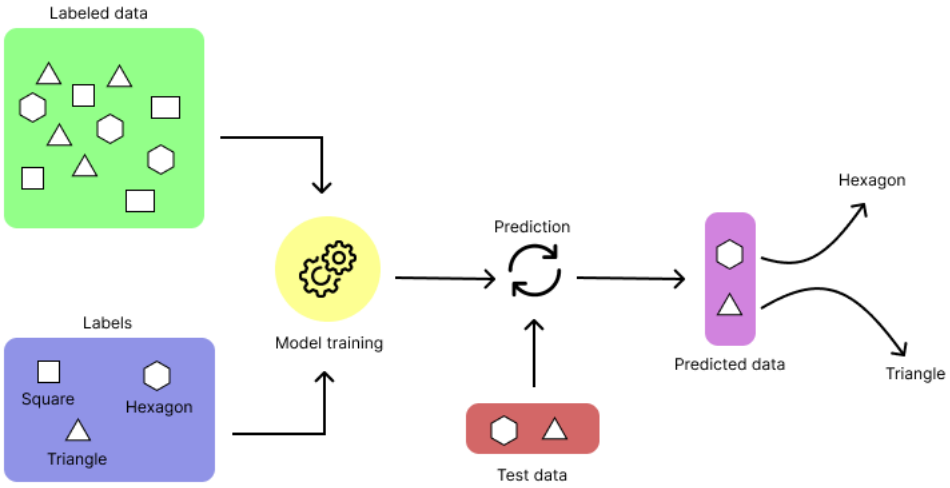

Figure 2.2: Supervised Learning

temperature or ice cream sales etc. Some of the regression algorithms are linear regression, non-linear regression, polynomial regression etc.

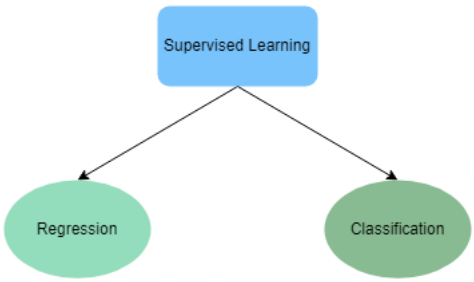

Figure 2.3: Types of Supervised Learning

Classification is when a machine learning model tries to output variables such as true-false, yes-no or distinct integer values. For example, when a model classifies cat or dogs, sunny or cloudy images. Some algorithms used for classification are SVM, decision tree, random forest etc.

2.1.2.2 Unsupervised Learning

ML without supervision is applied to data that lacks previous labels. The system is not provided the ”right answer.” It is up to the algorithm to determine what is being displayed. Finding organization in the data through exploration is the aim. With transactional data, unsupervised learning performs admirably.

Unlike supervised learning, where the input data is known but no matching output data, unsupervised learning cannot be immediately applied to a regression or classification task ("Unsupervised Machine Learning", 2023). Unsupervised learning aims to discover the underlying structure of a dataset, classify the data based on similarities, and display the dataset in a compact format.

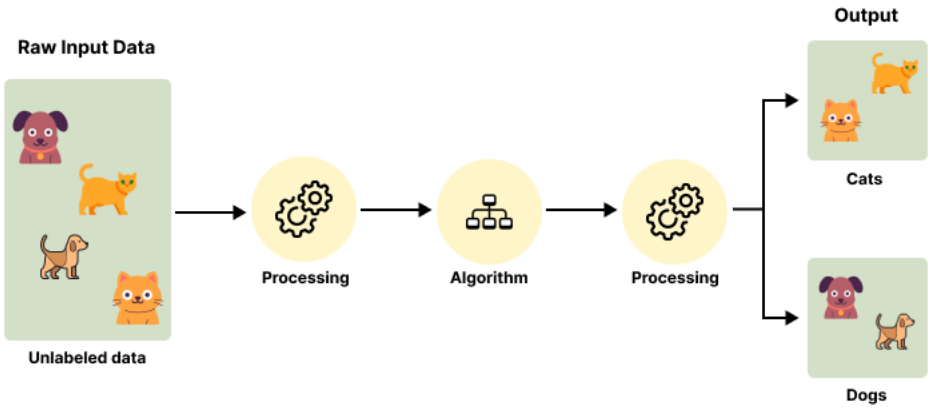

Figure 2.4: Unsupervised Learning

For instance, an input dataset comprising photographs of various breeds of dogs and cats is provided to the unsupervised learning algorithm in Fig. 2.4. The algorithm has no knowledge of the dataset's characteristics because it has never been trained on the provided dataset. The objective of the unsupervised learning algorithm is to allow the picture features do the talking. An unsupervised learning technique that groups the image collection according to image similarity will be used to finish this task.

The unlabeled images will be fed to a machine learning model first. Later, the model is going to interpret the data to find the hidden patterns and then the model is going to find an algorithm suitable for the task. Algorithms such as K-means clustering, decision tree etc. are some of the suitable choices. After the appropriate algorithm is applied, the data object is divided into clusters according to their similarities or differences.

There are two types of unsupervised learning, which are clustering and association, which is shown in Fig. 2.5. Using a technique called clustering, items are grouped into clusters so that those with the greatest similarities stay in one group and have little to no similar-

ities with those in other groups. By identifying patterns among the data objects, cluster analysis groups them into groups based on whether such patterns exist.

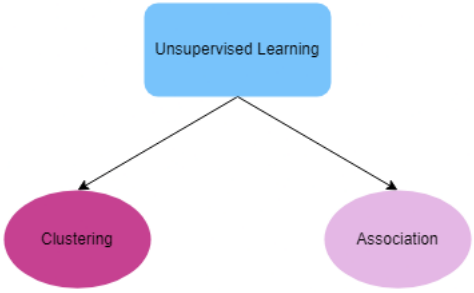

Figure 2.5: Types of Unsupervised Learning

To determine the links between variables in a huge database, an unsupervised learning technique called an association rule is employed. It establishes the group of objects that appear collectively in the dataset. Marketing strategy is enhanced by the association rule. This could be stated like when people buy X they tend to buy Y as well. For example, consumers who purchase bread also frequently buy butter or jam. Market Basket Analysis is an example of an association rule in action.

2.1.2.3 Semi Supervised Learning

Another type of ML is semi-supervised learning, shown in Fig. 2.6. It is situated in the middle of supervised and unsupervised learning ("Semi-Supervised Learning in Machine Learning", 2023). In case of this type, a model is trained using a small amount of labeled data and a large amount of unlabeled data is used. The goal of semi-supervised learning is to create a function that can accurately predict the output variable from the input variables, much like supervised learning does. Unlike supervised learning, the method is trained on a dataset that contains both labeled and unlabeled data.

Semi-supervised learning can be useful when there is a large amount of available unlabeled data, but it would be costly or difficult to label it all.

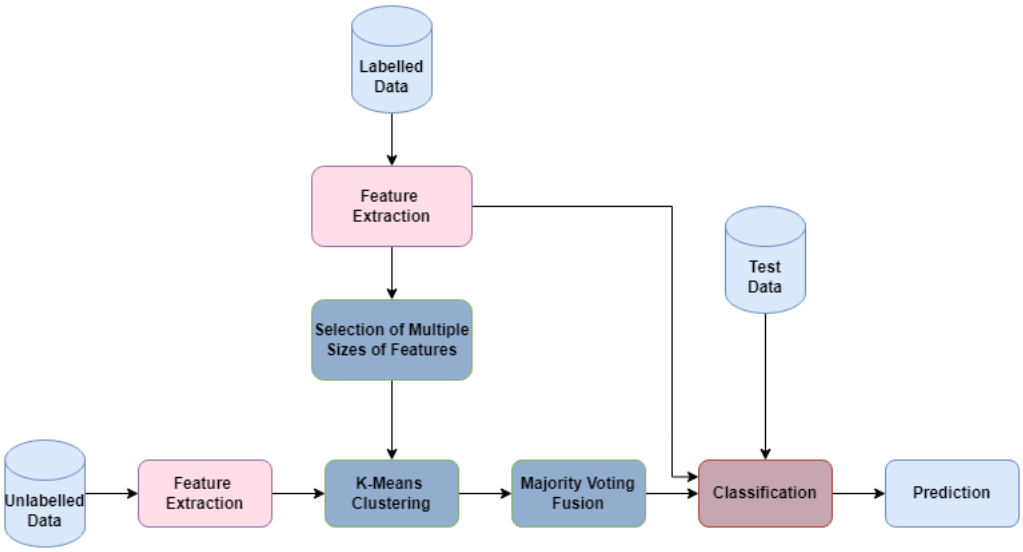

Figure 2.6: Semi Supervised Learning

Semi supervised learning can be applied in text classification, where a small amount of labeled and a large amount of unlabeled data is trained to classify a text into one or more groups. The same can be done in image classification and anomaly detection.

2.1.2.4 Reinforcement Learning

Reinforcement learning is ensuring that a model takes a suitable action in a particular situation that maximizes rewards ("What is Reinforcement Learning?", 2023). It is utilized by a variety of software and devices to determine the optimal course of action or behavior for a given circumstance. In contrast to supervised learning, which involves training the model with the correct answer pre-existing in the training set, reinforcement learning involves training the model in the absence of an answer and using the reinforcement agent's judgment to determine how to accomplish the given task. Without a training dataset, it will inevitably pick up knowledge from its experiences.

The algorithms used in reinforcement learning learn from outcomes and then these decide the next action. After an action is taken, the algorithms receive feedbacks which help these decide whether the taken was right or wrong. It's a useful method for automated systems that need to make numerous minor judgments on their own without human assistance.

A self-governing, self-teaching system, reinforcement learning basically learns by making mistakes. It learns by doing in order to attain the best results, or, to put it another way, it takes actions with the intention of maximizing rewards.

2.1.3 Base Level Classifier

There are some base level machine learning classifiers which can be used to create ensemble machine learning models.

2.1.3.1 KNN

KNN which stands for K Nearest Neighbors, is a Machine Learning model used for both classification and regression tasks based on supervised learning (Srivastava, 2018). The idea behind the machine learning algorithm is that similar things tend to stay around each other. The whole dataset is stored as a reference during KNN training. When classifying an input data point, the model uses a distance metric to decide its output. As the model does not learn from the dataset during its training, and it uses the reference of the dataset later, the model is also known as lazy learner algorithm "K-Nearest Neighbor Algorithm for Machine Learning", Year not specified. Fig. 2.7 shows training of KNN.

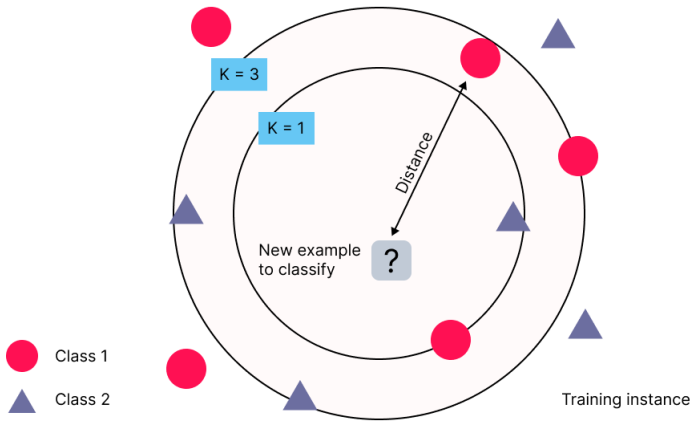

Figure 2.7: Training of KNN model

As a classification model, the model identifies K neighbors that are nearest to the input data point using metrics such as Euclidean or Manhattan distance. When giving output,

the model finds out the most common class label among neighbors and classify the input data as that label. However, in case of regression, the model also known as KNN Regression, calculator the average of output feature of the K nearest neighbors.

2.1.3.2 SVM

One of the most widely used supervised learning techniques for both classification and regression issues is support vector machine, or SVM. But it's mostly applied to machine learning classification challenges ("Machine Learning - Support Vector Machine Algorithm", 2023).

Decision boundaries called hyperplanes are used to help in data point classification (Gandhi, 2018). The classes of the data points that lie on either side of the hyperplane are distinct. The number of features also affects the hyperplane's dimension. The hyperplane is essentially a line if there are just two input features. The hyperplane transforms into a two-dimensional plane if there are three input features. Imagination becomes challenging when there are more than three features.

In order to create a hyperplane, SVM selects the extreme points or vectors. The technique known as the Support Vector Machine is named after these extreme situations, which are referred to as support vectors. The Fig. 2.8 could be examined, which uses a decision boundary or hyperplane to classify two distinct categories:

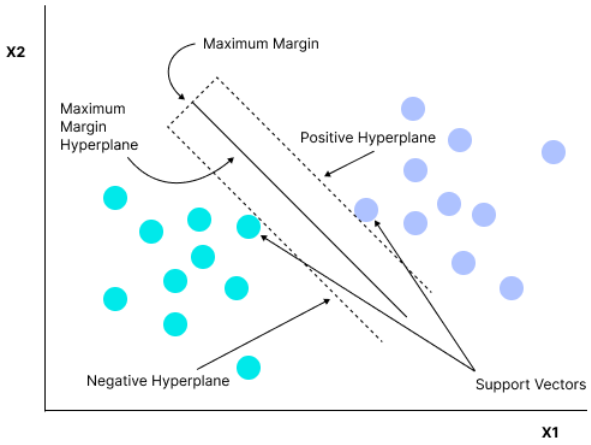

Figure 2.8: Training of SVM model

The hyperplane that shows the biggest margin or separation between the two classes is an appropriate candidate for best hyperplane (“Support Vector Machine Algorithm”, 2023). Fig. 2.9 shows the process of choosing the hyperplane.

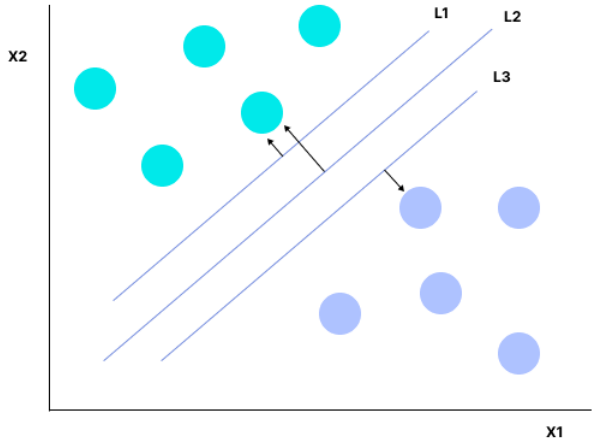

Figure 2.9: Training of SVM model

Thus, the hyperplane with the greatest distance between it and the closest data point on each side is chosen. If there is any hyperplane that satisfies the above condition, that is known as the maximum-margin hyperplane or hard margin.

There are two types of SVM. The first one is linear SVM, which is implemented when dealing with linearly separable data. A dataset is referred to as non-linear data and the classifier employed is referred to as a non-linear SVM classifier if it cannot be classified using a straight line.

The other type is non-linear SVM. When a dataset cannot be classified using a straight line, it is referred to as non-linear data, and the classifier employed is known as a non-linear SVM classifier. Thus, non-Linear SVM is used for non-linearly separated data.

2.1.3.3 Decision Tree

A decision tree is a supervised machine learning algorithm that can be used for both classification and regression (“Decision Tree: Introduction and Example”, 2023) (“Machine Learning - Decision Tree Classification Algorithm”, 2023). With a root node, branches,

internal nodes, and leaf nodes, it has a hierarchical tree structure. Decision trees offer simple-to-understand models and are utilized in regression and classification applications.

Depicting decisions and their possible outcomes, including chance events, resource costs, and utility, a decision tree is a hierarchical model used in decision support systems. The tool can be used in a variety of contexts. Problems regarding regression and classification can both be solved with decision trees. The term itself implies that it displays the predictions that arise from a sequence of feature-based splits using a flowchart akin to a tree structure. A decision made by the leaves marks the end of it, which begins with a root node, as shown in Fig. 2.10.

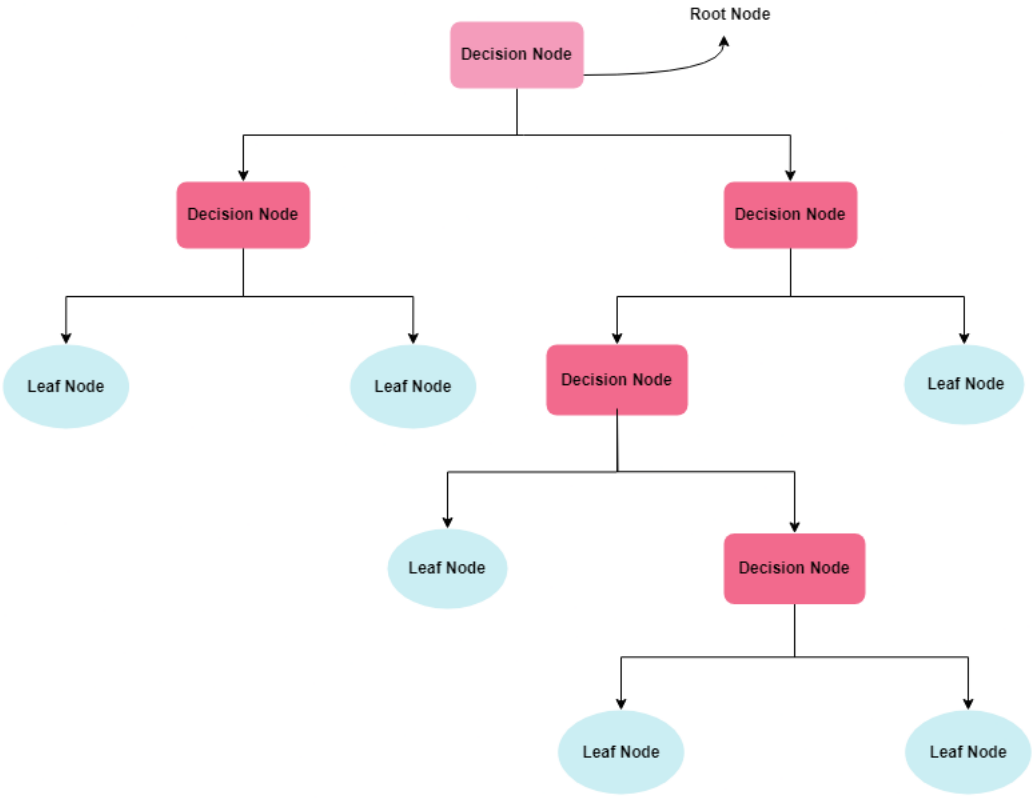

Figure 2.10: Training of DT model

In order to forecast the classification of the data set, the decision tree analyzes it ("Decision Tree: Introduction and Example", 2023). The process starts at the root node of the tree, where it compares the value of the root attribute to the attribute of the record in the real data set. It moves on to the next node, following the branch, based on the comparison. By comparing the attribute values of each succeeding node with those of the sub-nodes,

the algorithm continues this activity and moves on to the next node. It keeps going until it gets to the tree's leaf node. The algorithm provided below provides a better explanation of the entire operation.

- Step 1: Root node (containing the entire dataset) should be the starting point of the tree, according to S.
- Step 2: Use the Attribute Selection Measure (ASM) to determine which attribute in the dataset is the greatest.
- Step 3: Divide the S into subsets that include potential values for the best qualities in step three.
- Step 4: Create the node in the decision tree that has the best attribute.
- Step 5: Utilizing the subsets of the dataset generated in Step 3, recursively design new decision trees. Classification and Regression Tree algorithm - Once we reach a point where we are unable to categorize the nodes any further, call the final node a leaf node.

2.1.3.4 Naive Bayes

The Naïve Bayes algorithm is a supervised learning technique that solves classification issues ("Machine Learning - Naive Bayes Classifier", 2023). It is based on the Bayes theorem. Its primary application is in text categorization, where a high-dimensional training dataset is used. One of the most straightforward and efficient classification algorithms, the Naïve Bayes classifier, aids in the rapid development of machine learning models with rapid prediction capabilities. Being a probabilistic classifier, it makes predictions based on the likelihood that an object will occur.

Among the most well-known applications of the Naïve Bayes algorithm include article classification, sentiment analysis, and spam filtering. It is referred to as naïve because it makes the assumption that the existence of one characteristic is unrelated to the existence of other traits. For example, if the fruit is classified based on its color, shape, and flavor, then red, sweet, spherical fruit is identified as an apple. Therefore, without relying on one

another, each feature alone helps to identify that it is an apple. Because it is based on the Bayes' Theorem, it is known as Bayes. Fig. 2.11 shows the training of NB.

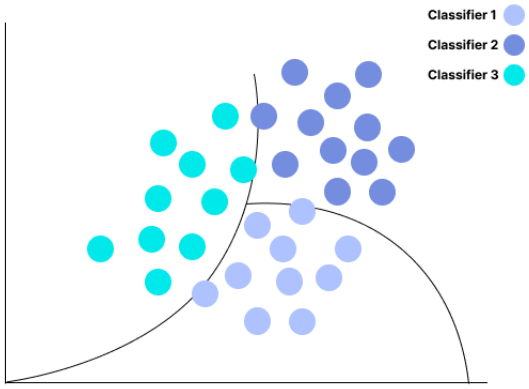

Figure 2.11: Training of NB model

2.1.4 Ensemble Learning Methods

Since an ensemble can be trained and then utilized to generate predictions, it is essentially a supervised learning algorithm in and of itself. Ensemble learning techniques aim to increase the accuracy of the predictive models in order to improve their performance. The method of strategically building several machine learning models (such classifiers) to address a specific issue is called ensemble learning.

The technique of creating an ensemble involves bringing together a varied group of learners, or individual models, to improvise on the model's stability and predictive capacity. Ensemble learning refers to the process of combining all of the predictions made in the example above.

In ensemble machine learning technique, various traditional machine learning algorithms are applied first to evaluate the model. After that, multiple models are combined on the basis of simple average or weighted average (Akhtar and Feng, 2022).

In the simple average technique, the mean value is obtained from different machine learning algorithms and then combined to get the ensemble voting value. In weighted average method, as shown in Eq. (2.1-2.2), the arithmetic mean value is obtained from different

machine learning algorithm given different weight according to their accuracy (Solano and Affonso, 2023).

$$\hat{y} = \sum_{j=1}^{m} \frac{\hat{y}_j}{m} \tag{2.1}$$

$$\hat{y} = \frac{\sum_{j=1}^{m} (w_j \hat{y}_j)}{\sum_{j=1}^{m} w_j} \tag{2.2}$$

2.1.5 Types of Ensemble Learning Methods

Many ensemble learning methods have been invented till now. However, the following ones are the most used nowadays.

2.1.5.1 Bagging Method

Bagging is a technique used in Ensemble formation, which is also referred to as Bootstrap Aggregation. Bootstrap creates the framework for the bagging method. Using the bootstrap sampling technique, "M" observations from a "N" observations in the population is taken. However, every choice is made at random, meaning that each from the initial population, an observation can be selected so that every observation is to have an equal chance of being chosen during each bootstrapping cycle.

2.1.5.2 Voting Method

One of the simplest ensemble learning strategies is voting, which combines predictions from several models. The process begins with using the same dataset to create two or more distinct models. After that, the earlier models can be wrapped and their predictions combined using an ensemble model that is based on voting. The Voting based Ensemble model can be used to forecast new data once it has been built. One can apply weights to the sub-models' predictions. One technique that can be used to figure out how to appropriately weigh these predictions is stacked aggregation.

There are two types of voting methods. One is hard voting and the other is soft voting (Atif et al., 2022). In hard voting approach the majority of the prediction by the classifiers is calculated while in soft voting the probability prediction of each classifier is combined to calculate the result (Karim et al., 2023).

Hard voting, sometimes referred to as majority voting, is the straightforward process of adding up each base model's predictions and designating the class with the highest number of votes as the final forecast. It works well in classification jobs where the classes are distinct and incompatible with one another.

Weighted voting, or soft voting, considers the probability scores of each base model for every class and determines the final forecast by averaging these probabilities. After adding up all the predictions, soft voting determines the winner with the highest weighted probability by averaging the probabilities for each class. It works well for jobs involving both regression and classification.

2.1.5.3 Boosting Method

Boosting is implemented by training models sequentially, where each model tries to support the other one in their classification task. The way this whole process works is, at first, all the instances in a training dataset are provided some weight which are equal initially. Later, some random instances are chosen from the training dataset to create a sub-dataset so that it could be used to train the first model. The model will be asked to classify the whole training dataset after training. The instances that the model incorrectly classifies will be given a higher weight, and those instances will get higher priority in case of being chosen for the next sub-dataset for training the second model. This is how the process continues till we are satisfied with the number of models. Finally, test data is given to all the models for prediction and majority voting is used to pick the prediction for the strong model or final model.

2.2 Literature Review

DDoS attacks are one of the most significant dangers in the field of information technology that must be identified and tackled. The section is related to the detection of DDoS attack using various feature selection techniques used with various machine learning and deep learning algorithms on various datasets.

Several Intrusion Detection Systems have been deployed using machine learning and deep learning algorithms. The capacity of a system to learn from a specific training dataset and the procedure of performing analysis on the given dataset to solve related tasks is known as Machine learning (Janiesch et al., 2021). Deep learning is a part of machine learning which is based on Artificial Neural Network (ANN) that uses numerous layers to analyze data and, in some situations, outperforms shallow machine learning algorithms in terms of accuracy. Some of the deep learning algorithms are Deep Brief Networks (DBNs), Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) (Liu and Lang, 2019). Again, SVM, KNN, NB, LR and DT are some machine learning algorithms.

Hailye Tekleselassie proposed a deep learning-based method for detecting DDoS attacks (Tekleselassie, 2021). A combined approach of Convolutional Neural Network (CNN) with Stacked Auto Encoder (SAE) is offered for DDoS detection using the CICIDS2017 dataset. Here, the TP rate, FP rate, Precision, Recall, F-measure, and accuracy of SMO, Bayes net, and RF have been compared. The study demonstrates that RF performs better in terms of all metrics.

A DNN has been proposed by Bhardwaj et al. to classify the network into normal and DDoS attack (Bhardwaj et al., 2020). To select features, stacked AutoEncoder (AE) is used. On two distinct datasets, NSL-KDD and CICIDS2017, the proposed optimized AE with DNN is compared with several state-of-the-art techniques in terms of performance measures, including detection accuracy, precision, recall, and F1-Score.

Then, many machine learning-based techniques have been considered. Azmi et al. used the Data Reduction and Information Gain approach to select features from the UNSW-NB

15 dataset (Azmi et al., 2021). After selecting the specific features, Accuracy, Precision, True Positive and False Positive have been measured using different algorithms such as ANN, NB, and DT. The accuracy obtained by the selected features comes out to be better than the accuracy obtained by the original dataset.

Feature selection methods are used, in order to get a smaller subset of input features and increase accuracy with a smaller number of features. Araujo et al. have measured the impact of feature selection techniques by XGBoost algorithm on a public dataset to detect DDoS attack classification (de Araujo et al., 2021). Moreover, accuracy, precision, recall, and F1 score metrics have been compared using various algorithms which include Mutual Information, ANOVA, RFE, XGBoost Gain and Ensemble methods and the outcome demonstrates that ANOVA is the most effective method for this dataset.

Polat et al. used a variety of machine learning methods to identify DDoS attacks in Software Defined Networks (SDN) (Polat et al., 2020). At first, filter based, wrapper based and embedded based feature selection techniques have been applied to select features from the experimental SDN topology dataset. Afterward, Accuracy, Sensitivity, Specificity, Precision, and F1-Score are measured using KNN, SVM, NB, and ANN algorithms. After completing the experiment, it is found that KNN gives the highest accuracy (98.3%) with wrapper feature selection technique.

Tuan et al. evaluated the performance using the Accuracy, Sensitivity, Specificity, False Alarm Rate (FAR), False positive rate (FPR), AUC, and Matthews Correlation Coefficient (MCC) on the UNBS-NB 15 and KDD99 Dataset (Tuan et al., 2020). In this experiment, Supervised Machine Learning Algorithms- DT, SVM, NB, ANN and Unsupervised Machine Learning Algorithms- K-means, X-means have been used to evaluate the performance.

Beulah et al. proposed an ensemble method by combining SVM and LR to detect DDoS attack on KDD CUP dataset (Beulah and Pitchai Manickam, 2022). In this work, voting classifier is used for the ensemble method. The overall performance of the proposed work has a high accuracy and low false positive. The accuracy of the proposed work is 99.2%.

Kumar and Kamatchi proposed an ensemble machine learning technique to detect anomaly based intrusions (Y. V. Kumar and Kamatchi, 2020). The authors used NSL-KDD dataset to apply algorithms like DT, Bayes classifier, RNN-LSTM and RF. In this work, ensemble voting is used to increase the overall performance compared with the existing algorithms. Table 2.1 offers a summary of the related works including dataset, feature selection technique, classifier, number of features and metrics.

Saikat et al. proposed an ensemble method by combining NN, SVM, KNN and DT-C4.5 to detect DDoS attack on NSL-KDD dataset (Das et al., 2019). In this work the performances were evaluated using Accuracy, TPR, FPR, Precision, Recall, F-Measure, and ROC curve. The proposed work gives low false positive and high accuracy rate. The accuracy of the proposed work is 99.1%.

Table 2.1: Related Work

Reference	**Dataset**	**Feature Selection Technique**	**Classifier**	**No. of Features**	**Metrics**
Bhardwaj et al., 2020	CICIDS-2017 NSL-KDD	Auto Encoder	DNN	23 40	accuracy precision recall F1-Score
Azmi et al., 2021	UNSWNB 15	Information Gain Data Reduction	ANN NB DT	10	accuracy precision True positive rate False positive rate

de Araujo et al., 2021	CICDDoS 2019	ANOVA MI XGBoost Gain RFE	Binary Multiclass	80	accuracy precision recall F1-Score
Polat et al., 2020	SDN	Filter based Wrapper based Embedded based	KNN SVM NB ANN	12	Accuracy Sensitivity Specificity Precision F1-Score
Tekleselassie, 2021	CICIDS 2017	Autoencoder RBM	SMO Bayes net RF	21	TP rate FP rate Precision Recall F-measure Accuracy
Das et al., 2019	NSL-KDD	Domain Knowledge	Ensemble (NN, SVM, KNN, DT-C4.5)	24	Accuracy Precision Recall F-measure TP rate FP rate ROC area

Tuan et al., 2020	UNBSNB 15 KDD99	Information Gain	SVM ANN NB DT K-means	10 9	accuracy False Alarm Rate Sensitivity specificity False positive rate AUC
Beulah and Pitchai Manickam, 2022	KDD CUP	Domain Knowledge	Ensemble (SVM, LR)	41	Accuracy True positive rate False positive rate
Y. V. Kumar and Kamatchi, 2020	NSL-KDD	ANOVA	DT, RF, KNN SVM DNN Adaboost	27	accuracy precision recall F-score time

This paper.,2023	DDoS classification	ANOVA Mutual Information Feature Importance	KNN, SVM, DT, NB Ensemble (KNN, SVM, DT) Ensemble (KNN, SVM, NB) Ensemble (SVM, DT, NB) Ensemble (KNN, DT, NB)	10 11	accuracy precision recall F1-score

2.3 Gap Analysis

Azmi et al. worked on UNSWNB 15 dataset to detect DDoS attack on 10 features using ANN, NB and DT (Azmi et al., 2021). Araujo et al. used CICDDoS 2019 dataset with a huge number of features that is 80 using XGBoost (de Araujo et al., 2021). Polat et al. worked on SDN to detect DDoS attack using KNN, SVM, NB, ANN with 12 features (Polat et al., 2020). Kumar et al. used NSL KDD dataset using DT, RF, KNN, SVM, DNN and Adaboost with 27 features (Y. V. Kumar and Kamatchi, 2020). Saikat et al. (Das et al., 2019) worked on NSL KDD dataset to detect DDoS attack by ensemble method that is combined with NN, SVM, KNN and DT-C4.5. Tuan et al. worked on UNBSNB 15 and KDD 99 dataset SVM, ANN, NB, DT and K-means with 9 and 10 number of features (Tuan et al., 2020). Beulah et al. worked on KDD CUP dataset to detect DDoS attack by ensemble method which is combined with SVM and LR (Beulah and Pitchai Manickam, 2022). Bhardwaj et al. worked on CICIDS 2017 and NSL KDD

dataset using DNN with 23 and 40 number of features (Bhardwaj et al., 2020) whereas HailyeTekleselassie worked on CICIDS 2017 dataset using SMO, Bayes Net and RF with 21 number of features (Tekleselassie, 2021).

In our research, ensemble machine learning technique is proposed by combining three traditional machine learning methods; KNN, SVM and DT. Then the accuracy, precision, recall and F1-score of the ensemble method is compared with the traditional machine learning algorithms that are KNN, SVM, DT and NB. As far as our knowledge, Beulah et al. used ensemble machine learning technique by combining SVM and LR with 41 number of features attaining 99.2% accuracy working on KDD CUP dataset and Saikat et al. combined NN, SVM, KNN and DT-C4.5 with 24 number of features attaining 99.1% accuracy working on NSL-KDD dataset, whereas we used only 10 and 11 number of features to detect the DDoS attack attaining 99.4% accuracy working on DDoS classification dataset as this one is newer than KDD CUP and NSL-KDD datasets.

CHAPTER 3

PROPOSED APPROACH

To complete the entire process of identifying DDoS attacks on our dataset, this section covers some phases. First, data is gathered from a reliable online source. The vital features are then obtained using a variety of feature selection strategies and domain knowledge. After that, data is pre-processed to remove any extraneous information, and then traditional and ensemble algorithms are used to detect DDoS attacks along with the performance evaluation shown in Fig. 3.1.

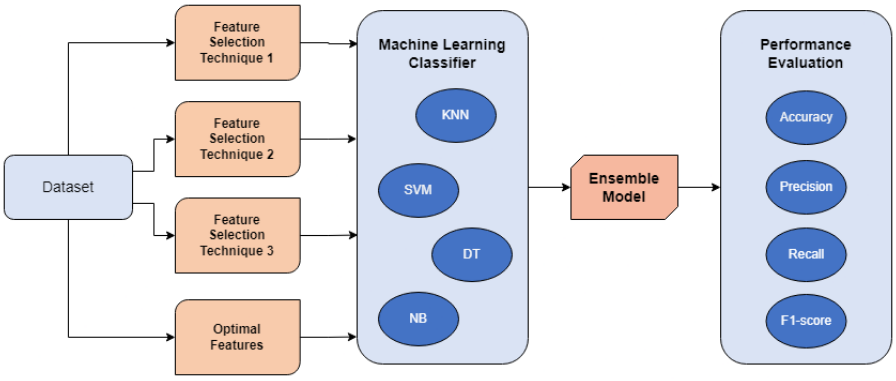

Figure 3.1: Diagram of the workflow

3.1 Dataset

The dataset has been downloaded from Kaggle (Krishna, 2020) and the ddostest csv file of the dataset has been used for both training and testing machine learning models. The dataset contains 17171 rows and 42 columns. The details of each feature are represented by the Table 3.1. The outcome column has three different types of categorical values. There are 9711 repetitions of “normal”, 7458 rows of DDoS and 2 entries of “worm”. The two rows have been dropped as the goal of the research is to detect DDoS attack. In addition, there are no null values in the dataset. Thus, after cleaning the data, the total number of rows is, 17169 and the number of columns is the same as before.

Table 3.1: Dataset Description

Serial No.	**Feature**	**Description**	**Data Type**
1	duration	The time the connection lasted.	int64
2	protocol_type	The communication protocol used (e.g., TCP, UDP).	int64
3	service	The network service on the destination host.	int64
4	flag	The status of the connection (e.g., normal, error).	int64
5	src_bytes	Number of data bytes sent by the source.	int64
6	dst_bytes	Number of data bytes sent to the destination.	int64
7	ddos	Indicates whether the connection is related to a DDoS attack (binary classification).	int64
8	wrong_fragment	Number of incorrect fragments.	int64
9	urgent	Level of urgency.	int64
10	hot	Number of ”hot” indicators.	int64
11	num_failed_0s	Number of failed login attempts.	int64
12	logged_in	Indicates whether a user is logged in.	int64
13	num_compromised	Number of compromised conditions.	int64
14	root_0	Indicates whether the root account was accessed.	int64
15	su_attempted	Indicates if ”su” (superuser) was attempted.	int64

16	num_root	Number of root accesses.	int64
17	num_file_creations	Number of file creation operations.	int64
18	num_0s	Number of ”0” accesses.	int64
19	num_access_files	Number of accesses to files.	int64
20	num_outbound_cmds	Number of outbound commands.	int64
21	is_host_0	Indicates whether the host is a zero-dollar value.	int64
22	is_guest_0	Indicates whether the user is a guest.	int64
23	count	Number of connections to the same host.	int64
24	srv_count	Number of connections to the same service.	int64
25	serror_rate	Percentage of connections with a SYN error.	float64
26	srv_serror_rate	Service-specific error rate.	float64
27	rerror_rate	Percentage of connections with a reject error.	float64
28	srv_rerror_rate	Service-specific reject error rate.	float64
29	same_srv_rate	Percentage of connections to the same service.	float64
30	diff_srv_rate	Percentage of connections to different services.	float64
31	srv_diff_host_rate	Rate of connections to different hosts for the same service.	float64
32	dst_host_count	Number of destination hosts.	int64
33	dst_host_srv_count	Number of destination hosts running the same service.	int64

34	dst_host_same_srv_rate	Percentage of hosts with the same service.	float64
35	dst_host_diff_srv_rate	Percentage of hosts with different services.	float64
36	dst_host_same_src_port_rate	Percentage of connections to the same source port.	float64
37	dst_host_srv_diff_host_rate	Rate of different hosts for the same service.	float64
38	dst_host_serror_rate	Percentage of connections with a destination host SYN error.	float64
39	dst_host_srv_serror_rate	Service-specific destination host SYN error rate.	float64
40	dst_host_rerror_rate	Percentage of connections with a destination host reject error.	float64
41	dst_host_srv_rerror_rate	Service-specific destination host reject error rate.	float64
42	outcome	The target variable indicating the outcome of the connection (e.g., normal, DDoS).	object

There can be many features in a dataset, but not all of them are important for an algorithm to learn. Rather, too many features can confuse the algorithm, and the algorithm might learn unnecessary patterns and predict incorrectly. Thus, it is important to select the important features and use only those features for training a model. In order to obtain a subset from the provided dataset with improved accuracy while using fewer features, feature selection technique is used. The features are chosen to use a variety of factors in this feature selection technique (Kabir et al., 2023) (Bagherzadeh et al., 2021). There are many types of feature selection techniques and filter method is one of them.

3.2 Feature Selection Method

Filter method is one of the popular feature selection technique types along with wrapper method, embedded method etc. Four techniques have been implemented for feature selection in this paper and these are ANOVA, mutual information, feature importance and domain knowledge. Two more feature selection techniques that are Chi-square and RFE have been worked on. However, the results are not competitive with those of the former mentioned ones, RFE and chi-square feature selection technique have not been taken in consideration.

3.2.1 ANOVA

Based on Fig. 3.2 Analysis of Variance (ANOVA) is a feature selection technique that calculates the F-score for every feature to classify the dataset (Zaini and Awang, 2023). ANOVA employs the ANOVA f-test for the features and only considers linear dependency (Ertan, 2020). The scores generated by various techniques differ greatly from one another. Every function actually returns the top features and ranks the relevance of each feature internally. In the Equation (3.1), the ratio of different variance is measured to calculate ANOVA (Azhar et al., 2023).

Figure 3.2: Feature Selection using ANOVA

Let,

- A = variance between groups
- B = variance within groups

$$F = \frac{A}{B} \tag{3.1}$$

$$variance\ between\ groups = \frac{\sum_{i=1}^{k}\sum_{j=1}^{ni} n_i(\overline{Y}_i - \overline{Y})^2}{(n-k)} \tag{3.2}$$

$$variance\ within\ groups = \frac{\sum_i^n n_i(Y_{ij} - \overline{Y_i})^2}{(k-1)} \tag{3.3}$$

A score for each feature has been calculated, and top 10 features have been selected. Fig. 3.3 shows the top 10 features that have scored the highest in case of ANOVA.

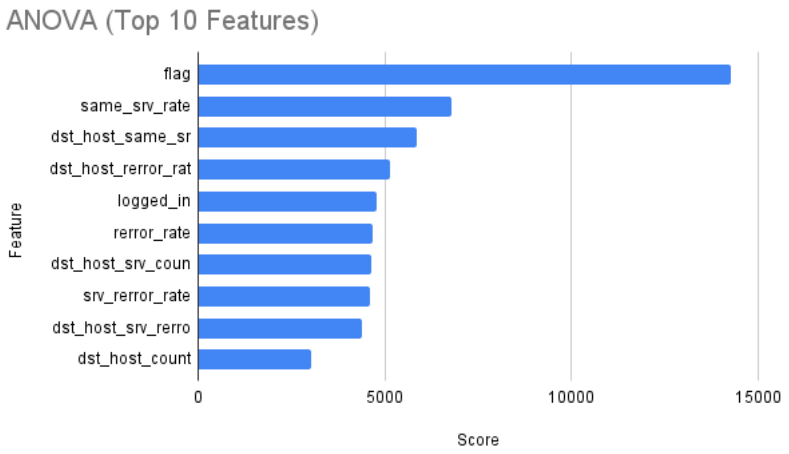

Figure 3.3: ANOVA (Top 10 Features)

The top 11 features and their scores are shown in Fig. 3.4.

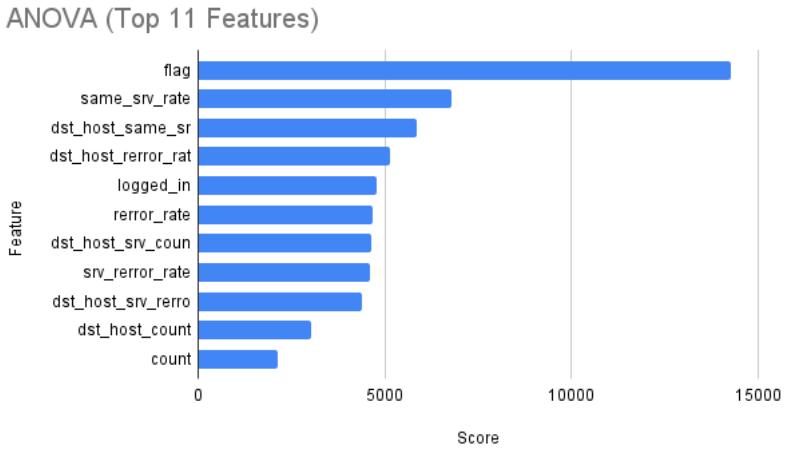

Figure 3.4: ANOVA (Top 11 Features)

3.2.2 Mutual Information

Another feature selection technique is mutual information that lessens feature redundancy (Ma et al., 2023) shown in Fig. 3.5. The mutual information feature selection technique evaluates the relationship between each feature and the target class in order to choose the most effective characteristics for DDoS attack detection (Hashim and Yassin, 2023) (Ertan, 2020).

In essence, this approach makes use of mutual information. It determines which independent variables have the greatest information gain by calculating each one's mutual information value in relation to the dependent variable. To put it another way, it essentially calculates how dependent each feature is on the desired value. There are more dependent variables with a higher score. Eq. (3.4) is used to calculate the mutual information.

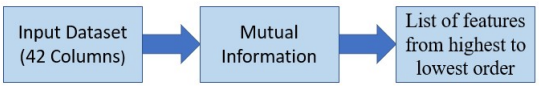

Figure 3.5: Feature Selection using Mutual Information

$$I(M,N) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(M_i,N_i) log \frac{P(M_i|N_i)}{P(M_i)} \tag{3.4}$$

Fig. 3.6 shows the top 10 features that have scored the highest in case of mutual information.

The top 11 features and their scores are shown in Fig. 3.7.

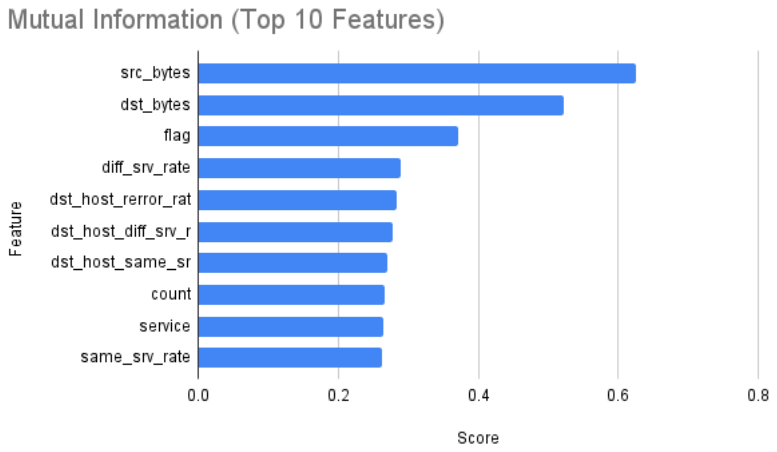

Figure 3.6: Mutual Information (Top 10 Features)

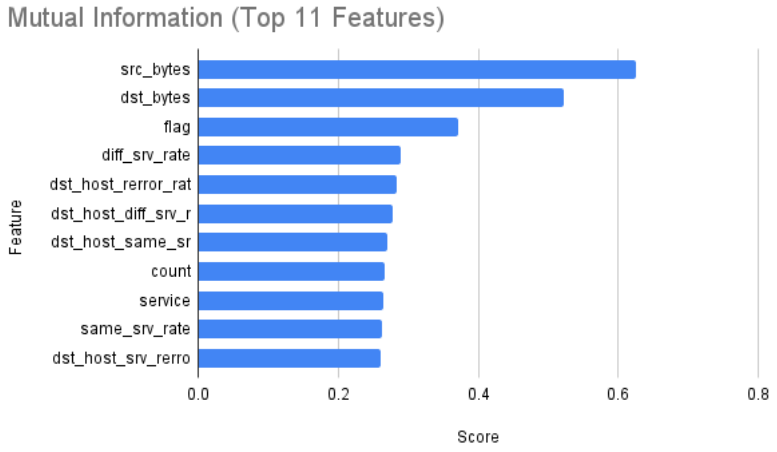

Figure 3.7: Mutual Information (Top 11 Features)

3.2.3 Feature Importance

Utilizing the gini information, the feature importance technique is applied in Fig. 3.8 to obtain the best features for detecting DDoS attacks. Gini information is calculated using a node's impurity reduction. Again, the impurity is determined by how many samples, out of all the samples, reached the node (Pierzyna et al., 2023) (Tikhe and Rana, 2023).

Figure 3.8: Feature Selection using Feature Importance

Fig. 3.9 shows the top 10 features that have scored the highest in case of feature impor-

tance.

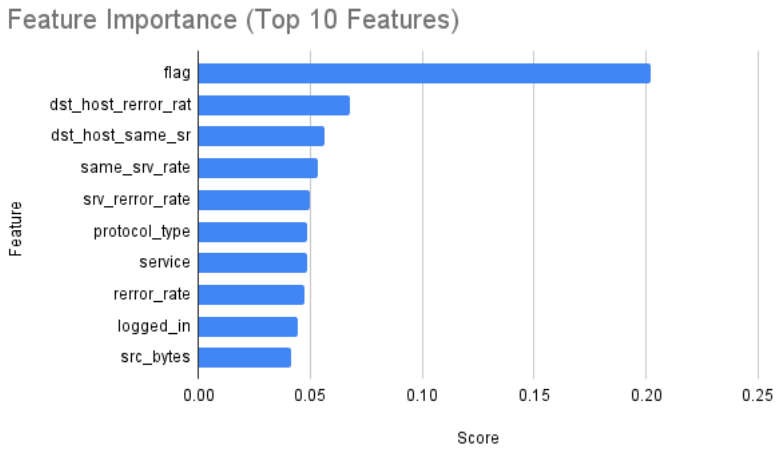

Figure 3.9: Feature Importance (Top 10 Features)

The top 11 features and their scores are shown in Fig. 3.10.

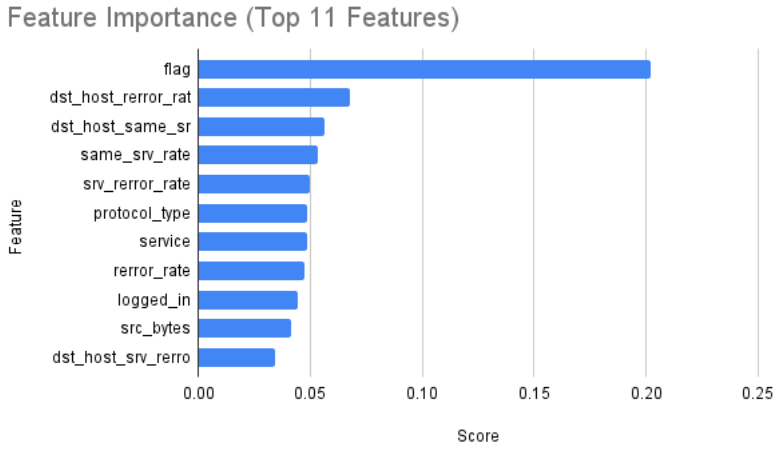

Figure 3.10: Feature Importance (Top 11 Features)

3.2.4 Optimal Features

After implementing the first three feature selection techniques, there were some techniques that were common. These common features along with some other features were considered while creating a subset for the optimal features. The Table 3.2 represents the 10 and 11 optimal features.

Table 3.2: Optimal Features

Serial No.	**Feature**	**10 Features**	**11 Features**
1	flag	Yes	Yes
2	src_bytes	Yes	Yes
3	service	Yes	Yes
4	same_srv_rate	Yes	Yes
5	dst_host_rerror_rate	Yes	Yes
6	rerror_rate	Yes	Yes
7	dst_host_srv_rerror_rate	Yes	Yes
8	count	Yes	Yes
9	dst_host_same_srv_rate	Yes	Yes
10	dst_host_srv_count	Yes	Yes
11	dst_host_diff_srv_rate	No	Yes

3.3 Methodology

The above-mentioned four feature selection techniques have been used to identify top 10 and 11 features.

Firstly, the "ddos" feature has been dropped from the dataset, which makes the total number of features 41. After that, the label column and the features have been separated from each other into two different variables.

Later, for ANOVA technique, SelectKBest function has been used which takes "f_classif" as the value of the key, "score_func". Then, the split dataset i.e. x and y variables have been fit into the technique. In addition, the score value of each feature has been calculated and the top 10 and 11 features associated with top 10 and 11 scores have been selected.

Again, to apply mutual information, "mutual_info_classif" has been passed as an argument to SelectKBest() function and then the divided dataset has been fit into the technique. Later, top 10 and 11 features have been selected.

Furthermore, ExtraTreeClassifier has been used to fit the variables into the feature importance technique. Later, "feature_importances_" has been used to calculate the important features, unlike the previous techniques.

Lastly, the common features from these techniques and some randomly selected features have been selected to find the optimal features. The random features have been selected as the result of trial and error.

After that, the variable x or features and the variable y or the label have been split into train and test data with a ratio of 75 to 25 that is 75% of the rows for training (x_train and y_train variables) and 25% of entries for testing (x_test and y_test variables) which can be seen in Fig. 3.11. Next, the both the training and the testing data are scaled using standard scaler. Finally, the training data is fed to the ML models for both 10 and 11 features.

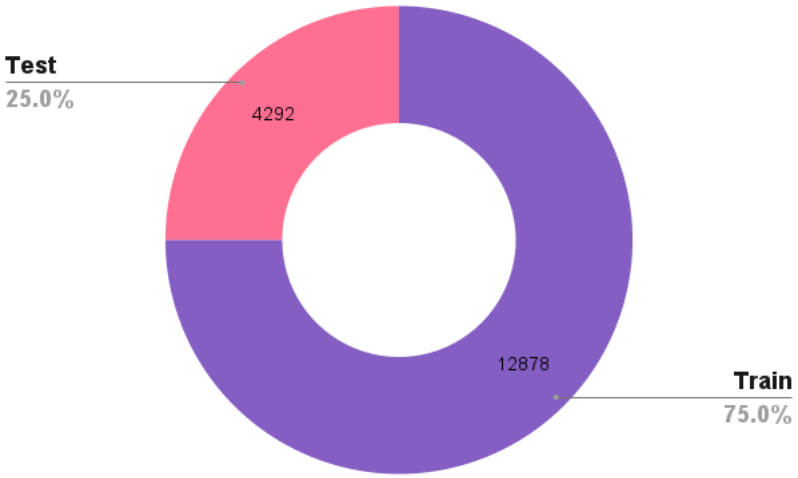

Figure 3.11: Dataset After Train Test Split

3.4 KNN

The number of neighbors considered is 3 and the distance metric used for measuring the distance between points is Manhattan. The power parameter is set to 2 for applying Euclidean distance in this case. After classifying the KNN model, it is trained using the x_train and y_train data. Furthermore, the model is used to predict the x_test and then the

prediction is matched against the y_test.

3.5 SVM

The value of kernel is “rbf” for the Support Vector Classifier, and the random state is set to 0. Later, the model is trained using x_train and y_trian data to predict x_test and then the y_test data is matched against the predictions.

3.6 Decision Tree

The random state has been set to 10 for the decision tree classifier and, like the previous models, the classifier has been trained and used to predict data. Lastly, the accuracy of the model is calculated.

3.7 Naive Bayes

The GaussianNB() function has been implemented for the naive Bayes classifier. The NB classifier is used to predict the x_test and match the results against y_test after training the model using x_train and y_train.

3.8 Ensemble of KNN, NB and DT

After collecting dataset from Kaggle repository, the dataset has been preprocessed. The dataset has a 75% training and 25% testing ratio. Then KNN is implemented considering 3 neighbor nodes and Manhattan distance metric. While applying Naïve Bayes, GaussianNB( ) function has been used. In case of implementing DT, the random state is set to 10 for decision tree classifier. After implementing KNN, NB and DT, the majority votes are taken into account through hard voting. Thus, as the result, the ensemble of KNN, NB and DT is achieved. The workflow of the ensemble model is as shown in Fig. 3.12:

Figure 3.12: The Workflow of the Ensemble of KNN, NB and DT

3.9 Ensemble of KNN, NB and SVM

The dataset has been preprocessed after being collected from the Kaggle source. The training and testing ratio of the dataset is 75% and 25%. Then KNN is implemented considering 3 neighbor nodes and Manhattan distance metric. During the Naive Bayes application, the GaussianNB() function was utilized. While applying SVM, the value of kernel is kept "rbf". The majority of votes for each class among the predictions of KNN, NB and SVM have been calculated through hard voting or averaging. This provides KNN, NB and SVM's ensemble model. The Fig. 3.13 above represents the workflow of the ensemble model:

Figure 3.13: The Workflow of the Ensemble of KNN, NB and SVM

3.10 Ensemble of NB, SVM and DT

After collecting dataset from Kaggle repository, the dataset has been preprocessed. The dataset has a 75% training and 25% testing ratio. While applying Naïve Bayes, GaussianNB( ) function has been used. In case of implementing DT, the random state is set to 10 for decision tree classifier. During the implementation of SVM, the value of kernel is kept "rbf". Hard voting is used to determine the majority vote following the training and prediction of NB, SVM, and DT. As a result, the trio of NB, SVM, and DT is attained. The above Fig. 3.15 is the ensemble model's workflow:

3.11 Ensemble of KNN, SVM, NB and DT

The dataset has been preprocessed after being collected from the Kaggle source. The training and testing ratio of the dataset is 75% and 25%. Then KNN is implemented con-

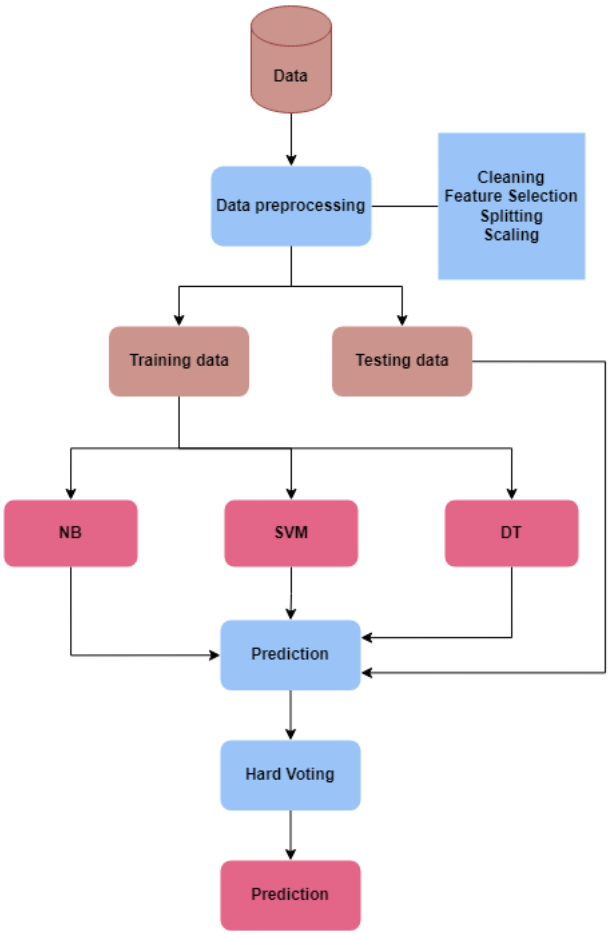

Figure 3.14: The Workflow of the Ensemble of NB, SVM and DT

sidering 3 neighbor nodes and Manhattan distance metric. At the time of applying SVM, the value of kernel is kept "rbf".While applying Naïve Bayes, GaussianNB( ) function has been used. In case of implementing DT, the random state is set to 10 for decision tree classifier. The majority of votes for each class among the predictions of KNN, SVM, NB and DT have been calculated through hard voting or averaging. This provides KNN, SVM, NB and DT's ensemble model. The Fig. 3.13 below represents the workflow of the ensemble model:

3.12 Proposed Ensemble Model

After collecting the dataset from Kaggle repository, the dataset has been preprocessed. The dataset has a 75% training and 25% testing ratio. Then KNN is implemented considering 3 neighbor nodes and Manhattan distance metric. While applying SVM, the value

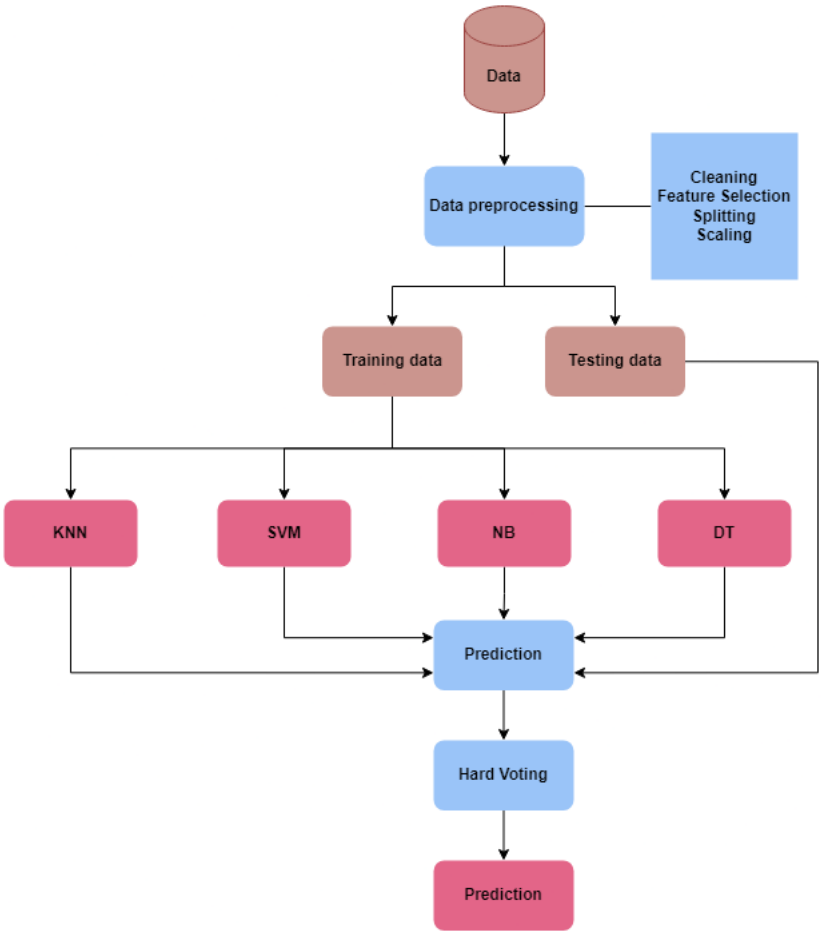

Figure 3.15: The Workflow of the Ensemble of KNN, SVM, NB and DT

of kernel is kept “rbf”. In case of implementing DT, the random state is set to 10 for decision tree classifier. Through hard voting or averaging, the majority of votes for each class among the KNN, SVM and DT predictions have been determined. This offers the ensemble model for KNN, SVM and DT. The proposed ensemble model’s workflow is depicted in the following Fig. 3.16:

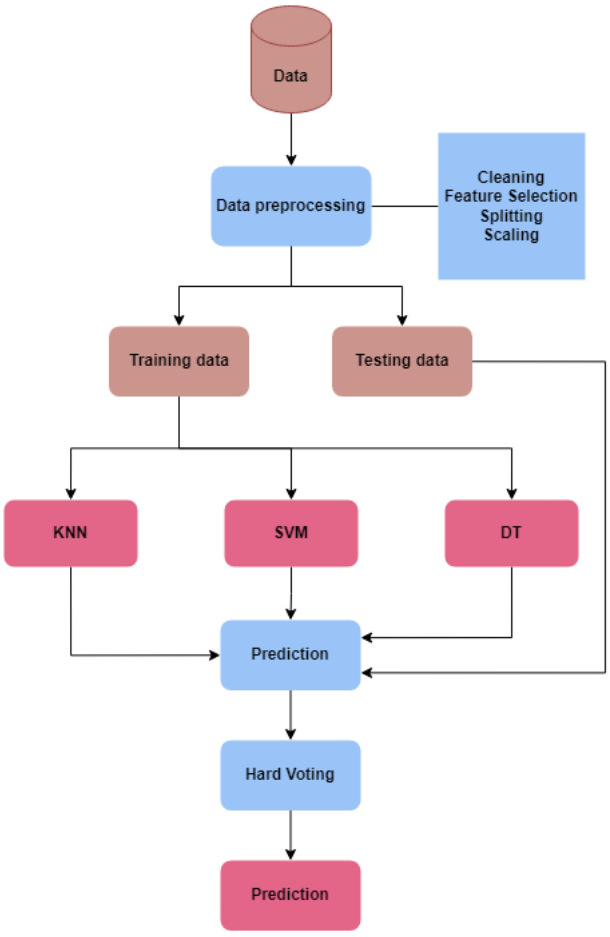

Figure 3.16: The Workflow of the Proposed Ensemble Model (KNN, SVM, DT)

CHAPTER 4

RESULTS

4.1 Experimental Setup

Table 4.1 shows the device configuration on which the research has been conducted. Again, Jupyter Notebook has been used on Anaconda environment to conduct all the experiments for this research. The programming language used was python.

Table 4.1: Experimental setup for the training models

CPU	Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
RAM	8 GB
Storage	1 TB

4.2 Performance Metrics

Our proposed ensemble model has been compared against the traditional models that are KNN, SVM, DT and NB. The comparison has been done over the dataset named DDoS classification. To evaluate the proposed ensemble model four performance metrics have been used- precision, recall, F1-score and accuracy. The following provides a description of these metrics.

4.2.1 Accuracy

Accuracy (Söğüt and Erdem, 2023) is the ratio of the accurately identified DDoS attack or benign instances to all instances showing in the Equation (4.1).

$$accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{4.1}$$

4.2.2 Precision

Precision, as shown in the Equation (4.2), is the ratio of accurately classified DDoS instances to the total number of incorrect classifications of benign instances as an attack and the DDoS attack instances (Saravanakumar et al., 2023).

$$precision = \frac{TP}{TP + FP} \tag{4.2}$$

4.2.3 Recall

In the Eq. (4.3), recall is measured by the proportion of correctly classified DDoS instances to those that were incorrectly classified as benign instances as well as accurately classified DDoS instances (Das et al., 2020).

$$recall = \frac{TP}{TP + FN} \tag{4.3}$$

4.2.4 F1 Score

The Eq. (4.4) shows the application of harmonic mean of recall and precision to calculate F1 score (Das et al., 2019).

$$F1score = 2 * \frac{P * R}{P + R} \tag{4.4}$$

In this work, recall is more important than precision. From precision, we get False Positive and from recall, we get False Negative. If there is not any ddos attack, but the model classifies it as attack is not as harmful as there is ddos attack but the model classifying it as normal. SO, recall must be emphasized more in this thesis work.

4.3 Results

This section provides the performance evaluation using four performance metrics- Accuracy, Precision, Recall and F1-score. The results have been evaluated by using the optimal features and the features that have been found using three feature selection techniques- ANOVA, Mutual Information and Feature Importance.

4.3.1 ANOVA

If precision and recall are taken into consideration then Voting (KNN, SVM, DT) scores the highest with an F1-score of 0.9337 while KNN stands at the second position by having 0.9304 as its F1-score followed by Voting (KNN, SVM, NB, DT), DT, Voting (KNN, NB, DT), SVM, Voting (KNN, SVM, NB), Voting (SVM, NB, DT) and NB having 0.9290, 0.9256, 0.9242, 0.9106, 0.9082, 0.9001 and 0.8583 as their F1-score which can be seen in Table 4.2.

Here, it is seen that the accuracy of voting (KNN, SVM, NB) is 0.9245 which is less than KNN that is 0.9411. The accuracy of SVM is 0.9262 and NB is 0.8861. As we know, voting is done through majority voting, so by the voting of KNN, SVM and NB the accuracy has been decreased than the base classifier, KNN.

Table 4.2: Performance of the models using ANOVA (10 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9411	0.9702	0.8938	0.9304
SVM	0.9262	0.9764	0.8531	0.9106
DT	0.9376	0.9754	0.8805	0.9256
NB	0.8861	0.9500	0.7828	0.8583
Voting (KNN, SVM, DT)	0.9443	0.9831	0.8890	0.9337
Voting (KNN, SVM, NB)	0.9245	0.9786	0.8473	0.9082
Voting (KNN, NB, DT)	0.9369	0.9816	0.8732	0.9242
Voting (SVM, NB, DT)	0.9185	0.9831	0.8890	0.9001
Voting (KNN, SVM, NB, DT)	0.9401	0.9728	0.8890	0.9290

From Fig. 4.1 it can be seen that Voting (KNN, SVM, DT) has an accuracy score of 0.9443 whereas KNN, a traditional ML algorithm has scored 0.9411 in terms of accuracy. On the other hand, traditional algorithms SVM, DT and NB have scored 0.9262, 0.9376 and 0.8861 respectively in relation to accuracy.

Fig. 4.2 shows that out of 1892 DDoS instances, the proposed ensemble model classifies 1682 of them correctly. In addition, as for the Benign class, the model classifies 29 out of 2401 entities inaccurately.

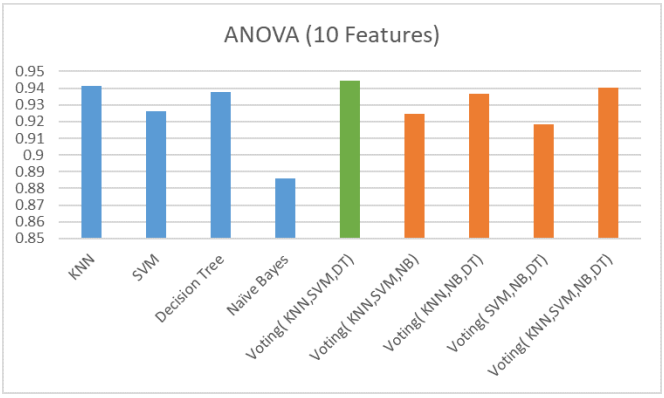

Figure 4.1: Accuracy of the models using ANOVA (10 features)

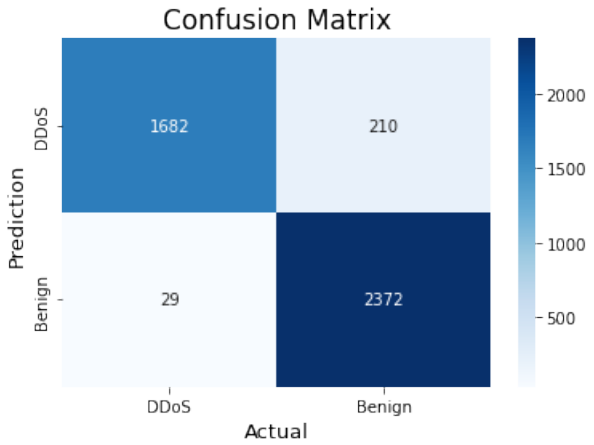

Figure 4.2: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using ANOVA (10 features)

The combined model of KNN, SVM and NB accurately classifies 1603 out of 1892, or 84.73%. In addition, the model correctly classifies 2366 occurrences of 2401 in Figure 4.3.

Out of 1892 DDoS occurrences, Fig. 4.4 demonstrates that 1652 of them are accurately classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 31 out of 2401 elements in the Benign class are incorrectly classified by the model.

The ensemble model of SVM, NB and DT classifies 1577 of 1892 i.e. 83.35% correctly. Furthermore, the model classifies 2366 instances of 2401 accurately in Fig. 4.5.

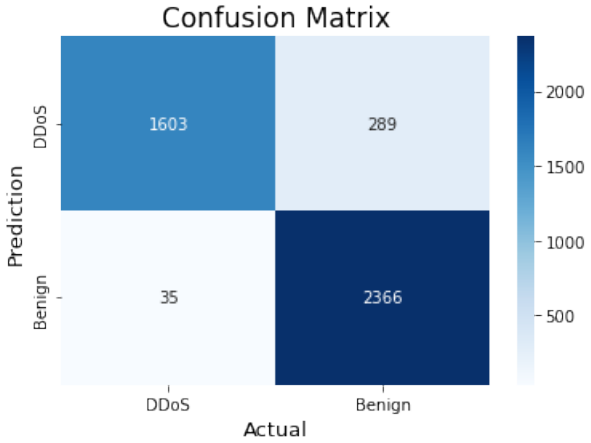

Figure 4.3: Confusion matrix of the ensemble model of KNN, SVM and NB using ANOVA (10 features)

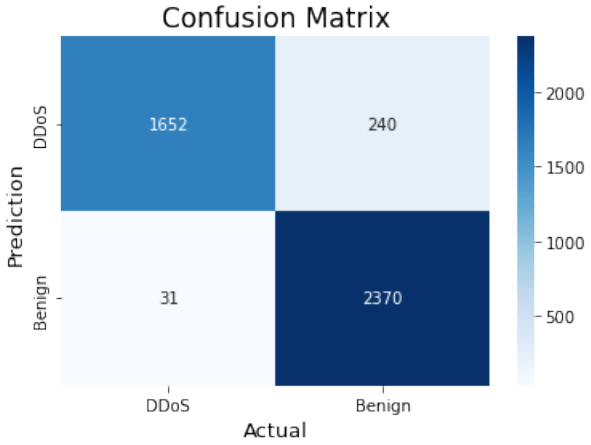

Figure 4.4: Confusion matrix of the ensemble model of KNN, NB and DT using ANOVA (10 features)

Figure 4.6 demonstrates that 1682 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 47 out of 2401 instances in the Benign class.

In terms of precision, Voting (KNN, SVM, NB) stands at the top with 0.9812 and the highest recall is 0.9561 by KNN as shown in Table 4.3. As the recall of the ensemble model of KNN, SVM, NB is 0.8536 which is the third least value in the recall list, the ensemble model of KNN, SVM, DT ends up at the top with an F1-score of 0.9430.

From Fig. 4.7 it can be seen that, the accuracy of voting (KNN, SVM, DT) stands at the

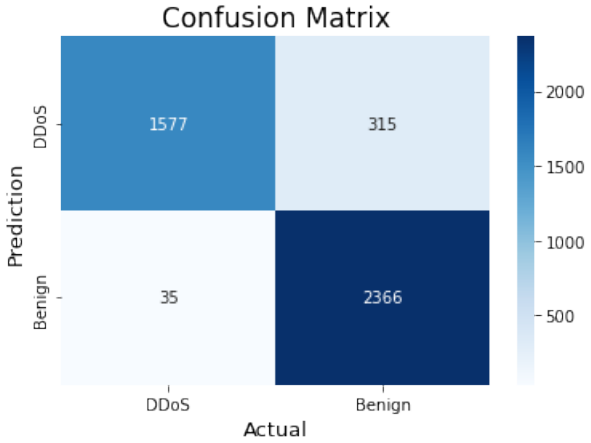

Figure 4.5: Confusion matrix of the ensemble model of SVM, NB and DT using ANOVA (10 features)

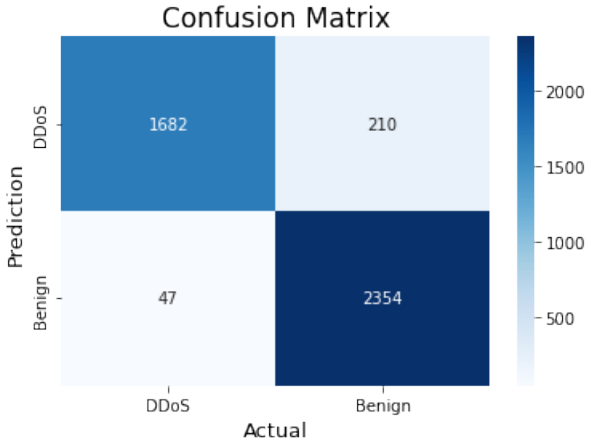

Figure 4.6: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using ANOVA (10 features)

top with 0.9499 in case of 11 features followed by 0.9462 of Voting (KNN, SVM, NB, DT). On the other hand, the lowest accuracy which is 0.8870 is achieved by NB.

Fig. 4.8 shows that out of 1892 DDoS instances, the proposed ensemble model (KNN, SVM, DT) classifies 1778 of them correctly. In addition, as for the Benign class, the model classifies 101 out of 2401 entities inaccurately.

The combined model of KNN, SVM and NB accurately classifies 1615 out of 1892, or 85.36%. In addition, the model correctly classifies 2370 occurrences of 2401 in Figure 4.9.

Table 4.3: Performance of the models using ANOVA (11 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9439	0.9197	0.9561	0.9375
SVM	0.9278	0.9782	0.8552	0.9126
DT	0.9425	0.9359	0.9334	0.9346
NB	0.8870	0.9513	0.7838	0.8595
Voting (KNN, SVM, DT)	0.9499	0.9462	0.9397	0.9430
Voting (KNN, SVM, NB)	0.9283	0.9812	0.8536	0.9129
Voting (KNN, NB, DT)	0.9439	0.9465	0.9249	0.9356
Voting (SVM, NB, DT)	0.9222	0.9809	0.8399	0.9049
Voting (KNN, SVM, NB, DT)	0.9462	0.9383	0.9397	0.9390

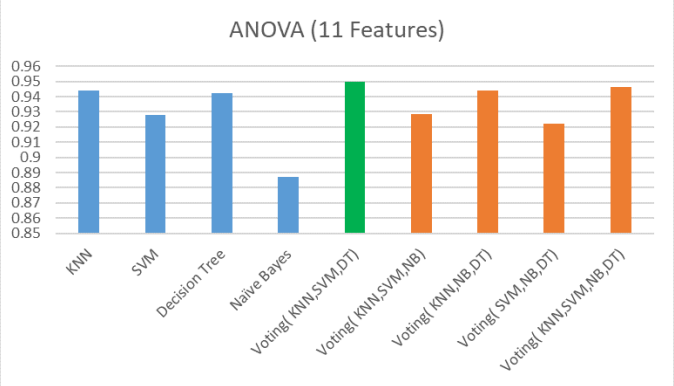

Figure 4.7: Accuracy of the models using ANOVA (11 features)

Out of 1892 DDoS occurrences, Fig. 4.10 demonstrates that 1750 of them are accurately classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 99 out of 2401 elements in the Benign class are incorrectly classified by the model.

The ensemble model of SVM, NB and DT classifies 1589 of 1892 i.e. 83.99% correctly. Furthermore, the model classifies 2370 instances of 2401 accurately in Fig. 4.11.

Figure 4.12 demonstrates that 1778 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 117 out of 2401 instances in the Benign class.

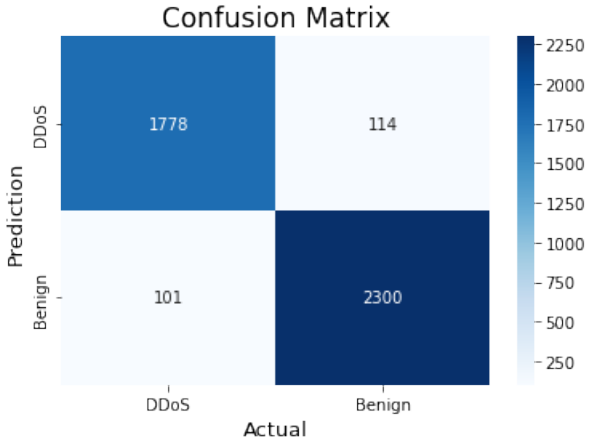

Figure 4.8: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using ANOVA (11 Features)

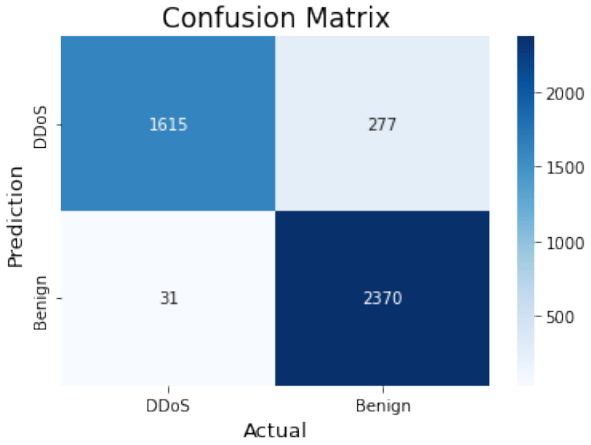

Figure 4.9: Confusion matrix of the ensemble model of KNN, SVM and NB using ANOVA (11 Features)

4.3.2 Mutual Information

Table 4.4 shows Voting (KNN, SVM, DT) tops the F1-score list with a score of 0.9814 followed by that of KNN which is 0.9776. On the other hand, NB is at the bottom with 0.8654 as its F1-score.

The Fig. 4.13 shows 0.9837 to be the highest accuracy which is of Voting (KNN, SVM, DT) followed by KNN's accuracy score of 0.9804. The other ensemble models have accuracy scores of 0.9718, 0.9720, 0.9727 and 0.9727 by Voting (KNN, NB, DT), Voting (KNN, SVM, NB), Voting (SVM, NB, DT) and Voting (KNN, SVM, NB, DT) respectively.

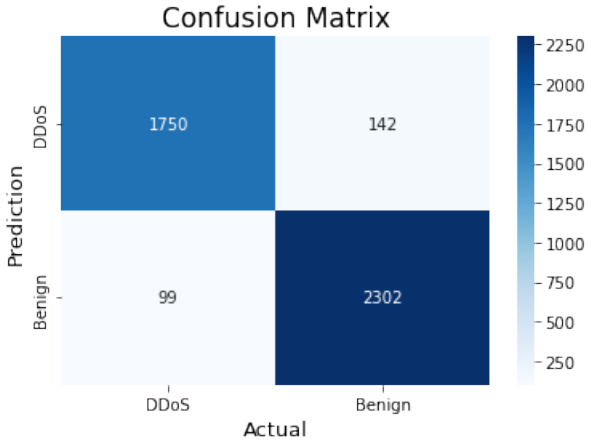

Figure 4.10: Confusion matrix of the ensemble model of KNN, NB and DT using ANOVA (11 Features)

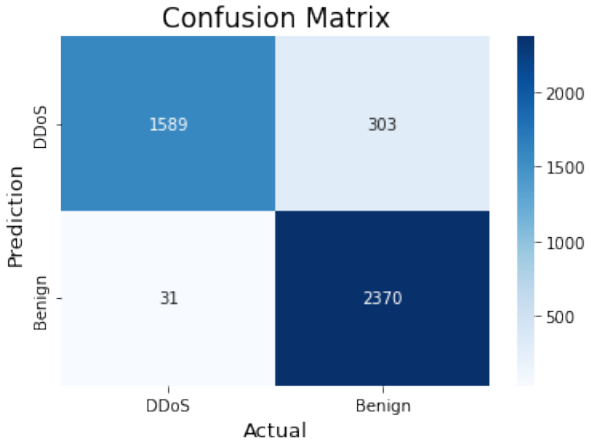

Figure 4.11: Confusion matrix of the ensemble model of SVM, NB and DT using ANOVA (11 Features)

Fig. 4.14 shows that out of 1892 DDoS instances, the proposed ensemble model classifies 1849 of them correctly. In addition, as for the Benign class, the model classifies 27 out of 2401 entities inaccurately.

The combined model of KNN, SVM and NB accurately classifies 1802 out of 1892, or 95.24%. In addition, the model correctly classifies 2371 occurrences of 2401 in Figure 4.15.

Out of 1892 DDoS occurrences, Fig. 4.16 demonstrates that 1801 of them are accurately

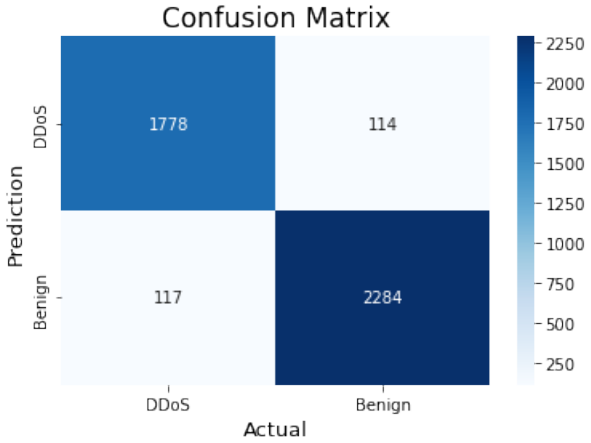

Figure 4.12: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using ANOVA (11 Features)

Table 4.4: Performance of the models using Mutual Information (10 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9804	0.9871	0.9683	0.9776
SVM	0.9769	0.9792	0.9683	0.9737
DT	0.9639	0.9738	0.9434	0.9584
NB	0.8901	0.9399	0.8018	0.8654
Voting (KNN, SVM, DT)	0.9837	0.9856	0.9773	0.9814
Voting (KNN, SVM, NB)	0.9720	0.9836	0.9524	0.9678
Voting (KNN, NB, DT)	0.9718	0.9836	0.9519	0.9675
Voting (SVM, NB, DT)	0.9727	0.9790	0.9588	0.9688
Voting (KNN, SVM, NB, DT)	0.9727	0.9790	0.9588	0.9688

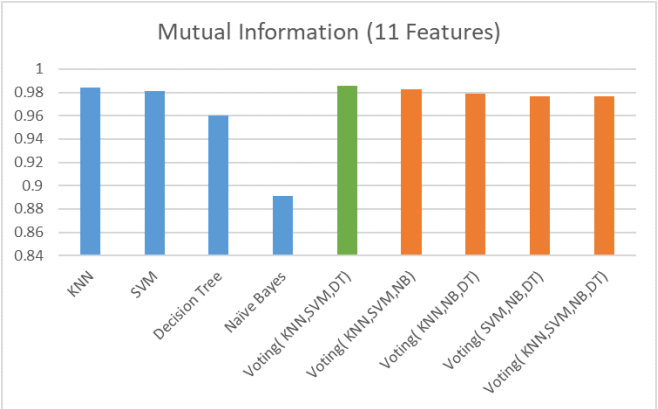

Figure 4.13: Accuracy of the models using mutual information (10 features)

classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 30 out of 2401 elements in the Benign class are incorrectly classified by the model.

The ensemble model of SVM, NB and DT classifies 1814 of 1892 i.e. 95.88% correctly.

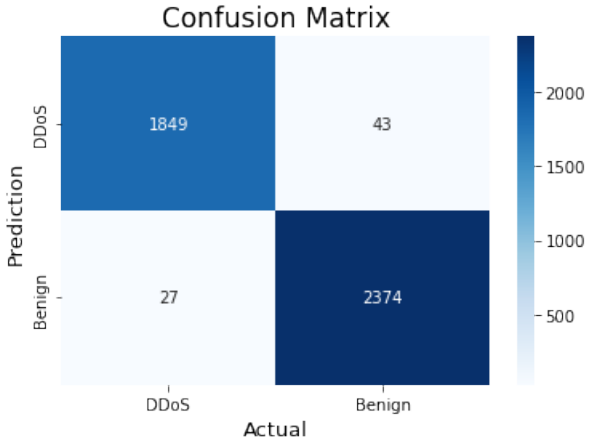

Figure 4.14: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using mutual information (10 Features)

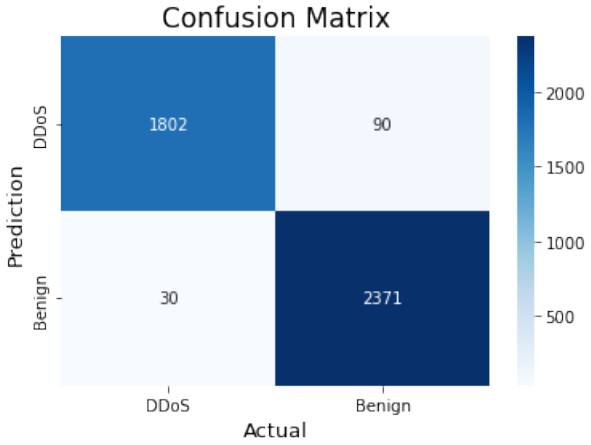

Figure 4.15: Confusion matrix of the ensemble model of KNN, SVM and NB using mutual information (10 Features)

Furthermore, the model classifies 2362 instances of 2401 accurately in Fig. 4.17.

Figure 4.18 demonstrates that 1852 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 40 out of 2401 instances in the Benign class.

Among the traditional algorithms, KNN is performing the best in case of mutual information (with 11 features) with 98.88% precision and 98.41% recall, showed in Table 4.5. Other traditional algorithms SVM, DT and NB have 0.9835; 0.9741, 0.9762; 0.9318 and 0.9541; 0.7918 in precision and recall respectively.

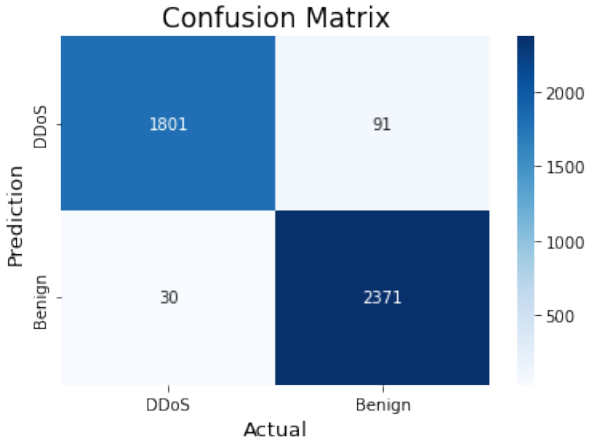

Figure 4.16: Confusion matrix of the ensemble model of KNN, NB and DT using mutual information (10 Features)

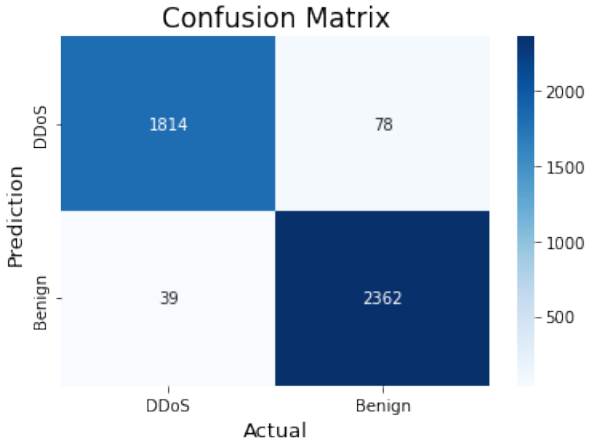

Figure 4.17: Confusion matrix of the ensemble model of SVM, NB and DT using mutual information (10 Features)

Fig. 4.19 reflects that ensemble methods such as Voting (KNN, SVM, DT), Voting (KNN, SVM, NB), Voting (KNN, NB, DT), Voting (SVM, NB, DT) and Voting (KNN, SVM, NB, DT) have scored 0.9856, 0.9830, 0.9786, 0.9765 and 0.9765 in terms of accuracy. Furthermore, most of the traditional algorithms perform well, except NB.

Fig. 4.20 shows that out of 1892 DDoS instances, the proposed ensemble model classifies 1852 of them correctly. In addition, as for the Benign class, the model classifies 22 out of 2401 entities inaccurately.

The combined model of KNN, SVM and NB accurately classifies 1842 out of 1892, or

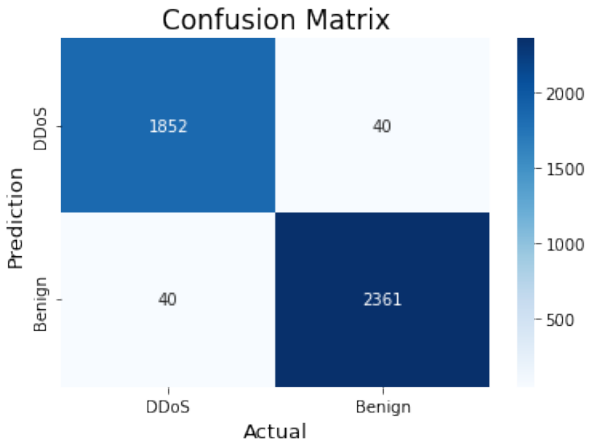

Figure 4.18: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using mutual information (10 Features)

Table 4.5: Performance of the models using Mutual Information (11 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9841	0.9888	0.9841	0.9865
SVM	0.9814	0.9835	0.9741	0.9788
DT	0.9599	0.9762	0.9318	0.9535
NB	0.8915	0.9541	0.7918	0.8654
Voting (KNN, SVM, DT)	0.9856	0.9883	0.9789	0.9835
Voting (KNN, SVM, NB)	0.9830	0.9877	0.9736	0.9806
Voting (KNN, NB, DT)	0.9786	0.9875	0.9635	0.9754
Voting (SVM, NB, DT)	0.9765	0.9848	0.9614	0.9730
Voting (KNN, SVM, NB, DT)	0.9765	0.9848	0.9614	0.9730

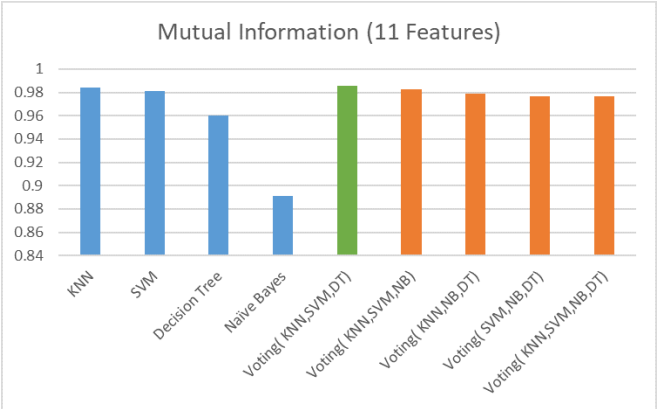

Figure 4.19: Accuracy of the models using mutual information (11 features)

97.36%. In addition, the model correctly classifies 2378 occurrences of 2401 in Figure 4.21.

Out of 1892 DDoS occurrences, Fig. 4.22 demonstrates that 1823 of them are accurately

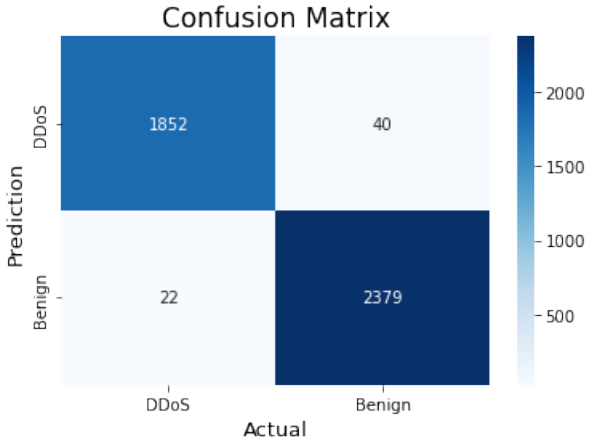

Figure 4.20: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using mutual information (11 Features)

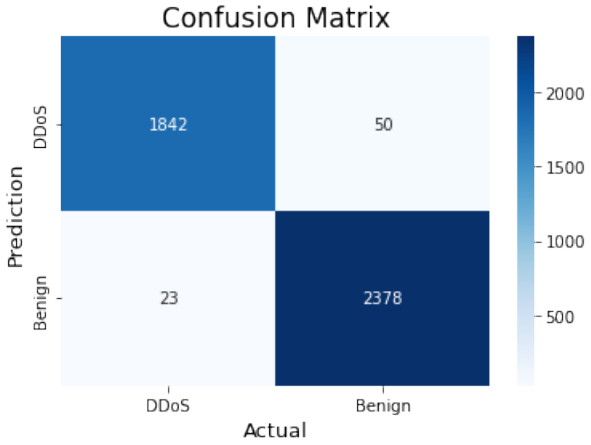

Figure 4.21: Confusion matrix of the ensemble model of KNN, SVM and NB using mutual information (11 Features)

classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 23 out of 2401 elements in the Benign class are incorrectly classified by the model.

The ensemble model of SVM, NB and DT classifies 1819 of 1892 i.e. 96.14% correctly. Furthermore, the model classifies 2373 instances of 2401 accurately in Fig. 4.23.

Figure 4.24 demonstrates that 1778 out of 1854 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 29 out of 2401 instances in the Benign class.

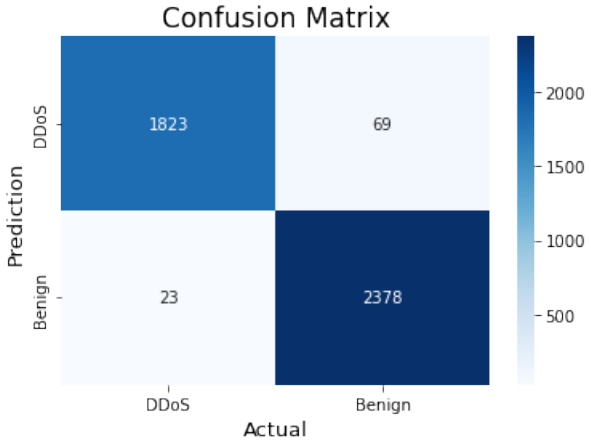

Figure 4.22: Confusion matrix of the ensemble model of KNN, NB and DT using mutual information (11 Features)

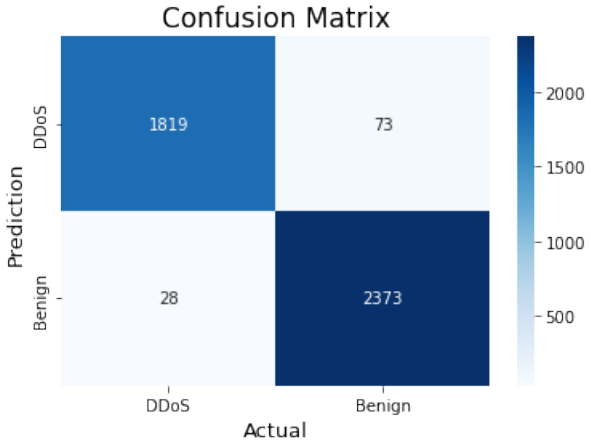

Figure 4.23: Confusion matrix of the ensemble model of SVM, NB and DT using mutual information (11 Features)

4.3.3 Feature Importance

Table 4.6 shows that, in terms of precision, KNN scored the highest, which is 0.9827. The next highest score for precision is 0.9826 that is for voting (KNN, SVM, DT). The highest score of F1-score is 0.9842 which is for Voting (KNN, SVM, DT) and the lowest F1-score is 0.8597 by NB.

Fig. 4.25 represents that two models have an accuracy score over 0.98 and among them, Voting (KNN, SVM, DT) has the highest accuracy which is 0.9860. The model is followed by Voting (KNN, SVM, DT), SVM and KNN with accuracy score of 0.98181,

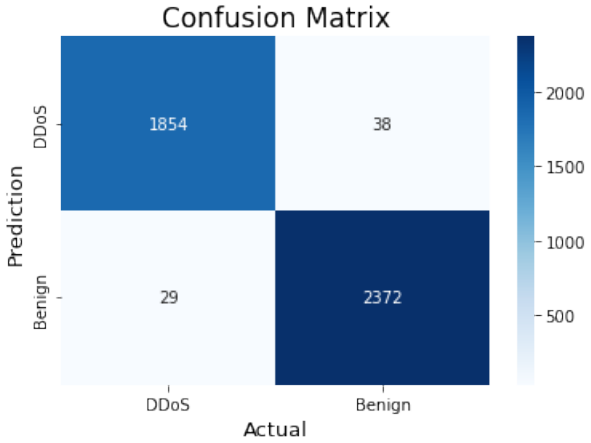

Figure 4.24: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using mutual information (11 Features)

Table 4.6: Performance of the models using Feature Importance (10 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9751	0.9827	0.9604	0.9714
SVM	0.9788	0.9747	0.9773	0.9760
DT	0.9625	0.9806	0.9334	0.9564
NB	0.8870	0.9495	0.7854	0.8597
Voting (KNN, SVM, DT)	0.9860	0.9826	0.9857	0.9842
Voting (KNN, SVM, NB)	0.9697	0.9788	0.9519	0.9652
Voting (KNN, NB, DT)	0.9669	0.9818	0.9424	0.9617
Voting (SVM, NB, DT)	0.9730	0.9779	0.9604	0.9691
Voting (KNN, SVM, NB, DT)	0.9818	0.9734	0.9857	0.9795

0.9788 and 0.9751 respectively.

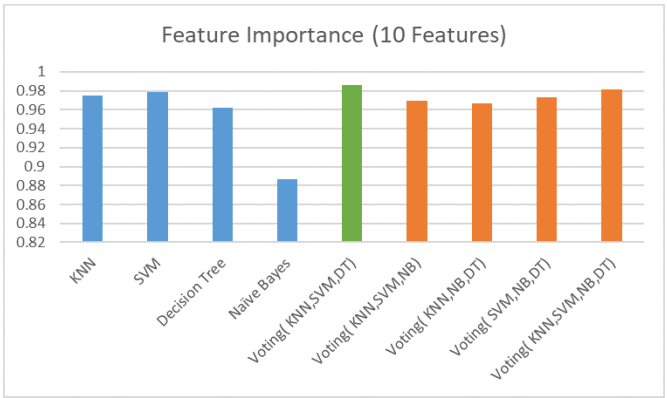

Figure 4.25: Accuracy of the models using feature importance (10 features)

Fig. 4.26 shows that out of 1892 DDoS instances, the proposed ensemble model classifies

1865 of them correctly. In addition, as for the Benign class, the model classifies 33 out of 2401 entities inaccurately.

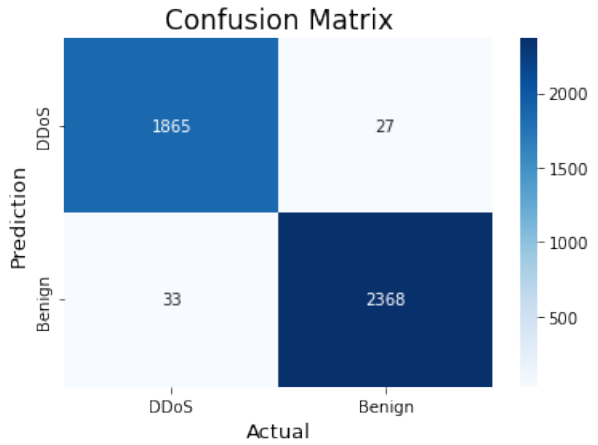

Figure 4.26: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using feature importance (10 Features)

The combined model of KNN, SVM and NB accurately classifies 1801 out of 1892, or 95.19%. In addition, the model correctly classifies 2366 occurrences of 2362 in Figure 4.27.

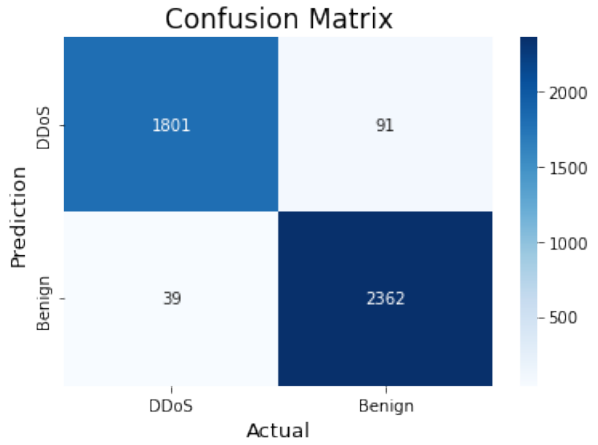

Figure 4.27: Confusion matrix of the ensemble model of KNN, SVM and NB using feature importance (10 Features)

Out of 1892 DDoS occurrences, Fig. 4.28 demonstrates that 1783 of them are accurately classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 33 out

of 2401 elements in the Benign class are incorrectly classified by the model.

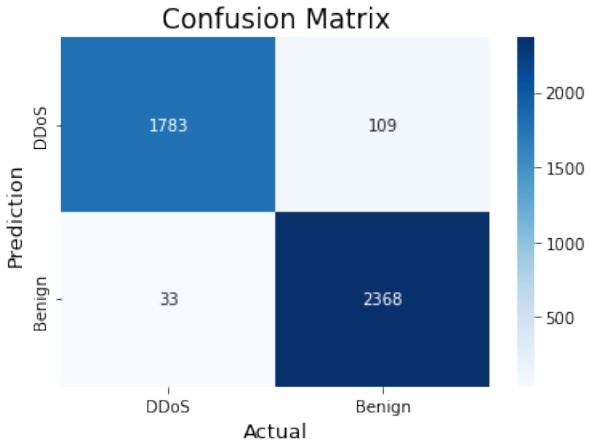

Figure 4.28: Confusion matrix of the ensemble model of KNN, NB and DT using feature importance (10 Features)

The ensemble model of SVM, NB and DT classifies 1817 of 1892 i.e. 96.03% correctly. Furthermore, the model classifies 2360 instances of 2401 accurately in Fig. 4.29.

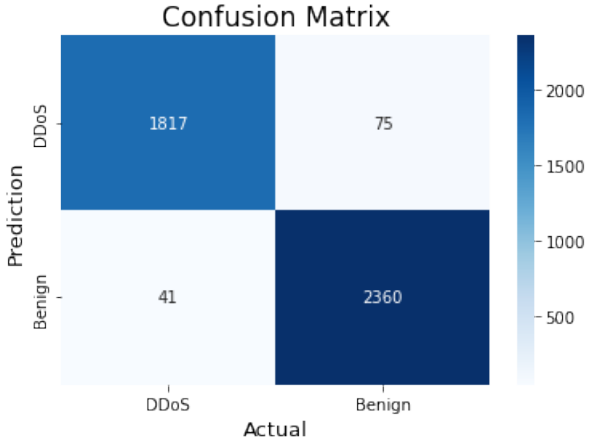

Figure 4.29: Confusion matrix of the ensemble model of SVM, NB and DT using feature importance (10 Features)

Figure 4.30 demonstrates that 1865 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 51 out of 2401 instances in the Benign class.

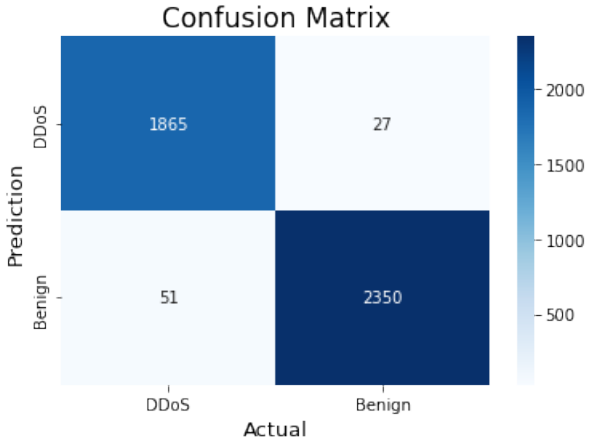

Figure 4.30: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using feature importance (10 Features)

Among the traditional algorithms, SVM is performing the best in case of feature importance (with 11 features) with an F1-score of 0.9789, showed in Table 4.7. However, Voting (KNN, SVM, DT) performs far better with an F1-score of 0.9871.

Table 4.7: Performance of the models using Feature Importance (11 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9760	0.9833	0.9619	0.9725
SVM	0.9814	0.9753	0.9826	0.9789
DT	0.9711	0.9825	0.9514	0.9667
NB	0.8861	0.9494	0.7833	0.8584
Voting (KNN, SVM, DT)	0.9886	0.9832	0.9910	0.9871
Voting (KNN, SVM, NB)	0.9704	0.9794	0.9530	0.9660
Voting (KNN, NB, DT)	0.9676	0.9824	0.9434	0.9625
Voting (SVM, NB, DT)	0.9755	0.9786	0.9656	0.9721
Voting (KNN, SVM, NB, DT)	0.9846	0.9745	0.9901	0.9827

Fig. 4.31 reflects that ensemble methods such as Voting (KNN, SVM, DT), Voting (KNN, SVM, NB), Voting (KNN, NB, DT), Voting (SVM, NB, DT) and Voting (KNN, SVM, DT, NB) have scored 0.9886, 0.9704, 0.9676, 0.9755 and 0.9846 in terms of accuracy. In addition, SVM, KNN and DT score 0.9814, 0.9760 and 0.9711 respectively.

Fig. 4.32 shows that out of 1892 DDoS instances, the proposed ensemble model classifies 1875 of them correctly. In addition, as for the Benign class, the model classifies 32 out of 2401 entities inaccurately.

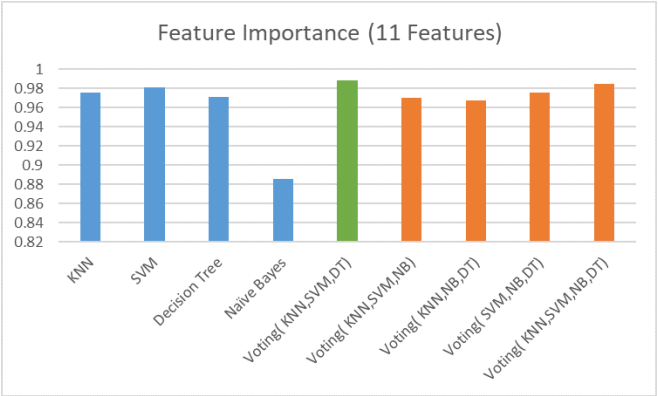

Figure 4.31: Accuracy of the models using feature importance (11 features)

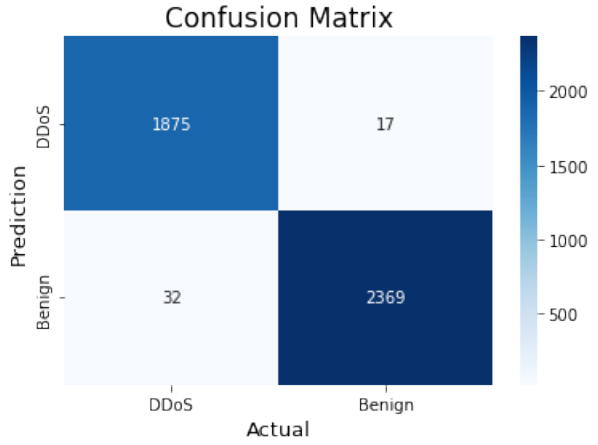

Figure 4.32: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using feature importance (11 Features)

The combined model of KNN, SVM and NB accurately classifies 1803 out of 1892, or 95.30%. In addition, the model correctly classifies 2363 occurrences of 2401 in Figure 4.33.

Out of 1892 DDoS occurrences, Fig. 4.34 demonstrates that 1785 of them are accurately classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 32 out of 2401 elements in the Benign class are incorrectly classified by the model.

The ensemble model of SVM, NB and DT classifies 1827 of 1892 i.e. 96.56% correctly. Furthermore, the model classifies 2361 instances of 2401 accurately in Fig. 4.35.

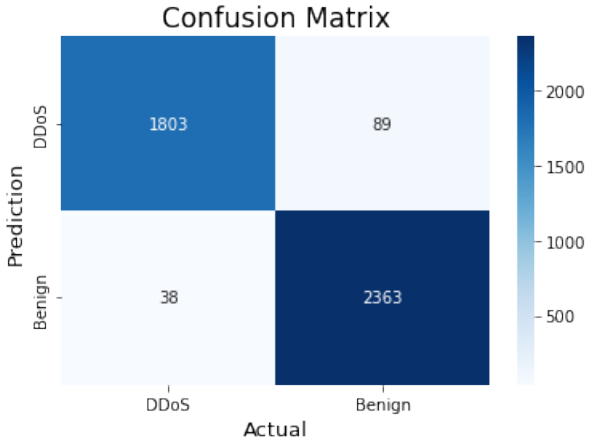

Figure 4.33: Confusion matrix of the ensemble model of KNN, SVM and NB using feature importance (11 Features)

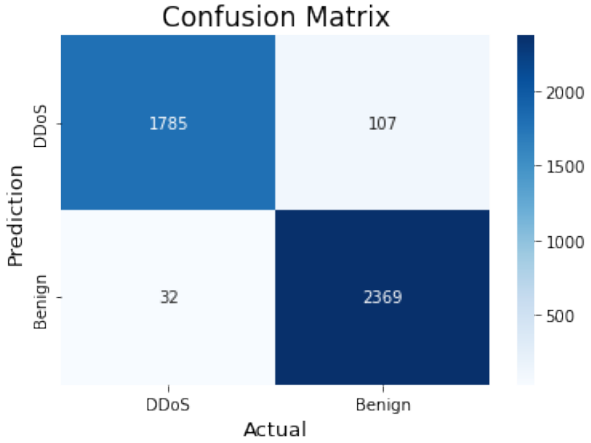

Figure 4.34: Confusion matrix of the ensemble model of KNN, NB and DT using feature importance (11 Features)

Figure 4.36 demonstrates that 1875 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 49 out of 2401 instances in the Benign class.

4.3.4 Optimal Features

Table 4.8 shows Voting (KNN, SVM, DT) tops the F1-score list with a score of 0.9910 followed by that of SVM which is 0.9845. On the other hand, NB is at the bottom with 0.8581 as its F1-score.

The Fig. 4.37 shows 0.9921 to be the highest accuracy which is of Voting (KNN, SVM, DT) followed by SVM's accuracy score of 0.9863. The other ensemble methods have

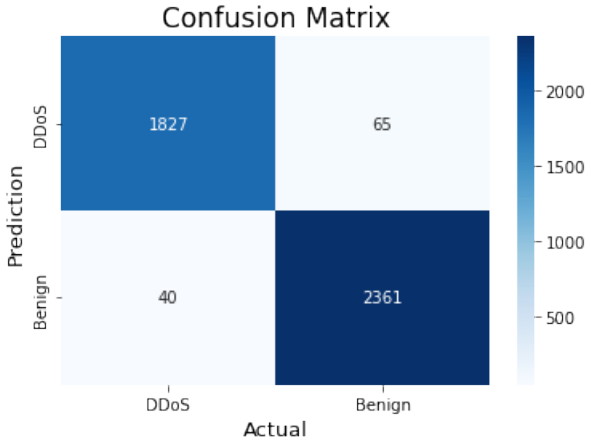

Figure 4.35: Confusion matrix of the ensemble model of SVM, NB and DT using feature importance (11 Features)

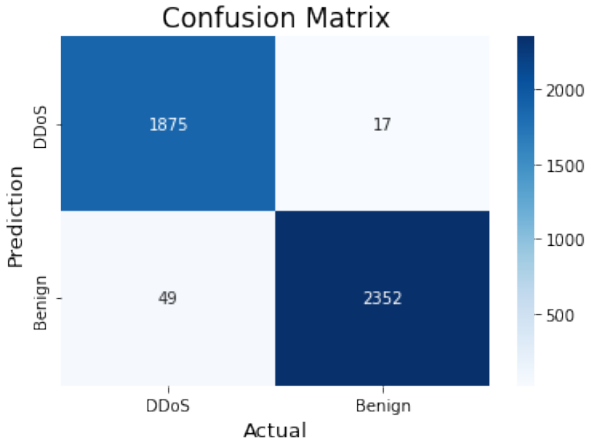

Figure 4.36: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using feature importance (11 Features)

accuracy scores of 0.9790, 0.9790, 0.9783 and 0.9732 by Voting (KNN, SVM, NB, DT), Voting (SVM, NB, DT), Voting (KNN, SVM, NB) and Voting (KNN, NB, DT) respectively.

Fig. 4.38 shows that out of 1892 DDoS instances, the proposed ensemble model classifies 1872 of them correctly. In addition, as for the Benign class, the model classifies 14 out of 2401 entities inaccurately.

The combined model of KNN, SVM and NB accurately classifies 1827 out of 1892, or 96.56%. In addition, the model correctly classifies 2373 occurrences of 2401 in Figure 4.39.

Table 4.8: Performance of the models using Optimal Features (10 features)

Algorithm	Accuracy	Precision	Recall	F1-score
KNN	0.9818	0.9855	0.9730	0.9793
SVM	0.9863	0.9816	0.9873	0.9845
DT	0.9776	0.9880	0.9609	0.9743
NB	0.8856	0.9465	0.7849	0.8581
Voting (KNN, SVM, DT)	0.9921	0.9926	0.9894	0.9910
Voting (KNN, SVM, NB)	0.9783	0.9849	0.9656	0.9752
Voting (KNN, NB, DT)	0.9732	0.9901	0.9487	0.9690
Voting (SVM, NB, DT)	0.9790	0.9829	0.9693	0.9761
Voting (KNN, SVM, NB, DT)	0.9790	0.9829	0.9693	0.9761

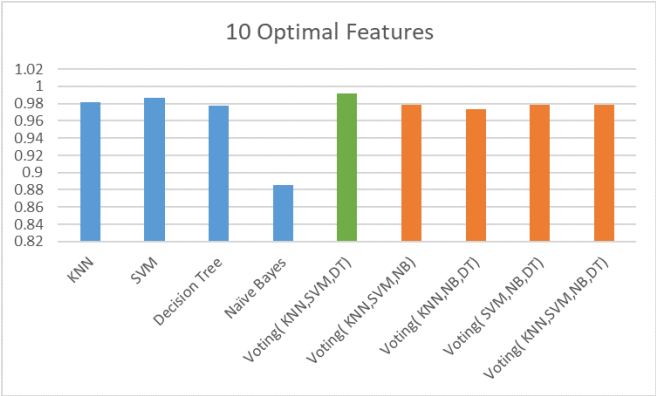

Figure 4.37: Accuracy of the models using optimal features (10 features)

Out of 1892 DDoS occurrences, Fig. 4.40 demonstrates that 1795 of them are accurately classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 18 out of 2401 elements in the Benign class are incorrectly classified by the model.

The ensemble model of SVM, NB and DT classifies 1834 of 1892 i.e. 96.93% correctly. Furthermore, the model classifies 2369 instances of 2401 accurately in Fig. 4.41.

Figure 4.42 demonstrates that 1872 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 33 out of 2401 instances in the Benign class.

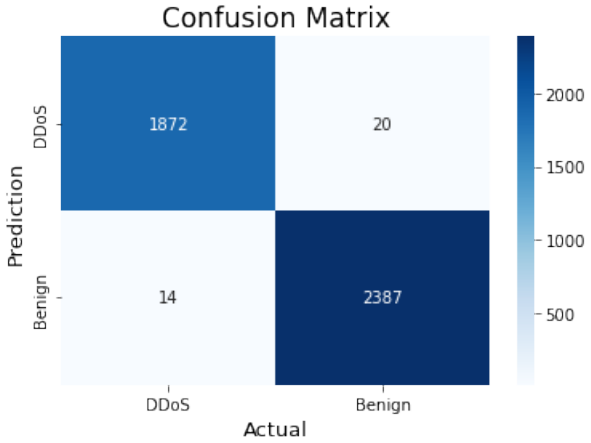

Figure 4.38: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using 10 optimal features

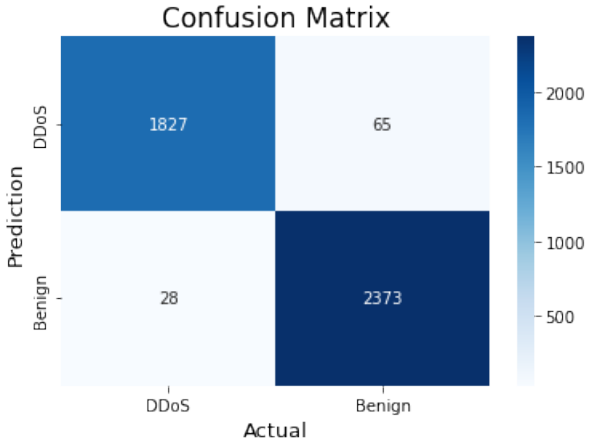

Figure 4.39: Confusion matrix of the ensemble model of KNN, SVM and NB using 10 optimal features

Among the traditional algorithms, SVM is performing the best in case of optimal features (with 11 features) with 98.89% precision and 99.15% recall, showed in Table 4.9. Other traditional algorithms KNN, DT and NB have 0.9882; 0.9736, 0.9864; 0.9598 and 0.9465; 0.7849 in precision and recall respectively.

Fig. 4.43 reflects that ensemble methods such as Voting (KNN, SVM, DT), Voting (KNN, SVM, NB, DT), Voting (KNN, SVM, NB), Voting (SVM, NB, DT) and Voting (KNN, NB, DT) scored 0.9940, 0.9846, 0.9832, 0.9814 and 0.973 in terms of accuracy. Furthermore, most of the traditional algorithms perform well, except NB.

Fig. 4.44 shows that out of 1892 DDoS instances, the proposed ensemble model classifies

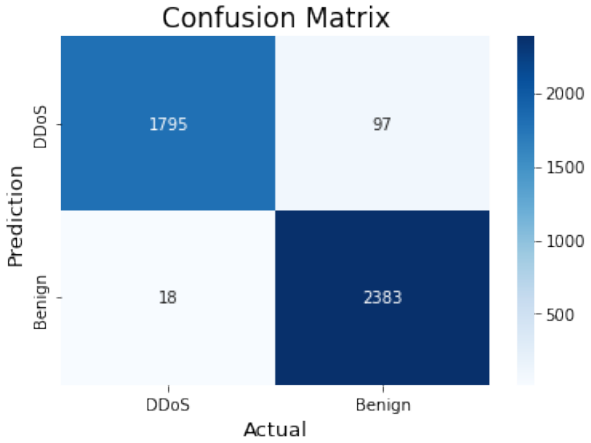

Figure 4.40: Confusion matrix of the ensemble model of KNN, NB and DT using 10 optimal features

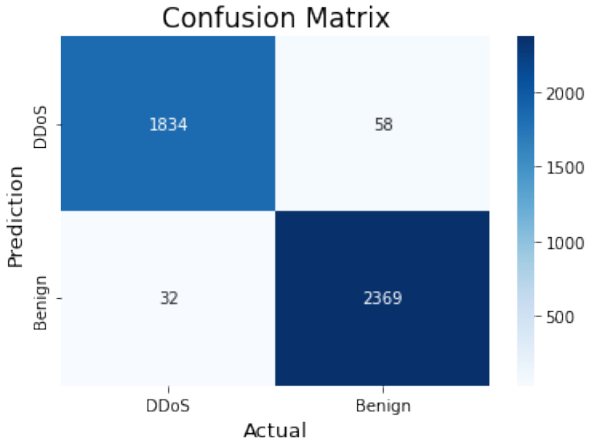

Figure 4.41: Confusion matrix of the ensemble model of SVM, NB and DT using 10 optimal features

1877 of them correctly. In addition, as for the Benign class, the model classifies 11 out of 2401 entities inaccurately.

The combined model of KNN, SVM and NB accurately classifies 1837 out of 1892, or 97.09%. In addition, the model correctly classifies 2384 occurrences of 2401 in Figure 4.45.

Out of 1892 DDoS occurrences, Fig. 4.46 demonstrates that 1795 of them are accurately classified by the ensemble model consisting of KNN, NB, and DT. Furthermore, 19 out of 2401 elements in the Benign class are incorrectly classified by the model.

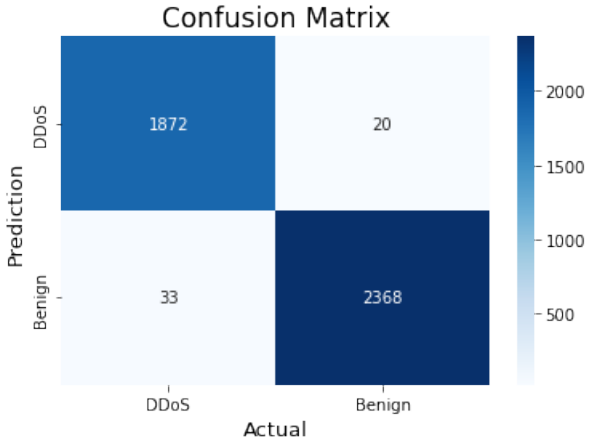

Figure 4.42: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using 10 optimal features

Table 4.9: Performance of the models using Optimal Features (11 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.9832	0.9882	0.9736	0.9808
SVM	0.9914	0.9889	0.9915	0.9902
DT	0.9765	0.9864	0.9598	0.9729
NB	0.8856	0.9465	0.7849	0.8581
Voting (KNN, SVM, DT)	0.9940	0.9942	0.9921	0.9931
Voting (KNN, SVM, NB)	0.9832	0.9908	0.9709	0.9808
Voting (KNN, NB, DT)	0.9730	0.9895	0.9487	0.9687
Voting (SVM, NB, DT)	0.9814	0.9876	0.9699	0.9787
Voting (KNN, SVM, NB, DT)	0.9814	0.9876	0.9699	0.9787

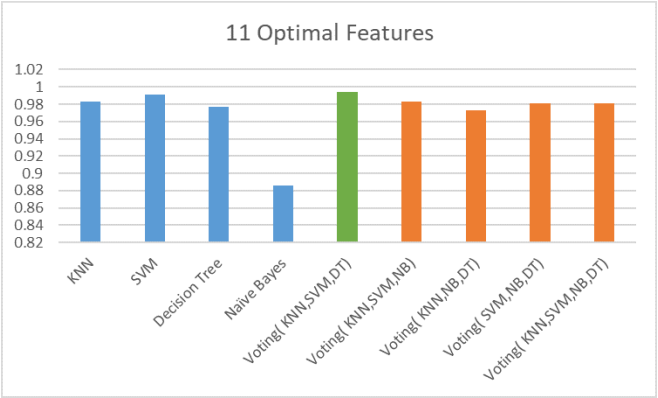

Figure 4.43: Accuracy of the models using optimal features (11 features)

The ensemble model of SVM, NB and DT classifies 1835 of 1892 i.e. 96.99% correctly. Furthermore, the model classifies 2378 instances of 2401 accurately in Fig. 4.47.

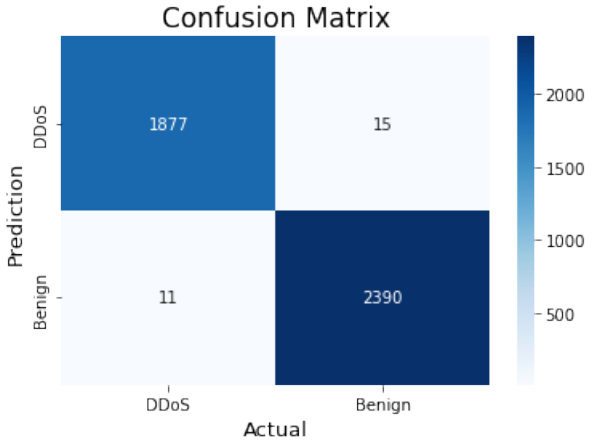

Figure 4.44: Confusion matrix of the proposed ensemble model (KNN, SVM, DT) using 11 optimal features

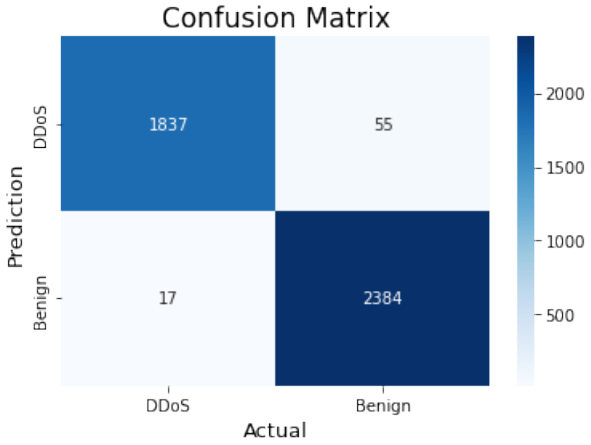

Figure 4.45: Confusion matrix of the ensemble model of KNN, SVM and NB using 11 optimal features

Figure 4.48 demonstrates that 1879 out of 1892 DDoS occurrences are accurately classified by the ensemble model including KNN, SVM, NB and DT. Furthermore, the model incorrectly classifies 27 out of 2401 instances in the Benign class.

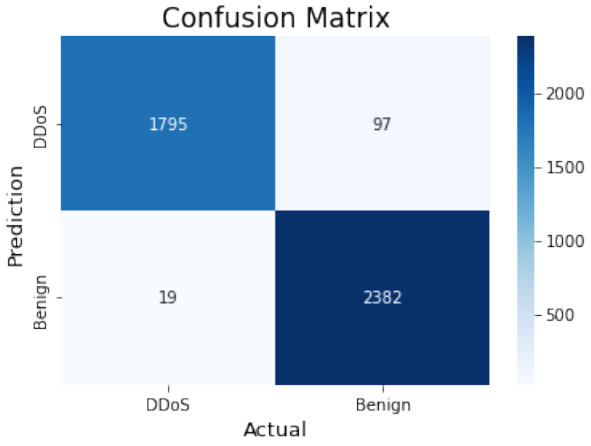

Figure 4.46: Confusion matrix of the ensemble model of KNN, NB and DT using 11 optimal features

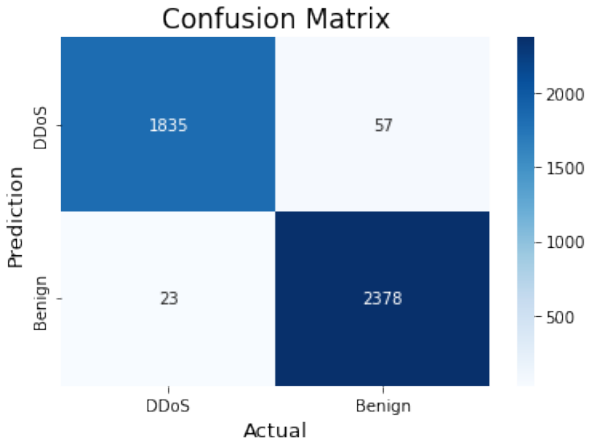

Figure 4.47: Confusion matrix of the ensemble model of SVM, NB and DT using 11 optimal features

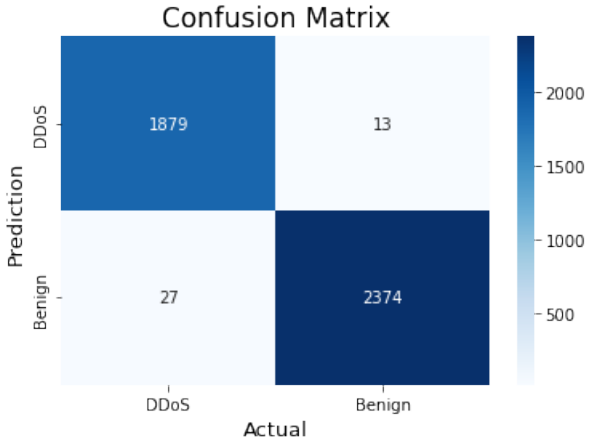

Figure 4.48: Confusion matrix of the ensemble model of KNN, SVM, NB and DT using 11 optimal features

4.4 Discussion

From Table 4.10 it is seen that the accuracy by KNN is 0.4428 and accuracy by SVM is 0.5593. Accuracy given by DT, NB, Voting (KNN, SVM, DT), Bagging and Boosting is same, that is 0.5595. If recall are taken into consideration, then KNN scores the highest with 1.0000 and F1-score of 0.6127. In terms of precision, SVM gives the lowest score that is 0.0000 and KNN gives second lowest which is 0.4416. Highest precision is given by DT, NB, Voting (KNN, SVM, DT), Bagging and Boosting which is 1.000.

Table 4.10: Performance of the models using RFE (10 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.4428	0.4416	1.0000	0.6127
SVM	0.5593	0.0000	0.0000	0.0000
DT	0.5595	1.000	0.0005	0.0011
NB	0.5595	1.000	0.0005	0.0011
Voting (KNN, SVM, DT)	0.5595	1.000	0.0005	0.0011
Bagging	0.5595	1.000	0.0005	0.0011
Boosting	0.5595	1.000	0.0005	0.0011

It is shown in Table 4.11 that the accuracy by SVM is 0.5593 and the accuracy by KNN is 0.4438. The same accuracy, 0.5595, is provided by DT, NB, Voting (KNN, SVM, DT), Bagging, and Boosting. With a recall score of 1.0000 and an F1-score of 0.6137, KNN receives the highest rating. SVM and KNN have the lowest and second-lowest precision scores, respectively, at 0.0000 and 0.4426. DT, NB, Voting (KNN, SVM, DT), Bagging, and Boosting have the highest precision, which is 1.000.

Table 4.11: Performance of the models using RFE (11 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.4438	0.4426	1.0000	0.6137
SVM	0.5593	0.0000	0.0000	0.0000
DT	0.5595	1.000	0.0005	0.0011
NB	0.5595	1.000	0.0005	0.0011
Voting (KNN, SVM, DT)	0.5595	1.000	0.0005	0.0011
Bagging	0.5595	1.000	0.0005	0.0011
Boosting	0.5595	1.000	0.0005	0.0011

From Table 4.12 it is seen that the highest accuracy is given by SVM that is 0.9292 and the second-highest accuracy is 0.8465 which is given by voting(KNN,SVM,DT). In terms of precision, the highest score is given by NB that is 0.9531. The highest recall is 0.9915

and the highest f1-score 0.9902 that is given by SVM. The lowest f1-score is 0.7437 that is given by bagging.

Table 4.12: Performance of the models using chi square (10 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.8463	0.7512	0.9736	0.8481
SVM	0.9292	0.9682	0.9915	0.9902
DT	0.7249	0.6305	0.9080	0.7442
NB	0.8428	0.9531	0.6765	0.7913
Voting (KNN, SVM, DT)	0.8465	0.7549	0.9651	0.8471
Bagging	0.7209	0.6247	0.9186	0.7437
Boosting	0.7354	0.6410	0.9080	0.7515

Table 4.13 shows that voting (KNN,SVM,DT) provides the second-highest accuracy, 0.8619, while SVM gives the highest accuracy, 0.9485. With a precision score of 0.9659, NB has the highest score. The highest f1-score given by the SVM is 0.9391. In terms of recall, DT gives the highest score that is 0.9925.

Table 4.13: Performance of the models using chi square (11 features)

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
KNN	0.8535	0.7579	0.9810	0.8551
SVM	0.9485	0.9810	0.9006	0.9391
DT	0.7529	0.6478	0.9925	0.7744
NB	0.9154	0.9659	0.8377	0.8973
Voting (KNN, SVM, DT)	0.8619	0.7646	0.9921	0.8636
Bagging	0.7309	0.6347	0.9286	0.7537
Boosting	0.7827	0.6827	0.9471	0.7934

Table 4.14 shows the performance of three feature selection techniques (ANOVA, mutual information, and feature importance) and three ensemble learning techniques (bagging, boosting, and voting) on a classification task with 10 and 11 features. The performance is measured by the accuracy of a machine learning model trained on the selected features. In all the cases, voting performs the best out of bagging, boosting and voting.

Table 4.14: Performance of the models using bagging, boosting & voting (10 and 11 features)

Feature Selection Technique	**Bagging accuracy**	**Boosting accuracy**	**Voting accuracy (KNN, SVM, DT)**
ANOVA (10 Features)	0.9415	0.9238	0.9443
ANOVA (11 Features)	0.9439	0.9259	0.9499
Mutual Information (10 Features)	0.9676	0.9043	0.9837
Mutual Information (11 Features)	0.9660	0.9033	0.9856
Feature Importance (10 Features)	0.9855	0.9623	0.9860
Feature Importance (11 Features)	0.9865	0.9557	0.9886
Optimal (10 Features)	0.9112	0.8993	0.9921
Optimal (11 Features)	0.9156	0.9034	0.9940

Bagging and boosting tends to work well when the base models are unstable or have high variance. If the base models are already strong and low in variance, bagging and boosting might not provide significant improvement.

In voting, multiple models are trained independently on the same dataset. Then, the predictions from each model are combined using a ”voting” mechanism. In our proposed model hard voting is applied which depends on majority. As the base models of our proposed one are stable, so it gave higher accuracy by voting rather than bagging and boosting.

Table 4.15 shows the performance of four feature selection techniques on a dataset with 9, 10, 11, and 12 features. The techniques are ANOVA, mutual information, feature importance, and optimal features. The performance is measured by the accuracy of a machine learning model trained on the selected features.

As the number of features increases, the performance of mutual information and feature importance improves. However, with 11 features, ANOVA gives the accuracy of 0.9499

Table 4.15: Performance of the models using no of features

Feature Selection Technique	**9 Features**	**10 Features**	**11 Features**	**12 Features**
ANOVA	0.9430	0.9443	0.9499	0.9489
Mutual Information	0.9812	0.9837	0.9856	0.9858
Feature Importance	0.9838	0.9860	0.9886	0.9898
Optimal Features	0.9899	0.9921	0.9940	0.9934

and 0.9489 with 12 features, which is less than 11 features. In case of optimal features, with 11 features, the accuracy is 0.9940 and with 12 features it is 0.9934. For this reason, 10 and 11 features have been taken.

ANOVA is a powerful tool for comparing means in multiple groups, but it assumes that the variances within each group are approximately equal and that the data are normally distributed. For our dataset, the variance of the 11 features are not equal to that of the 12 features and that might be the reason why the accuracy is lower in case of 12 features compared to 11 features'. In case of optimal features, domain knowledge is used. If 12 features are selected, the accuracy never outperforms the accuracy when 11 features are selected for our dataset. For this reason, 10 and 11 features have been worked on in this research.

4.5 Summary of Results

- The optimal features, that have been chosen from the dataset using domain knowledge, get us the highest accuracy in case of both 10 and 11 features.
- The feature importance technique is the best feature selection technique for the dataset used in this work.
- The ensemble model of KNN, SVM and DT have performed the best in every feature selection technique among the five ensemble models.

CHAPTER 5

CONCLUSIONS

5.1 Summary

Both businesses and non-profits of all sizes, DDoS attacks carry a serious risk. Businesses and organizations can lessen their chance of becoming the target of a DDoS attack by adopting precautions for their own safety. The current dataset has 42 features and freely available in Kaggle. Optimal features have been selected from the dataset using domain knowledge. 10 and 11 number of optimal features have been chosen to get higher accuracy rate and with 11 optimal features, the highest accuracy has been found which is 99.4%.

We introduced a DDoS attack detection system in this work that uses machine learning methods to identify the attack. We applied hard ensemble voting technique to detect the attack with more accuracy and fewer features. To select the features, three feature selection techniques have been applied. By analyzing our result, we can conclude that DDoS attack can be detected with more accuracy by feature importance feature selection technique. Moreover, Four ensemble models using voting have been implemented and among them the combination of KNN, SVM and DT performs better than the other algorithms. The top 10 or 11 features can be utilized for classifying DDoS attacks using our proposed ensemble model. Therefore, through analyzing the features, it would be possible to determine whether there is a chance of a DDoS attack if it is possible to obtain the values of the same features in real time.

5.2 Limitations and Future Work

Although the recommended ensemble model outperforms the traditional algorithms, there are certain issues that were found throughout the thesis research. They will help to assess the algorithm's potential for future improvement. A list of these limitations is as follows:

- Only the traditional and ensemble classifier algorithms of machine learning have been applied. No deep learning model has been used.
- The results obtained from one dataset may not be fully applicable to different scenarios.
- Only four machine learning algorithms have been used in ensemble hard voting technique. More traditional algorithms can be put in use to implement ensemble method.

There are a number of areas where future work might be done to improve the ensemble model and give extra performance evaluation. Those are as follows:

- The proposed ensemble model can be applied in different dataset to get the robustness of the thesis work.
- Three feature selection techniques have been applied to choose 10 and 11 number of features. More feature selection techniques could be applied to detect various number of features so that more analysis could be done on the results.
- A new ensemble model could be developed from multiple traditional ML models to make the prediction more reliable.
- The DDoS attack detection system could be implemented in real time to reduce the attack in a huge scale.

REFERENCES

Akhtar, M. S., & Feng, T. (2022). Comparison of classification model for the detection of cyber-attack using ensemble learning models. *EAI Endorsed Transactions on Scalable Information Systems*, *9*(5).

Applications of machine learning [Accessed on November 10, 2023]. (2023).

Atif, M., Anwer, F., & Talib, F. (2022). An ensemble learning approach for effective prediction of diabetes mellitus using hard voting classifier. *Indian Journal of Science and Technology*, *15*(39), 1978–1986.

AWS Shield Threat Landscape Report – Q1 2020. (2020). https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf

Azhar, M., Ullah, S., Ullah, K., Shah, H., Namoun, A., & Rahman, K. U. (2023). A three-dimensional real-time gait-based age detection system using machine learning. *CMC-COMPUTERS MATERIALS & CONTINUA*, *75*(1), 165–182.

Azmi, M. A. H., Foozy, C. F. M., Sukri, K. A. M., Abdullah, N. A., Hamid, I. R. A., & Amnur, H. (2021). Feature selection approach to detect ddos attack using machine learning algorithms. *JOIV: International Journal on Informatics Visualization*, *5*(4), 395–401.

Bagherzadeh, F., Mehrani, M.-J., Basirifard, M., & Roostaei, J. (2021). Comparative study on total nitrogen prediction in wastewater treatment plant and effect of various feature selection methods on machine learning algorithms performance. *Journal of Water Process Engineering*, *41*, 102033.

Başţanlar, Y., & Özuysal, M. (2014). Introduction to machine learning. *miRNomics: MicroRNA biology and computational analysis*, 105–128.

BBC News. (2020). *Ddos attack takes belgian government sites offline* [Published: June 18, 2020]. https://www.bbc.com/news/technology-53093611

Beulah, M., & Pitchai Manickam, B. (2022). Detection of ddos attack using ensemble machine learning techniques. *Soft Computing for Security Applications: Proceedings of ICSCS 2021*, 889–903.

Bhardwaj, A., Mangat, V., & Vig, R. (2020). Hyperband tuned deep neural network with well posed stacked sparse autoencoder for detection of ddos attacks in cloud. *IEEE Access*, *8*, 181916–181929.

Cook, S. (2023). *Ddos statistics and facts 2023* [Updated: June 13, 2023]. Comparitech. https://www.comparitech.com/blog/information-security/ddos-statistics-facts/

Das, S., Mahfouz, A. M., Venugopal, D., & Shiva, S. (2019). Ddos intrusion detection through machine learning ensemble. *2019 IEEE 19th international conference on software Quality, Reliability and Security Companion (QRS-C)*, 471–477.

Das, S., Venugopal, D., & Shiva, S. (2020). A holistic approach for detecting ddos attacks by using ensemble unsupervised machine learning. *Advances in Information and Communication: Proceedings of the 2020 Future of Information and Communication Conference (FICC), Volume 2*, 721–738.

de Araujo, P. H. H. N., Silva, A., Junior, N. F., Cabrini, F., Santiago, A., Guelfi, A., & Kofuji, S. (2021). Impact of feature selection methods on the classification of ddos attacks using xgboost. *Journal of Communication and Information Systems*, *36*(1), 200–214.

Decision tree: Introduction and example [Accessed on December 10, 2023]. (2023).

Ertan, H. (2020). Which features to use in your model [Published on February 20, 2020. Accessed on December 10, 2023]. *Medium*. https://medium.com/@hertan06/which-features-to-use-in-your-model-350630a1e31c.

Gandhi, R. (2018). Support vector machine: Introduction to machine learning algorithms [Published on June 7, 2018. Accessed on December 10, 2023]. *Towards Data Science*. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

Hashim, M. S., & Yassin, A. A. (2023). Using pearson correlation and mutual information (pc-mi) to select features for accurate breast cancer diagnosis based on a soft voting classifier.

Janiesch, C., Zschech, P., & Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, *31*(3), 685–695.

K-nearest neighbor algorithm for machine learning [Accessed on December 10, 2023]. (Year not specified).

Kabir, M. H., Mahmood, S., Al Shiam, A., Musa Miah, A. S., Shin, J., & Molla, M. K. I. (2023). Investigating feature selection techniques to enhance the performance of eeg-based motor imagery tasks classification. *Mathematics*, *11*(8), 1921.

Karim, A., Shahroz, M., Mustofa, K., Belhaouari, S. B., & Joga, S. R. K. (2023). Phishing detection system through hybrid machine learning based on url. *IEEE Access*, *11*, 36805–36822.

Kaspersky Lab. (2020). *Ddos attacks in q3 2020* [Published: October 28, 2020]. Securelist. https://securelist.com/ddos-attacks-in-q3-2020/99171/

Krishna, R. (2020). Ddos classification.

Kumar, K., & Barver, A. (2021). A ddos attack detection using deep learning-a review. *IJFMR-International Journal For Multidisciplinary Research*, *5*(3).

Kumar, Y. V., & Kamatchi, K. (2020). Anomaly based network intrusion detection using ensemble machine learning technique. *International Journal of Research in Engineering, Science and Management*, *3*, 290–297.

Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, *9*(20), 4396.

Ma, G., Zhang, J., Liu, J., Wang, L., & Yu, Y. (2023). A multi-parameter fusion method for cuffless continuous blood pressure estimation based on electrocardiogram and photoplethysmogram. *Micromachines*, *14*(4), 804.

Machine learning - decision tree classification algorithm [Accessed on December 10, 2023]. (2023).

Machine learning - naive bayes classifier [Accessed on December 10, 2023]. (2023).

Machine learning - support vector machine algorithm [Accessed on December 10, 2023]. (2023).

Machine learning (ml) [Accessed on November 10, 2023]. (2023).

Pierzyna, M., Saathof, R., & Basu, S. (2023). {\Pi}-ml: A dimensional analysis-based machine learning parameterization of optical turbulence in the atmospheric surface layer. *arXiv preprint arXiv:2304.12177*.

Polat, H., Polat, O., & Cetin, A. (2020). Detecting ddos attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability*, *12*(3), 1035.

Samat, N. A. (2022). *Intrusion detection system: Challenges in network security and machine learning* (tech. rep.). EasyChair.

Saravanakumar, G., Naveen, V., Koushik, P., Sneha, C., et al. (2023). A ddos attack categorization and prediction method based on machine learning. *Journal of Population Therapeutics and Clinical Pharmacology*, *30*(9), 300–307.

Savita, T., & Sharma, M. R. (2023). Ddos attack detection using soft voting classifier. *52*(3).

Semi-supervised learning in machine learning [Accessed on December 1, 2023]. (2023).

Söğüt, E., & Erdem, O. A. (2023). A multi-model proposal for classification and detection of ddos attacks on scada systems. *Applied Sciences*, *13*(10), 5993.

Solano, E. S., & Affonso, C. M. (2023). Solar irradiation forecasting using ensemble voting based on machine learning algorithms. *Sustainability*, *15*(10), 7943.

Srivastava, T. (2018). Introduction to k-neighbours algorithm in clustering [Last updated on October 20th, 2023. Accessed on December 10, 2023].

Supervised machine learning [Accessed on November 15, 2023]. (2023).

Support vector machine algorithm [Accessed on December 10, 2023]. (2023).

Tekleselassie, H. (2021). A deep learning approach for ddos attack detection using supervised learning. *MATEC Web of Conferences*, *348*, 01012.

Tikhe, S. A., & Rana, D. P. (2023). Fine-tuned predictive models for forecasting severity level of covid-19 patient using epidemiological data. In *Frontiers of ict in healthcare: Proceedings of eait 2022* (pp. 431–442). Springer.

Tuan, T. A., Long, H. V., Son, L. H., Kumar, R., Priyadarshini, I., & Son, N. T. K. (2020). Performance evaluation of botnet ddos attack detection using machine learning. *Evolutionary Intelligence*, *13*, 283–294.

Unsupervised machine learning [Accessed on December 1, 2023]. (2023).

What is reinforcement learning? [Accessed on December 10, 2023]. (2023).

Yoachimik, O. (2022). *Cloudflare ddos threat report 2022 q3* [Published: October 12, 2022]. Cloudflare. https://blog.cloudflare.com/cloudflare-ddos-threat-report-2022-q3/

Zaini, N. A. M., & Awang, M. K. (2023). Hybrid feature selection algorithm and ensemble stacking for heart disease prediction. *International Journal of Advanced Computer Science and Applications*, *14*(2).