A FRAMEWORK FOR DETECTION OF DDOS ATTACK IN SOCIAL NETWORKS USING DEEP LEARNING APPROACH

A Thesis

Submitted in partial fulfillment of the requirements for the award of the degree of

DOCTOR OF PHILOSOPHY

in

Computer Science Engineering

By

Sagar Dhanraj Pande

Reg. No. 41800173

Supervised By	**Co-Supervised by**
Dr. Aditya Khamparia	**Dr. Deepak Gupta**

LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

2021

Declaration

I hereby declare that the thesis entitled “A FRAMEWORK FOR DETECTION OF DDOS ATTACK IN SOCIAL NETWORKS USING DEEP LEARNING APPROACH” submitted by me for the Degree of Doctor of Philosophy in Computer Science and Engineering is the result of my original and independent research work carried out under the guidance of Supervisors Dr. Aditya Khamparia and Dr. Deepak Gupta, and it has not been submitted for to any university or institute for the award of any degree or diploma.

Date:20/08/2021

Signature

Certificate

This is to certify that the thesis entitled “A FRAMEWORK FOR DETECTION OF DDOS ATTACK IN SOCIAL NETWORKS USING DEEP LEARNING APPROACH” submitted by Sagar Dhanraj Pande for the award of the degree of the Doctor of Philosophy in Computer Science and Engineering, Lovely Professional University Punjab, India is entirely based on the work carried out by him under our supervision and guidance, The work recorded, embodies the original work of the candidate and has not been submitted for the award of any degree, or diploma of any university and institute, according to the best of our knowledge.

Supervisor Signature:

Deepak Gupta

Date:20/08/2021 Co-Supervisor Signature:

Abstract

This research work has been established on the Distributed Denial of Service (DDOS) Attack in social networks using the machine and deep learning approach. Network Security is one of the crucial factors for any organization. If the intrusion detection system is powerful then definitely the reliability of the network will be enhanced. The existing system is not capable of handling the distributed attacks. As the data is increasing rapidly the level of attack and its complexity is also increasing. The majority of the researcher are implementing several machines and deep learning approaches for the detection of the distributed attacks.

In this work, the detection of large and complex network attacks specifically from the perspective of DDOS attacks has been explored. Moreover, the existing techniques used for DDOS attack detection in social networks were also analyzed. We have also analyzed the anomalies present in the various online social networks such as Facebook, Twitter, Google+, and Orkut. After performing an exhaustive study of existing DDOS detection approaches in social networks, the performance of existing techniques needs to be improved. As the behavior and properties of the social network vary with its nature no technique can seldom efficiently enough to address the DDOS attack detection in all kinds of social networks. The performance of existing techniques from a complexity perspective was also affected by the scalability of networks.

The major goals are considered as the part of this research work can be mentioned in certain points such as study the various aspects related to DDOS threats and their impact on both the organizational level and individual level in terms of social networks, exploring the issue related to DDOS threats in terms of analysis aspects, identifying the present existing solution for the effective identifying and classification of various categories of the DDOS threat, identifying the limitations of the present solution and possible ways of generating novel solutions to cover up the existing limitations. Proposing a novel framework that needs to be implemented, and evaluated using certain performance metrics.

In this work, a Deep Learning approach is being implemented with various rigorous hyperparameter tuning. Artificial Neural Network was implemented on the well-known dataset.

In this research work, we have also provided the recommendation on two famous datasets which are KDD and NSL-KDD. The recommendation was done based on the analysis performed using Office 365, Tableau, and Shap techniques. Explainable AI also helps us to identify more impactful features based upon the Shap value.

Evaluation of the proposed model was conducted using various performance measures such as accuracy, precision, and recall it is observed that the method is fairly effective and efficient in the detection of a DDOS attack.

Acknowledgments

First and foremost, on the successful completion of the research work and this thesis, I would like to pay all praise to the almighty God. In this thesis the work presented would not have been possible without my close association with many people. With utmost humbleness, I would like to take this opportunity to extend my deep sense of gratitude to all those who made this research thesis possible. First and foremost, I am indebted to my supervisor Dr. Aditya Khamparia, Associate Professor, School of Computer Science, Lovely Professional University, Phagwara Punjab, for his dedicated support, timely advice, inspiration, encouragement, and continuous support throughout my Ph.D. I gratefully acknowledge my co-supervisor Dr. Deepak Gupta, Assistant Professor Department of Computer Science and Engineering, Maharaja Agrasen Institute of Technology (MAIT), under whom supervision I had started my research work. I am grateful to my parents who always inspired me to do this work. Their affection and love can’t be expressed in words. As always it is not possible to mention everybody who had a constructive impact on this work, directly and indirectly, however, there are those whose support is even more important, and I extend my thanks to all those people.

Sagar Dhanraj Pande

(41800173)

Contents

Declaration……………………………………………………………………………………ii

Certificate………………………………………………………………………………….iii

Abstract……………………………………………………………………………………….iv

Acknowledgments………………………………………………………………………………vi

Contents…………………………………………………………………………………………vii

List of Figures……………………………………………………………………………….....x

List of Tables………………………………………………………………………………………xii

Glossary………………………………………………………………………………….....xiii

Chapter 1 Introduction……………………………………………………………..1

1.1 Overview……………………………………………………………………...1

1.2 Scope of Research……………………………………………………………7

1.3 Main Goals……………………………………………………………………7

1.4 Research Contribution………………………………………………………...8

1.5 Motivation……………………………………………………………………9

1.6 Problem Statement……………………………………………………………9

1.7 Research Objectives…………………………………………………………..10

1.8 Research Questions…………………………………………………………...10

1.9 Research Gap…………………………………………………………………11

1.10 Thesis Organization…………………………………………………………..11

1.11 Summary……………………………………………………………………...12

Chapter 2 Basic Concepts and DDOS Attack Background……………………...13

2.1 Overview……………………………………………………………………13

2.2 Discussion on Security Aspects……………………………………………...13

2.3 DDOS Attack………………………………………………………………..14

2.4 History of DDOS……………………………………………………………15

2.5 Various DDOS Tools………………………………………………………..17

2.6 Related Work………………………………………………………………..18

2.7 Comparative Study on Existing Methodologies………………………… 23

2.8 Summary……………………………………………………………….28

Chapter 3 Methods and Material……………………………………………....29

3.1 Overview……………………………………………………………….29

3.2 Data Information………………………………………………………….29

3.3 Methodologies…………………………………………………………….33

3.3.1 Preprocessing…………………………………………………………33

3.3.2 Feature Selection Methodologies…………………………………….34

3.3.2.1 Principal Component Analysis(PCA)………………………. 35

3.3.2.2 Linear Discriminant Analysis(LDA)…………………………35

3.3.2.3 Recursive Feature Elimination(RFE)…………………………36

3.3.2.4 Univariate Feature Elimination………………………………36

3.3.2.5 Proposed Methodology for Feature Selection as a Stack…… 37

3.3.2.6 Genetic Algorithm(GA)………………………………………38

3.3.3 Classification Methodologies…………………………………………40

3.3.3.1 K-nearest Neighbor(KNN)……………………………………40

3.3.3.2 Logistic Regression…………………………………………...41

3.3.3.3 Support vector Machine(SVM)………………………………42

3.3.3.4 Decision Tree…………………………………………………42

3.3.3.5 Random Forest………………………………………………..42

3.3.3.6 XGBoost……………………………………………………...43

3.3.3.7 Stochastic Gradient Descent(SGD)…………………………..44

3.3.3.8 Perceptron……………………………………………………44

3.3.3.9 Artificial Neural Network…………………………………… 45

3.3.3.10 Explainable AI………………………………………………..47

3.3.3.11 Explainable AI-Based Proposed Model………………………50

3.4 Performance Evaluation Metrics…………………………………………52

3.5 Summary……………………………………………………………….54

Chapter 4 Experimental Analysis…………………………………………...55

4.1 Overview……………………………………………………………….55

4.2 Analysis of NSL-KDD Dataset…………………………………………...55

4.2.1 Recommendations Based on Analysis……………………………………66

4.3 Machine Learning-Based Framework Along with Feature Selection Results68

4.4 ANN Based Framework Results………………………………………… 77

4.5 Explainable AI-Based results…………………………………………… 79

4.6 Summary…………………………………………………………………..82

Chapter 5 Conclusion and Future Works……………………………………… 83

Chapter 6 Bibliography……………………………………………………….86

List of Research Publications…………………………………………………..102

List of Figures

Chapter No.	Figure	Page
1.1	Effective DDOS assault operation	3
1.2	Various Categories of DDOS Threats	4
1.3	DDOS Threat on GITHUB (Right Graph) and Representation of the Size of DDOS Threats During 2008 – 2017	4
1.4	Anomaly categories	5
2.1	CIA Triad	14
3.1	Stacked Based Feature Selection	38
3.2	The Genetic Algorithm Flowchart	40
3.3	Rough Representation of ANN	45
3.4	Demonstration of working of ANN	46
3.5	Generalized SHAP System	49
3.6	The Proposed Framework Flowchart Overview	51
4.1	Graphical comparison Among the Various Forms of KDD Dataset	56
4.2	DDOS Vs Normal Activities Distribution for NSL-KDD & KDD Datasets	57
4.3	Distribution of DDOS and Normal Activities in NSL-KDD Dataset	58
4.4	The Distribution of Proportions of DDOS and Normal Activities in the KDD Dataset	58
4.5	Distribution of DDOS and Normal Activities As Per the Various Flags in NSL-KDD	59
4.6	Distribution of various classes in NSL-KDD Dataset	60
4.7	Distribution of various flags in NSL-KDD Dataset	61

4.8	Distribution of various protocols in NSL-KDD Dataset	62
4.9	Distribution of various services in NSL-KDD Dataset	63
4.10	Distribution of classes concerning protocol in the NSL-KDD dataset	64
4.11	Distribution of services concerning protocol in the NSL-KDD dataset	65
4.12	Distribution of services concerning classes in the NSL-KDD Dataset	66
4.13	Plot of Correlation Matrix	69
4.14	Features Selected Using Univariate Features Selection Technique	69
4.15	Dataset Dimensions	71
4.16	Labeled Distribution of Training Data	71
4.17	Labeled Distribution of Test Data	72
4.18	Training Dataset First 5 Instances	72
4.19	Testing Dataset First 5 Instances	73
4.20	Model Training and Validation	73
4.21	Accuracies With 17 Features	75
4.22	Accuracies With 11 Features	75
4.23	Accuracies of Classifiers By Using Different Feature Selection Techniques	76
4.24	The Distribution of SHAP Values	81
4.25	Interpretation of the Deep Neural Network Classifier	81
4.26	Top 20 Attributes of DoS Attack	82

List of Tables

Chapter No.	Table	Page
2.1	History of DDOS Threats	15
2.2	Comparative study over the collected research articles	23
3.1	Breakdown of Different Sub-classes of Each Attack in NSL-KDD	30
3.2	The Distribution Breakdown of Various Threats on NSL-KDD Dataset	32
4.1	Summary of the KDD & NSL-KDD Dataset	56
4.2	Analysis of Protocol Attack	67
4.3	Analysis of Various Services	67
4.4	Analysis of Various Flags	68
4.5	Accuracies Using K-Fold Cross-Validation	74
4.6	Accuracy Comparison With Other Techniques	74
4.7	Results Obtained Using Various Machine Learning Techniques	77
4.8	Results Obtained Using ANN on KDD Dataset	78
4.9	Result Obtained Using ANN on NSL-KDD Dataset	79

Glossary

DDOS	Distributed Denial-of-Service
DOS	Denial-of-Service
R2L	Remote to Local
U2R	User to Root
IT	Information Technology
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
RFE	Recursive Feature Elimination
SVM	Support Vector Machine
KNN	K-Nearest Neighbour
SGD	Stochastic Gradient Descent
ANN	Artificial Neural Network
GA	Genetic Algorithm
SHAP	Shapley Additive exPlantions
IDS	Intrusion Detection System
Explainable AI	Explainable Artificial Intelligence
ML	Machine Learning
WSN	Wireless Sensor Networks

Chapter 1

Introduction

1.1 Overview

With the increasing necessity to utilize the Internet in both technologies and sectors, the volume of movement of information packets and the demand on the networks have enhanced in recent years. As a result, despite the availability of many internet security systems, such as the firewall system, which provides outstanding defense and prevention, the most sensitive information remains vulnerable. The firewall systems prevent unauthorized entry to the systems after the data has been transferred, but they are unable to detect the monitoring. It will be unable to identify any attack that tried to breach it. To keep the network within surveillance, it should be secured by an IDS[1]. The intrusion is characterized as a threat to resource confidentiality, integrity, and availability triggered perhaps by authorized network operators abusing specific privileges or by unauthorized network operators being able to utilize the application through specific holes[2]. There are two types of IDSs depending on the detection methodologies[3]: signature-based IDS and anomaly-based IDS. Since the signature-based approach relies on pre-stored principles as a basis for defining known types of attack, the non-stored attack was missed. Although the phenomenon creates a reference file recording the networks' activities or routine activity, any abnormality resulting from these activities is evidence of the existence of an attack[4, 5]. A large number of characteristics of network information are repetitive and useless. Examining both features, which is one of the most difficult problems, often requires time. As a result, using all of the IDS functionality isn't essential. Furthermore, such functionality harms the detection network's performance. As a result, it tries to choose features that have a positive impact on efficiency[6, 7].

The system of identifying activities aimed at gaining unlawful entry to a computing machine is known as intrusive detection. An attacker may come from the outside or the inside. The above refers to network providers that have a certain level of permission. That being said, they proceed to misuse their legitimate access credentials to obtain non-authorized access. The first group of controllers is endeavoring to gain unlawful entry to machine description [8]. Various network or host information forms, such as packages with headers, telephone services, document mechanism updates, and statistics on mobility, are gathered and analyzed by a framework for threat detection. In the intrusion detection method, the analysis methodology provides the data gathered by the networks. This methodology tries to examine the data to find inconsistencies and invasions. The IDS device sends a warming document to the controller of the system if it identifies a threat[9].

The suggested architecture is primarily DDOS-based. The DDoS attack is one of the most powerful pieces of network weaponry. It means that hackers attempted to make a platform or system unavailable by overburdening or crashing them with heavy traffic. The DDOS threat can be portrayed in an efficient model as seen in Figure 1.1. Distributed Denial of Service (DDoS) attacks target server connectivity, taking advantage of the advantages of gaining access to various assault source computers. A DDOS attack is described as gaining access to multiple networks utilizing command and control (C & C) and attempting denial of service from the side of the user. The intruder’s primary objective is to pull the device or server down by submitting frequent requests for information. DDOS uses the Command and Control (C & C) approach to achieve access to applications and servers. Botnet or Zombie networks are those that are influenced by C & C. Hacker is the individual in charge of the botnet programs. The organizational workings of a DDOS attack are depicted in Figure 1.1.

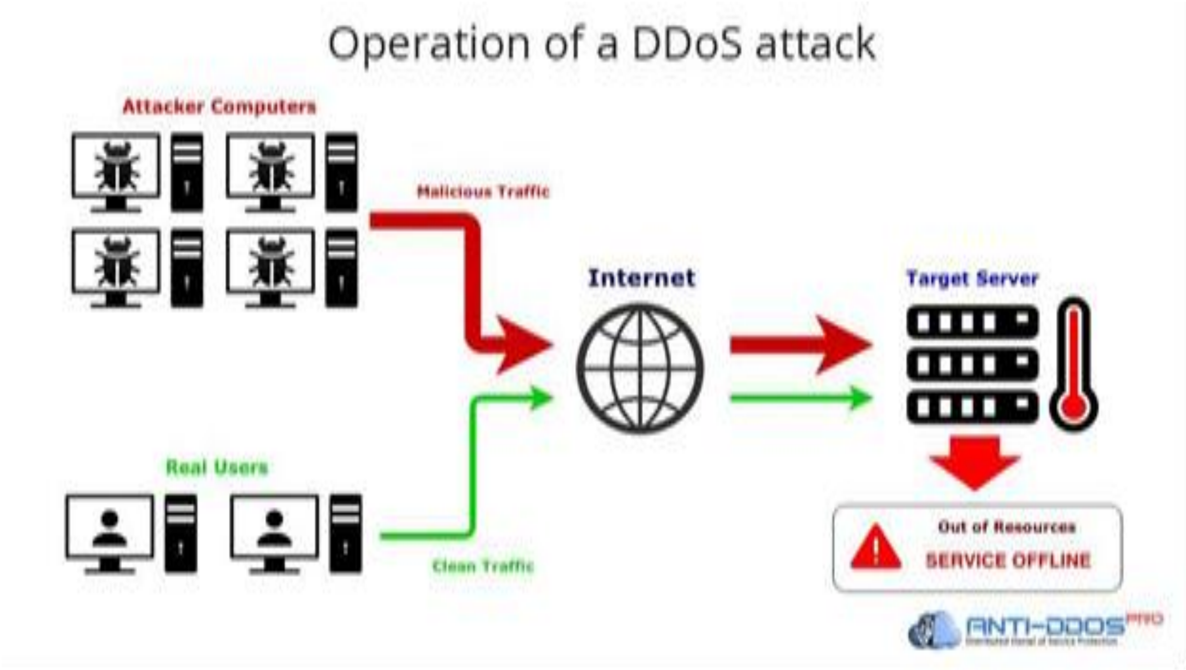


Figure 1.1 Effective DDOS assault operation [10]

Various forms of DDOS network attacks, as well as their prevalence, are depicted in Figure 1.2. The intruder's main goal is to load up reserves and wreck them. The main goal of an HTTP flood DDOS attack is to generate assault traffic that faithfully imitates the legitimacy of a human client. As a result, it becomes more difficult for an accidental casualty to distinguish between legitimate and attack traffic. The server becomes unavailable to authentic clients as a result of such an attack. Figure 1.3 depicts recent assaults on Github as well as the magnitude of the risks.

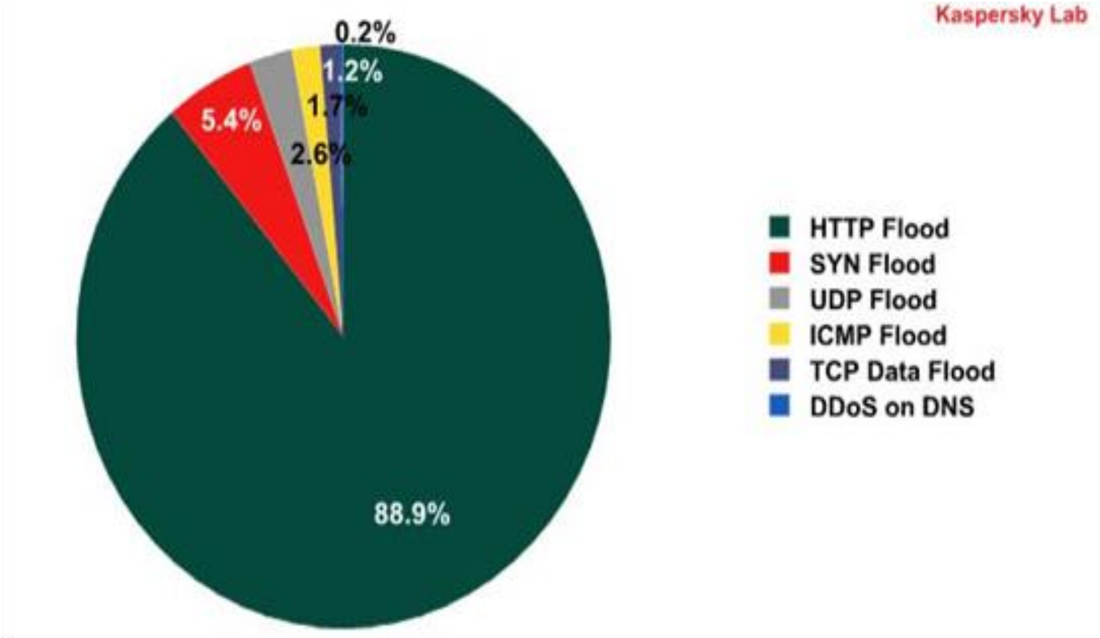


Figure 1.2 Various Categories of DDOS Threats[11]

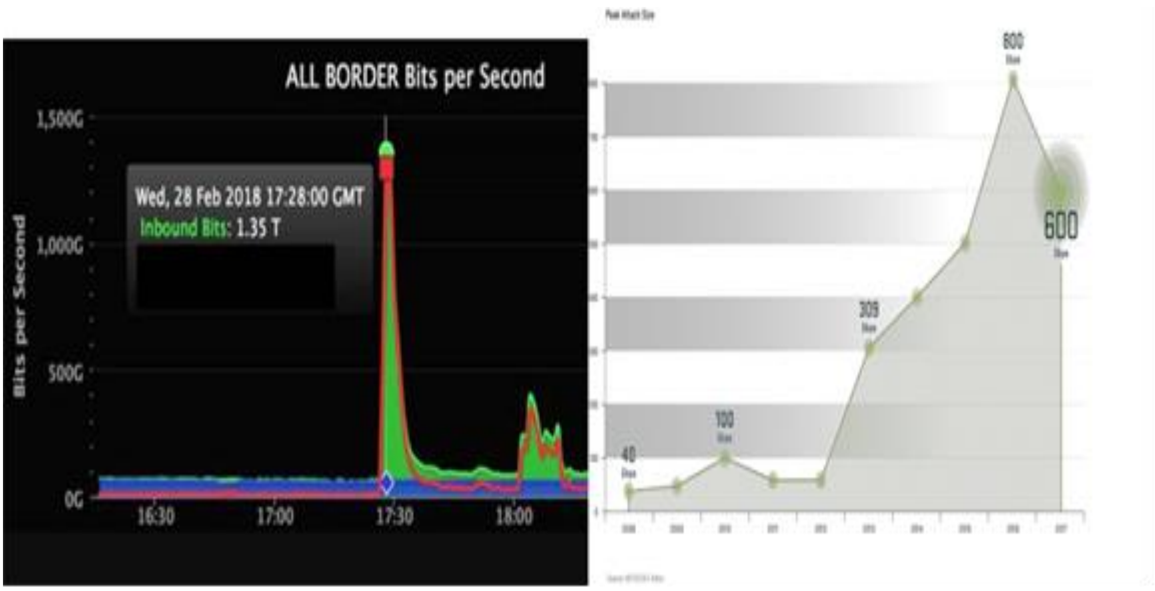


Figure 1.3 DDOS Threat on GITHUB (Right Graph), Representation of the Size of DDOS Threats During 2008 – 2017 [12, 13]

Anomaly identification techniques are used explicitly or legitimately to track deviations from what is expected or expected. Consider the following scenario: a robber breaks into your house; your neighbors will spot a potential intruder by observing the suspicious behavior of a stranger near your home. Different programs can provide phenomenal instances using anomaly detection techniques such as grouping. Banks can detect counterfeiting activity by looking at unusual spending patterns. Various applications in anomaly identification are bank fraud identification, identification of fraud on mobile cellular networks, identification of insurance fraud, and identification of healthcare.

Anomalies can be divided into three categories and they are local discrepancies, situational discrepancies, and aggregate discrepancies. This categorization of classes is mentioned in the following fig. 1.4.

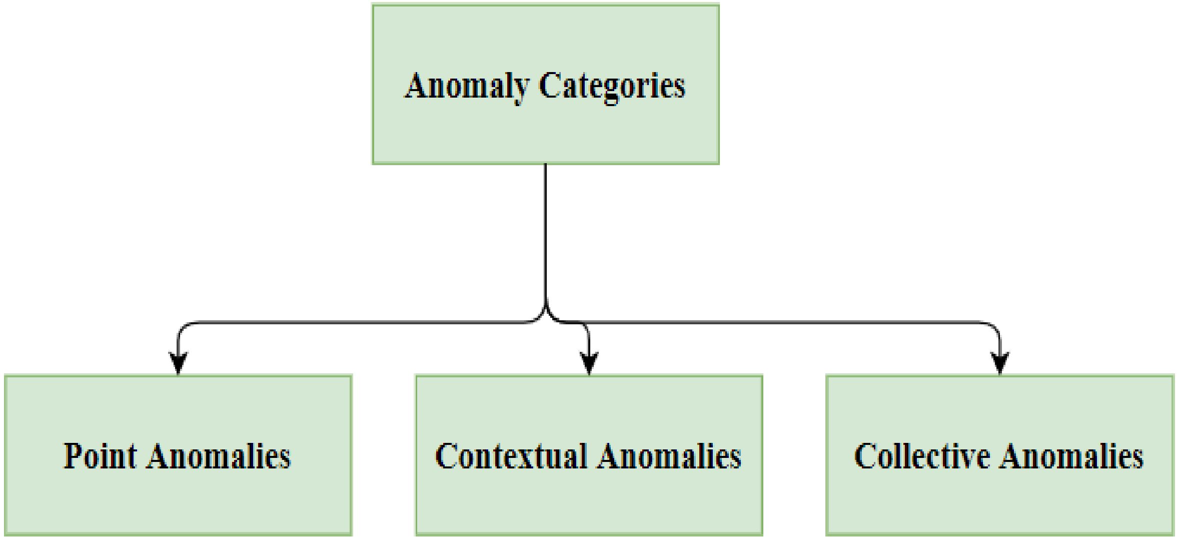


Figure 1.4 Anomaly categories

Point Anomalies: It refers to the ability to differentiate an unusual information event from the rest of the information. It’s also called a single information example that’s out of the ordinary in terms of the information[14].

Contextual Anomalies: It is also called conditional discrepancies, which refers to a knowledge event that seems strange in one context but not in another. It may be described as an information instance that is abnormal within a context. For instance, although a temperature drop is unusual in many countries, it is common in others. Contextual anomalies are knowledge occurrences that are distinguished by conceptual and social characteristics. Contextual abnormalities have mostly been studied in time-arrangement knowledge[14].

Collective Anomalies: It refers to detecting anomaly approaches that mark a collection of data as abnormal in terms of the whole set of data. If there are some abnormal information cases, it may be referred to. Under the case where the knowledge events do not appear together as a group, they may or may not be discrepancies[14-15]. The focus of the investigation is on the local discrepancy. This has resulted in a variety of techniques being offered. Detecting anomalies can be categorized into three parts such as supervised techniques, semi-supervised techniques, and unsupervised techniques[14].

Supervised Anomalies Identification Approaches: It's difficult to come up with indistinguishable and precise names, especially for the abnormality category. Engineered discrepancies have been integrated into a typical data collection that has been suggested by a small number of researchers.

Semi-supervised Anomalies Identification Approaches: The semi-modulated oddity location approaches acknowledge that only instances for the typical group are included in the preparatory subsets that have been named. In the case of shuttle criticize identification, for instance, a malfunction reveals an abnormality that is difficult to illustrate. These systems are remarkable because it is difficult to find a preparation dataset that includes all possible outcomes of an unusual practice.

Unsupervised Anomalies Identification Approaches: In the unsupervised methodology, abnormality position methodologies do not need to worry about information planning. This sort of plan is extensively utilized, legitimately presumes ordinary examples to be discrepancies in the information are more prevalent than discrepancies in the information. If this assumption is not accurate, the false warning rate would be increased in these grouping-based approaches. Ordinary knowledge instances are expected to fall into a category; anomalies, on the other hand, do not fit into either category and appear as anomalies. The second gathering anticipates that, while normal knowledge situations will be similar to their nearest party centroid, anomalies will be far apart. The third category expects knowledge manifestations to be If they fit in huge, thick groups, they are typical; otherwise, they are aberrant, if they tiny enough to squeeze in, insufficient groups. The closest neighbor strategies are predicated on the premise that regular information examples exist. exist nearer to one another, while discrepancies occur in places unavailable to their closest acquaintances. These structures necessitate an Isolation or proximity metric that is defined as the distance between two information events.

1.2 Scope of Research

A DDoS attack is a major security issue for the IT sector. Hundreds of thousands of unprotected computers will quickly overload victims' websites by blocking legitimate traffic. DDoS attacks are network-based threatening activities that drain network bandwidth or exhaust the user's energy. Several specific security indicators are used to counteract a DDoS attack. Various researchers have proposed mechanisms to combat DDOS threats such as stopping the assault, tracking back, the response of the assault, identification of the assault, and description of the assault, but DDOS risks continue to grow year after year. Rather than a specific solution to combat the previous and potential shortcomings of DDoS attacks, a holistic solution for DDoS assaults is needed. To build such a solution, all of the elements that could allow hackers to cause zombies and launch a DDoS attack must be comprehended. The best approach to this conundrum so far has been ignored. Specific signature and anomaly-based approaches for detecting DDoS risks have been applied in the past, but only a handful of them have focused on the presence of irregularities. Most recognition strategies do not have accurate real-time recognition, despite having a high detection accuracy and fewer mistakes. The focus of the research on this topic was primarily on the definition of known threats to provide learners with a better understanding of DDOS assaults.

1.3 Main Goals

The below are the key goals that I asserted during the course of my research::

1. Study the various aspects related to DDOS threats and their impact on both the organizational level and individual level in terms of social networks.
2. Exploring the issue related to DDOS threats in terms of analysis aspects.
3. Identifying the present existing solution for the effective identifying and classification of various categories of the DDOS threat.

4. Identifying the limitations of the present solution and possible ways of generating novel solutions to cover up the existing limitations.
5. Proposing a novel framework that needs to be implemented, and evaluated using certain performance metrics.

1.4 Research Contribution

The major research contribution involved in the proposed research aspect can be mentioned as follows:

1. Reviewed the various methodologies for the detection of the DDOS threats, especially focused on machine learning and deep learning methodologies.
2. Primary features have been selected through various methodologies such as correlation-based feature selection, principal component analysis(PCA), linear discriminant analysis(LDA), recursive feature elimination(RFE), and univariate feature selection. And, a novel feature selection was introduced by combining the above feature selection methodologies as a stack.
3. By using the generated novel framework for feature selection along with various classification methodologies such as support vector machine(SVM), k-nearest neighbor(KNN), stochastic gradient descent(SGD), XGBoost, and perceptron were implemented and compared the results among these classifiers.
4. A novel framework based on a customized artificial neural network(ANN) methodology along with a genetic algorithm(GA) is introduced for the identification of various DDOS threats. This framework performance has been compared with various methodologies such as random forest along with GA, logistic regression along with GA, decision tree along with GA, and XGBoost along with GA.
5. Generated a novel thought for recommendations that can act as a basis for the development of intrusion detection systems (IDS).
6. Explainable artificial intelligence(Explainable AI) concept introduced with the aid of shapley additive explanations(SHAP) values which will enhance the explainability of the interpretation of the obtained results. These explanations

are useful for the generation of recommendations for the development of IDS to detect various DDOS threats.

1.5 Motivation

DDoS threats also called distributed (DoS)denial of service assaults, are one of the leading network cyberweapons utilized solely for personal pleasure by anyone nowadays, from hackers and various organizations to frustrated gamers and the people who seek thrills. By utilizing assimilated online tools such as zombies and servers or IoT bot network systems that flood and overload users with web traffic, the threats block entry to sites and servers or drive them offline entirely. Although personal and social motives can vary, DDOS threats have alike goals: to take network traffic offline on a selected server or servers till the network services are no longer in effect. The aims of DDoS vary from people to organizational agencies and corporations, such as e-commerce websites, bank organizations, stock markets, credit agencies, gaming websites, or providers of network services.

On the contrary, if one able to look at this aspect of the DDOS threat from an organizational or an individual perspective, then one can understand the financial or privacy, or policymaking losses. It might look easier but these attacks might disturb the organizations or individuals both structurally as well as intellectually. To confront such attacks, identifying the solutions are more crucial. In the same instance, achieving the solution for this issue is not that easy without analyzing the various reasons or aspects related to DDOS threats. This situation motivated and lead to identifying the essentiality of analyzing the various categories of threats related to DDOS threats. Identifying the DDOS threat might be the essential objective of the research, at the same instance, categorization and analyzing various categories of the DDOS threats are also essential aspects of the present research. This analysis generates the recommendations related to various attributes of the DDOS threat and these recommendations would help in confronting such threats and thereby generating the possible optimal solution for these threats.

1.6 Problem Statement

DDOS threats are very common and high impactful threats that can crash a network by interfering with the natural flow of the network which is a major concern for the IT industry as well as the individual users. In that aspect, there is a necessity to get into an optimal solution to confront the DDOS threats on the network. For generating such a solution, the previous data on DDOS threats need to be analyzed. Based on that analysis, recommendations can be generated which will be helpful for future research to get into an optimal solution for DDOS threats. In this aspect, the research is designed in various steps such as analysis of the evolution of network features, classification of various kinds of DDOS threats, novel approach for identification of DDOS threats through a framework, measuring the classification through performance metrics, and finally, a recommendation system for the mitigation of these threats.

1.7 Research Objectives

The major research objectives of the proposed framework can be mentioned as follows:

1. **To analyze the evolution of network features in the social network.**
2. **To design and implement the mechanism for the classification of different types of a DDOS attacks.**
3. **To propose a novel approach framework for the detection of a DDOS attack.**
4. **To implement and evaluate the proposed approach using performance metrics.**
5. **To propose a recommendation system for the mitigation of DDOS attacks.**

1.8 Research Questions

Online social network growth and development is rapid concerning time and it is very difficult to comprehend evolution within. Due to that growth, the threats on the networks also increasing day by day. Particularly, the DDOS threats are very impactful threats on the network. It brings to a situation to analyze the various aspects related to DDOS threats for providing the optimal solution for confronting the DDOS threats. The general questions arise when going through this kind of research as follows:

1. How an online social network advanced?
2. Which kind of design mechanisms need to be considered for the classification of various kinds of DDOS threats?
3. How to translate the designs into implementation mechanisms for the classification of various kinds of DDOS threats?
4. How to frame a novel mechanism for the identification of DDOS threats?
5. Which are the various performance metrics that need to be considered for the identified framework?
6. What are the various recommendations that need to be identified for the mitigation of threats?

This research mainly focused not only on proposing the novel approach for the identification of various DDOS threats and their classification but also on identifying the various recommendations based on various threats. These recommendations caper a key part in the creation of ideal results for confronting DDOS threats.

1.9 Research Gap

- Low accuracy of detecting anomalous behaviors in social networks and absence of customized methods for social networks due to diverse behaviors of the users.
- Increased in volume of network data causes slow detection rate.
- Limited parameters were explored for DDOS detection and hence less accuracy occurs in the detection results if the same method is used with more parameters.
- Lack of Freely available dataset and also the available dataset needs to be preprocessed.

1.10 Thesis Organization

The present research is well-organized and structured as mentioned below:

Chapter 2 DDOS attacks research backdrop

This chapter demonstrates the previous research aspects in terms of their literature accessible in DDOS threats. This regressive study enabled me to understand the various research scenarios that have been using for the detection of DDOS threats.

Chapter 3 Methods and Materials

This chapter demonstrates the various methods and related materials that are utilized for introducing novel frameworks, and their comparative studies. This chapter mainly includes the various techniques for machine learning that were used in the suggested research. Besides these, the concept of explainable AI is also mentioned.

Chapter 4 Results and Discussion

This chapter demonstrates the findings acquired by the various comparative studies, novel framework-based results, recommendations along with the results obtained through Explainable AI. Depending on the obtained results, the corresponding discussion is also provided in this chapter.

Chapter 5 Conclusion and Future Work

This chapter serves as a wrap-up that was obtained from the conducted research on the detection and classification of various DDOS threats. This chapter also illustrates the conclusions obtained for the proposed research objectives. Besides these aspects, future work has also been provided in terms of possible challenges and research opportunities in the detection of DDOS threats,

1.12 Summary

This chapter mainly focused on introducing the subject on which the research was conducted. This chapter also includes an overview of the topic, scope of the research, main goals of the research, research contribution, motivation for the research, The research issue, the research goals, and the concerns that developed during the research procedure, and the structure and organization of the thesis. This chapter provides information on the impact of DDOS threats on the network and the necessity of identification and classification of various kinds of DDOS threats.

Chapter 2
Basic Concepts and DDOS Attack Background

2.1 Overview

This chapter mainly deals with the discussion of basic concepts that are related to security and the DDOS threats and their background. The major past research considered in this particular aspect to understand the various problems related to DDOS threats. The popular datasets considered for this past research are also discussed. The solutions for the identified problems are also discussed from the various conducted research works. Also, identified the research gaps from the past research work and the various frameworks are proposed for the analysis of datasets, based on machine learning methodologies, based on ANN, and based on Explainable AI.

2.2 Discussion on Security Aspects

Confidentiality, confidentiality, and availability, or the CIA triad, are essential security foundations as depicted in fig 2.1. It may also be referred to as the AIC triad by re-ordering the essential security aspects. Each of the pillars is defined in detail below:

Confidentiality: Secrecy is another term for confidentiality. Confidentiality aims to keep classified information out of the hands of unauthorized users while allowing legal users access. This promise must be accompanied by a guarantee of information access restrictions.

Integrity: Integrity refers to maintaining the information in its original state, without any changes. At the receiver end, information must be obtained in its original format. File permissions and user access restrictions should be used to ensure data privacy. Integrity can be achieved through several methods, including checksums, cryptography, and so on.

Availability: Accessibility is another term for availability. Availability refers to the ability to provide information whenever and wherever it is required, as well as the ability to resolve any problems as quickly as possible. It can be challenging to get out of a situation created by way of blockade constraints. Amongst the most popular ways for assuring reliability is RAID. Precaution is often needed in the hardware background. The hardware must be stored in a safe location. Firewalls can also be utilized to protect against malicious activities.

Confidentiality
Integrity
Availability

Figure 2.1: CIA Triad [1]

2.3 DDOS Attack

A DDOS assault is described as gaining access to multiple networks utilizing command and control (C & C) and attempting to refuse service from the side of the user. The assailant's main goal is to pull the server down by submitting frequent queries for pages. DDOS uses the Command and Control (C & C) approach to achieve access to applications and servers. Botnet or Zombie networks are those that are influenced by C & C. Botmaster is the individual in charge of the botnet programs. If the number of botnets increases, so does the attack's impact. The HTTP flood attack had a worldwide impact. There are four types of HTTP-based flooding attacks: The demand rates for Session associations started by the aggressors are higher than the solicitations generated by actual clients in a session flooding assault. As a result, the server's supplies are drained, resulting in floods. In a request buffer overflow, the attacker sends sessions with more solicitations than normal clients, resulting in flooding. In an inherently unstable invasion, the attacker sends sessions that include a greater number of high-value task-related requests. The intruder's primary objective is to consume properties such as the CPU and the server's memory, degrading everything. In a slower trespassing assault, the assailant brings HTTP requests in bits (one at a time), and the request isn't completed right away. As a result, the server holds the exposed properties in a holding pattern until it receives all of the details. The main goal of an HTTP overflow DDOS assault is to generate assault movement that faithfully imitates the identity of a human user. As a result, it becomes more difficult for an accidental casualty to distinguish between legitimate and attack traffic. The server becomes unavailable to legitimate clients as a consequence of such an attack. Application layer DDOS attacks have the following main effects: unusually slow organize implementation (visiting webpages or retrieving documents), difficulty to get any site, inaccessibility of a particular site, and a significant enhancement in the number of spamming emails sent.

2.4 History of DDOS

In 1974, David Dennis was the very first individual to launch a DDOS attack[15]. On February 28, when a massive DDOS attack attacked the popular coding site Github, volume peaked at 1.3 TB per second, introducing a high intrusion record[16]. Five days later, the milestone was shattered. On March 5, a US-based "wired media transmission transporter" received a massive flood of traffic, surpassing the previous mark by around a third[17]. The previous record, set two years ago by an attack that thrashed out the BBC's website and reached 602 GB per second [18], was not necessarily a substantial chunk of this website. The following table-2.1 summarises various past DDOS attacks.

Table 2.1: History of DDOS Threats

8Sr. No	Year of Attack	Targets for the Attacks
1	1998	Morris worm
2	1990, 1990	IRC chat floods
3	2000	Yahoo

4	2001	Code red worm attacks
5	2002	DNS root servers
6	2003	Al-Jazeera
7	2004	SCO
8	2005	E-bay
9	2006	Storm pay battling
10	2007	Estonian
11	2008	Georgia president Web site
12	2009	Iranian Government Web sites, Facebook, Twitter, and Google, Russian blog
13	2010	Wordpress.com
14	2011, 2012	Sony
15	2013	South Korean Web sites, Spamhaus
16	2014	JP Morgan
17	2015	Github
18	2016	Russian banks and RIO Olympics
19	2017	Melbourne IT, DreamHost, UK National Lottery, Electroneum, Boston Globe
20	2018	Github
21	2020	Amazon Web Services, New Zealand Stock Exchange, HSBC, Bitcoin, Sony, and PayPal

2.5 Various DDOS Tools

DDOS attacks are dealt with using a variety of tools: [11]

- *PyLoris:* It's a sluggish HTTP DoS tool that allows the intruder to customize HTTP request headers. The packet header, cookies, packet size, timeout, and CRLF choice are among them. The tool's key goal is to maintain TCP links open between the intruder and the defendant's servers for as long as possible.
- *HULK:* Barry Shteiman came up with the idea of this tool. It's a DDoS tool for website servers that were created for testing purposes. It's made to send large amounts of special, confusing traffic to a webserver, bypassing caching mechanisms and entering the server's immediate pool of resources.
- *LOIC:* The abbreviation of LOIC is Low Orbit Ion Cannon. Praetox Technologies initially created it as an open-source network stress testing platform. It permitted programmers to subject their servers to high network traffic spikes for diagnostic intentions, but it has since been updated and extensively utilized by anonymous as a DDoS application in the public interest. It's a platform that generates a lot of network traffic to make utilization of network or device services.
- *DAVOSET:* It is a piece of software that allows you to launch DDOS assaults by abusing any website's features. This command-line interface makes it simple to carry out distributed denial-of-service assaults.
- *Tor's Hammer:* phiral.net developed a sluggish-rate HTTP POST (Layer 7) DoS application. This method made its first appearance in the public eye in early 2011. It launches a DoS assault by utilizing a traditional sluggish POST assault, in which HTML POST domains are sent at a sluggish pace within the same session.
- *XOIC:* Another good DoS-attacking application is XOIC. If the attacker has an IP address, a target port, and a protocol to utilize in the assault, it will launch a DoS assault against any server. In several aspects, XOIC's designers say, XOIC is more efficient than LOIC. It, like LOIC, has a user-friendly interface, so even a novice will utilize it to launch assaults.

- *Trinoo:* This is a form of DDoS assault. The assault server is Trinoo. Trinoo checks for a response from a remote device and then executes a DDoS assault against a third party until it receives it.
- *R-U-Dead-Yet:* This type of assault is popularly known as R.U.D.Y which means R-U-Dead-Yet. This is a denial-of-service application that uses long-form domain submissions to carry out sluggish-rate assaults. R.U.D.Y. is a popular low-level assault application for crashing a website server by having to submit long-form domains. The assaults are carried out using a DoS application that scans the targeted server for integrated website forms.
- *TFN:* When a Tribal Flood Network (TFN) master station has sent a command to a list of TFN servers to start a DoS assault, this is known as a TFN assault. The TFN system is capable of generating packets with spoofed source IP addresses. The werewolves produce the stipulated DoS assault against the listed targets when they receive a command. The spoofed source IP address and source ports can be randomized to make the assault more prevalent. The packet sizes can be changed to avoid identification.

2.6 Related Work

Deep learning is gaining popularity these days as a consequence of its higher accuracy and performance. Its implementations in this field are being researched by a community of scholars. Automotive architecture, healthcare, manufacturing, and law enforcement are some of the well-known realms. The study that has already been completed by various scholars is mentioned below. Hariharan et al. [19] equate the effects of the C5.0 machine learning methodology to those of other The Naive Bayes categorization methodology and the C4.5 decision-tree categorization methodology are two examples of machine learning techniques. The researcher mostly attempted to act in a version that isn't online. If the scale of the dataset grows larger, the output disparity has become more pronounced.

Deep Intelligence was introduced by Bhuvaneswari Amma N.G et al [20]. The knowledge was derived using a radial base function with a variety of abstraction levels. The research was conducted on well-known datasets such as NSL-KDD and UNSW NB15 that included 27 functions. In comparison to other existing methods, the researcher believed that his method was more accurate. Muhammad Aamir et al. [21] used a clustering technique to apply a feature selection process. Five related machine learning methodologies were used to compare the algorithm. For preparation, SVM and RF methodologies were utilized. The best accuracy was attained by RF, which was about 96 percent.

Dayanandam et al. [22] classified packages depending on their characteristics. Through inspecting the IP header, the protection strategy attempts to identify IP addresses. These IP addresses are utilized to distinguish between spoofed and legitimate addresses. As the scale of the assault becomes larger, firewalls are ineffective. For separating the regular and assaulted traffics, Narasimha et al [23] utilize anomaly identification and machine learning methodologies. Real-time datasets were utilized in the research. For classification, the well-known naive Bayes ML methodology was utilized. The outcomes were compared to those of other methodologies such as J48 and RF.

J. Cui et al. [24] combined intellectual-stimulated computation and the entropy methodology in their research. For the classification, Support Vector Machine Learning was utilized. The platform's flow chart was being mined for information. In terms of identification precision, the outcomes were satisfactory. Omar E. Elejla et al. [25] used an IPv6 classification strategy to incorporate a methodology for the identification of DDOS assaults. The findings were compared to 5 different well-known machine learning methodologies by the scientist. DT, SVM, NB, KNN, and NN were the methodologies utilized. The research was conducted on a well-known dataset. According to the source, the KNN methodology achieved a precision of about 85 percent.

Mohamed Idhammad et al. [26] used the ML methodology to create an entropy-depend semi-supervised methodology. Unsupervised and supervised designs are used in this development. Unsupervised techniques, in particular, have high precision and low false-positive rates. Supervised methods, contrarily, minimize the number of false positives. The datasets utilized in the research were NSL-KDD, UNB ISCX 12, and UNSW-NB15. For the recognition of the assault, Nathan Shone et al. [27] use a DL methodology. It also utilized the NDAE function for unsupervised instruction. On the well-known KDD Cup 99 and NSL-KDD datasets, the proposed methodology was executed on a GPU utilizing TensorFlow. The researcher believed that he was able to get more precise identification outcomes.

To identify the DDOS threat, Olivier Brun et al. [28] researched the domain of IoT. For network identification, the researcher used the random neural networks (RNN) method. In comparison to conventional approaches, this deep learning-depend approach effectively produces more promising outcomes. For network behavior-specific IoT-based DDOS identification, Rohan Doshi et al [29] proposed utilizing function selection methodologies. High identification sensitivity can be achieved with careful function selection. The distinction is made using a variety of machine learning methodologies as well as neural networks. A flow-based methodology was used to achieve the data.

Alekhya Kaliki et al. [30] developed a novel machine learning-dependent bio-inspired process. For identifying App-based DDOS threats, this system employs an anomaly identification technique. The http flooding assault was assigned the most weight. The study was conducted utilizing the acclaimed CAIDA information collection. Chuanhuang Li et al. [31] utilized a DL-based approach to identify and avoid DDOS attacks in a SoftwareDefined Network (SDN). The traffic series was produced and evaluated. The approach removes the reliance on the datasets' accessibility.

Muna AL-Hawawreh et al. [32] used an anomaly identification approach to design a deep learning methodology for IICSs. Utilizing data from TCP/IP packets, the modeled framework was learned and tested. Preprocessing was carried out on well-known datasets such as NSL-KDD and UNSW-NB15. The feedforward neural network was created to obtain improvised effects. Xiaoyong Yuan et al. [33] developed a system for processing network traffic and tracking network operations using recurrent neural networks (RNN). The research was conducted on well-known datasets, and the researcher believed that novel findings were acquired.

For network traffic analysis, Kim et al. [34] utilized a DNN. The Rectified Linear Unit method is used in conjunction with 100 hidden neurons in this methodology. On well-known KDD datasets, the approach was executed on GPU utilizing TensorFlow. The researcher believed that the precision he received was 99 percent. Indraneel Sreeram et al. [35] developed a DDOS identification methodology that attempts to incorporate the bat methodology's functionality. This strategy was designed to counteract app-based

DDOS assault. To test the hypothesis, the popular CAIDA dataset was utilized. In comparison to previous research, the accuracy findings obtained were excellent.

To avoid DDOS attacks, Marwane Zekri et al. [36] used a machine learning methodology dependent on C 4.5. Along with the decision tree, this methodology was paired with signature recognition techniques. Other machine learning methodologies were compared, and the findings were believed to be more reliable. The research of Hossein Hadian Jazi et al. [37] emphasizes application layer-based DDOS assault. The nonparametric CUSUM methodology was utilized to apply the framework. The identification research was carried out on a subset of device flow. The research produced findings that reduced failure rates and increased identification accuracy.

Abebe Abeshu Diro et al. [38] developed a method for identifying IoT attacks deep learning-based methodology. This model was compared to conventional machine learning methodologies. Based on the findings, distributed assault identification was compared to a centralized method, which was found to be an effective solution. For modeling IDS, Syed Ali Raza Shah [39] used a hybrid method. SVM and fuzzy logic approaches are combined with this hybrid method. Snort and Suricata were the names assigned to the two IDS. The SVM, when used in conjunction with the firefly methodology, produces stronger performance.

Hodo et al. [40] conducted a thorough investigation into network intrusion detection technologies (NIDS). Different deep learning approaches have been extensively explored by the scholar. There was also a discussion of the most positive findings. In a software-defined network (SDN) system, Niyaz et al. [41] suggested a DL approach for detecting DDOS assaults. The outcomes were produced using customized traffic. The research revealed that binary classification (99.82 percent) and 8-class classification each had a high accuracy rate (95.65 percent).

For semiconductor processing, Lee et al. [42] developed an autoencoder based on an unsupervised process. Stacked de-noising Auto-encoder was the name given to the created system (SDA). When compared to the previous method, the precision achieved was improved by 14%. Besides, the findings of layer 4 were said to be more positive. Tang et al. [43] created a network flow-based approach for identifying DDOS assault's inaccessible internet traffic. The NSL-KDD dataset was utilized for the research, and the accuracy estimated was greater than 75% when 6 specific features were taken into account.

M.S. Hoyos et al. [44] developed a supervised classification system based on machine learning SVM. After the network movement was registered, the HTTP header was examined. In comparison to standard methods, the researcher claimed to attain a 99 percent accuracy score. Elike Hodo et al. [45] developed a hazard detection approach focused on deep learning techniques. For research, an Artificial Neural Network (ANN)-based deep learning methodology was utilized with a supervised implementation. They attempted to identify and classify usual and assault packets in an IoT-based network by analyzing IP fragments. The findings were encouraging, with a 99.4 percent accuracy score.

Deep learning-based DDN approach developed by Kang and Kang [46]. The classification was done using an unsupervised DBN technique, and the outcomes were achieved using well-known datasets. The acquired outcomes reveal that the identification rate has improved accuracy. Wang et al. [47] developed a JavaScript-based approach that combines three layers of SDA with linear regression. The research was conducted on a well-known dataset. They appeared to have the largest frequency of true positives and the following largest frequency of false positives of the current approaches were compared.

A DL-based framework for a regular system was developed by Cordero et al. [48]. This method employs an unsupervised technique, as well as an RNN and an autoencoder. Even though the research was carried out, the researcher did not reveal the findings. Utilizing the deep learning approach, Javaid et al. [49] applied STL methodology. The technique employs a meager auto-encoder in conjunction with the regression type of softmax. For the observational investigation, the NSL-KDD dataset was utilized, and the binary and 5-class classification outcomes achieved were about 75% for the f-score. Potluri and Diedrich [50] used the DNN framework to apply a deep learning technique. The researcher culled a total of 41 features, of which 27 were included in the research. The research yielded a mixed bag of findings.

You et al. [51] utilized the RNN methodology to implement a deep learning methodology. The research was carried out on a well-known dataset. A comparison of SVM and Naive-related approaches was performed. The researcher appeared to have gotten impressive outcomes with a 92.7 percent accuracy rate. Alrawashdeh et al. [52] used unsupervised feature reductions to apply deep learning. The research was carried out using a logistic regression classifier. The popular KDD Cup '99 dataset was utilized, with a 97.90 percent identification score.

Dong et al. [53] utilize a consolidation of DL and traffic anomaly techniques. The researcher attempted to fix the dataset's issues. According to the scientist, encouraging outcomes were achieved. For the health tracking unit, Zhao et al. [54] used deep learning techniques. The researchers compared and studied four deep learning processes. The research was carried out on a well-known dataset and yielded positive outcomes.

Mohammed et. al [55] created a structure focused on the Multilayer Perceptron(MLP) methodology. DDoS and HTTP flood-based assaults were the subjects of the experiments. The researcher believed that the achieved findings were about 98 percent accurate. Tesfahun et al. [56] developed a methodology based on Oversampling. The study was conducted on the acclaimed NSL-KDD information collection, with feature selection dependent on information gain. During the classification research, the machine learning Random Forest methodology was utilized.

2.7 Comparative Study on Existing Methodologies

Table 2.2: Comparative study over the collected research articles

Authors	Year	Methods used	Methodology	Conclusions
Hariharan. M et. al [19] [Modern Education press]	2019	Machine Learning (ML) Classification	C5.0 ML algorithm	Once the harm has been done, the identification process begins.
Bhuvaneswari Amma N.G et. al [20] [Neurocomputing]	2019	Deep Intelligence	Radial Basis Function	The rate of false alarms is lower as there are more identification thresholds.

Muhammad Aamir et. al [21] [ELSEVIER]	2019	Clustering-based semi-supervised machine learning	The clustering methods	On labeled results, supervised learning methodologies are used.
G. Dayanandam et. al [22] [Springer Nature]	2019	Access Control List (ACL)	Firewall	If traffic grows too quickly, the firewall can shut down.
K.Narasimha Mallikarjunan et. al [23] [Springer Nature]	2019	10-fold cross-validation	ML Algorithms	The Naive Bayes method produces better outcomes.
Jie Cui et. al [24] [Future Generation Computer Systems]	2019	Cognitive inspired computing	cognitive-stimulated computing with dual address entropy	After an assault has been identified, fast identification and protection are important.
Omar E. Elejla et. al [25] [Springer Nature]	2019	ICMPv6-based	Five related methodologies are compared.	KNN methodology yields more encouraging outcomes.
Mohamed Idhammad et. al [26] [Springer Nature]	2018	the online sequential semi-supervised ML approach	a semi-supervised approach	On a sliding window, predictions and evaluations are made.
Nathan Shone et. al[277] [IEEE TRANSACTIONS]	2018	Nonsymmetric deep autoencoder (NDAE)	RF	Obtains greater precision
Olivier Brun et. al [28] [ELSEVIER]	2018	Random neural networks (RNN)	RNN	The packet will be used to make predictions.

Rohan Doshi et. al [29] [IEEE]	2018	packet-level machine learning DoS detection	Modeled machine learning	KN, LSVM, DT, RF, NN are evaluated and compared.
Alekhya kaliki et. al [30] [IJPAM]	2018	Bio-inspired approach	Bio-inspired based	Minimal process Complexity
Chuanhuang Li et. al [31] [Wiley]	2018	Deep learning	OpenFlow flow entries based Software-Defined Network (SDN)	LSTM,CNN/LSTM ,GRU, 3LSTM utilized for outcome comparison
Muna AL-Hawawreh et. al [32] [ELSEVIER]	2018	Anomaly Detection System (ADS)	Unsupervised learning	The capacity of automated dimensionality is being reduced.
Xiaoyong Yuan et. al [33] [IEEE]	2017	Recurrent Neural Network (RNN)	CNN, RNN	The transition is carried out based on a window.
Jin Kim et. al [34] [IEEE]	2017	Deep neural network (DNN) model,	DNN model	RNN needs time-series data processing.
Indraneel Sreeram et. al[35] [Elsevier]	2017	Bio-Inspired Anomaly-based	ML-based on bat	minimal process complexity
Marwane Zekri et.al [36] [IEEE]	2017	C.4.5 algorithm	signature detection techniques	Targets were focused on layer 3 and layer 4

Hossein Hadian Jazi et. al [37] [Computer Networks]	2017	Nonparametric CUSUM algorithm	CUSUM algorithm	thirteen separate sampling techniques were investigated.
Abebe Abeshu Diro et. al [38] [Future Generation Computer Systems]	2017	IoT/Fog Computing	distributed deep learning-based IoT/Fog	The outcomes of centralized methodologies are less reliable.
Syed Ali Raza Shah [39] [Future Generation Computer Systems]	2017	Snort adaptive plug-in	Support Vector Machine (SVM)	To achieve optimal efficiency, the hybrid approach is utilized.
Q. Niyaz et.al [41] [EIA]	2017	Deep learning-based	ML and DL	The assessment is based on custom-created traffic traces.
Lee et al. [42] [IJFIS]	2017	Stacked de-noising Autoencoder (SDA)	unsupervised learning solution	Through more usage cases, more precision is gained.
Tuan A Tang et. al[43] [IEEE]	2016	Deep Neural Network (DNN) model	deep neural network	The derived outcomes cannot be utilized for commercial purposes.
M.S. Hoyos et. al [44] [Springer]	2016	Support Vector Machines (SVM)	SVM	The SVM framework has a lot of benefits.
Elike Hodo et. al [45] [IEEE]	2016	Artificial Neural Network (ANN)	A multi-level perceptron,	Classifying normal and threat patterns.

			supervised ANN	
Kang et. al [46] [PLoS One]	2016	unsupervised DBN	DNN	Errors in classification are minimized.
Wang *et al.* [47] [Security Communication Network]	2016	JavaScript	linear regression	Comparatively good positive rates are obtained
Cordero *et al.* [48] [IEEE]	2016	unsupervised method	RNN, auto-encoder	The degree of precision achieved isn't fully disclosed.
Javaid *et al.* [49] [IEEE]	2016	Self-taught learning (STL),	Softmax regression	The binary and grouping precision achieved is satisfactory.
Potluri et.al [50] [IEEE]	2016	DNN	DNN	Fewer class results obtained are comparatively good
You et al. [51] [IEEE]	2016	RNN	the RNN model	Improved classification accuracy results
Alrawashdeh et. al [52] [IEEE]	2016	RBM	Unsupervised feature reduction.	Better identification accuracy with minimal error.
B. Dong et.al [53] [IEEE]	2016	Deep learning	Deep learning-based methods.	Oversampling is a technique for resolving database issues.

Sang Min Lee et. al [57] [ELSEVIER]	2012	traffic matrix	Genetic Algorithm (GA).	Identification rate improvised with fewer parameters.

2.8 Summary

Even though a lot of methodologies has been designed to detect DDOS attack with high accuracy still there are a lot of areas where improvements can be done. These areas may include dependency on a human operator, lack of freely available datasets, long training and computing time, a lot of preprocessing of datasets, etc. Still, there are lots of areas that need to be focused upon. Depending on those research rs the present research work is framed.

Chapter 3
Methods and Material

3.1 Overview

This chapter mainly focused on the discussion of various methods and materials considered for the present study of research. In a major aspect of view, this chapter can be fragmented into various sections. First section deals with the discussion of the data that is utilized for the present study of research and a brief elaboration of the data mentioned. Second section deals with the discussion of the various machine learning methodologies. This section includes feature selection methodologies as well as classification methodologies along with proposed methodologies. Final section deals with the discussion of the measures for evaluating conduct for the evolution of the implemented methodologies with the aid of various machine learning methodologies as well as the ANN.

3.2 Data Information

Although different devices are situated in the computers/networks for the safety of an individual's important data, the computers/networks can avoid bugs and worse, network traffic from being breached. IDSs are technologies that can identify flaws in traffic that are used to practice network-related traffic details. The NSL-KDD is the most widely used data set and the standard for modern enlightened internet traffic. The NSL-KDD Dataset isn't the only one with this function. International Information Exploration and Data Mining Resources faced an obstacle in the KDD Cup. This competition aimed to collect network traffic records in 1999. The difficult challenge was to create an IDS, a statistical model capable of distinguishing between "unpleasant" links, also known as intrusions or attacks, and "reasonable" regular links. As a consequence of this challenge, a huge portion of the network movement reports was gathered and assembled into a data set known as the KDD'99, and with this information, another dataset known as the NSL-KDD information collection was created as an amended and improved adaptation of KDD'99 with information cleaning by the University of New Brunswick.

The NSL-KDD information collection is divided into four sub-information collections: KDDTest+, KDDTest-21, KDDTrain+, and KDDTrain+ 20Percent. The sub-datasets KDDTest-21 and KDDTrain+ 20Percent, in particular, are sub-datasets of KDDTest+ and KDDTrain+, respectively. The above datasets are network traffic records utilized by a typical intrusion prevention network, which are ghosts of the traffic that an individual IDS experiences and only serve as indicators of its existence. The data set contains 43 characteristics per document, with 41 of them referring to the traffic network, the last two being labels and a final characteristic named score that reflects the magnitude of network traffic. DoS danger, Inquest threat, U2R threat, and R2L hazards are the 4 hazard classes included in the dataset.

Table 3.1: Breakdown of Different Sub-classes of Each Attack in NSL-KDD

Classes	**DOS**	**Probe**	**U2R**	**R2L**

Sub-Classes	➢ apache2 ➢ back ➢ land ➢ neptune ➢ mailbomb ➢ pod ➢ processtable ➢ smurf ➢ teardrop ➢ udpstrom ➢ worm	➢ ipsweep ➢ mscan ➢ nmap ➢ portseep ➢ saint ➢ satan	➢ buffer_overflow ➢ loadmodule ➢ perl ➢ ps ➢ rootkit ➢ sqlattack ➢ xterm	➢ ftp_write ➢ guess_passwd ➢ httptunnel ➢ imap ➢ multihop ➢ named ➢ phf ➢ sendmail ➢ Snmpgetattack ➢ Spy ➢ snmpguess ➢ warezclient ➢ warezmaster ➢ xclock ➢ xsnoop
Aggregated Sum	11	6	7	15

The following is a brief description of each hazard.

- DoS is an attack that tries to stop network traffic from and to the intended user/device for a short period. The IDS is overwhelmed with an extraordinary amount of traffic that it can't accommodate, so it shuts down temporarily to secure itself. This prevents routine traffic from accessing a network. An example would be if, on the day of a big sale, an internet retailer is inundated with online orders, and when the network can't handle them all, it slows down to prevent potential customers from buying anything. This is the most common assault in information collection.

- Probe attacks are intrusions that attempt to access information from a network. The aim is to act like a thief and steal sensitive information, whether it's customer behavioral information or financial information.
- U2R is an attack that starts as a default user profile and then tries to access the computer or network as a super-user (root). To gain root authorization/admittance, the attacker attempts to exploit a device's limitations.
- R2L is an assault that tries to gain local entry to a remote computer. An attacker is someone who does not have local entry to a network and is attempting to hack into it.

Table 3.1 shows the breakdown of the NSL-KDD information collection sub-classes according to each of the assaults. Since these assaults arise in the information collection, the representation is heavily skewed. A record dispersion breakdown is seen in the table below, as described in Table 3.2. Each dataset containing more than half of the documents is regular traffic, and the distribution of challenges such as the U2R and R2L is exceptionally bad. Despite the lower frequency, this is an accurate indicator of the spread of current network traffic risks, with DoS being the most common assault to network traffic and U2R and R2L being seldom used as assaults to network traffic.

Table 3.2: The Distribution Breakdown of various threats on NSL-KDD Dataset

Dataset	Total number of records as per the sections					
	Total	**Normal**	**DOS Threats**	**Probe Threats**	**U2R Threats**	**R2L Threats**
KDDTrain+20%	25192	13449 (53%)	9234 (37%)	2289 (9.16%)	11 (0.04%)	209 (0.8%)
KDDTrain+	125973	67343 (53%)	45927 (37%)	11656 (9.11%)	52 (0.04%)	995 (0.85%)
KDDTest+	22544	9711 (43%)	7458 (33%)	2421 (11%)	200 (0.9%)	2654 (12.1%)

3.3 Methodologies

This part is mostly concerned with the discussion of distinct methodologies utilized for the development of the framework of research conducted. The list of methodologies considered for the present research can be categorized into two. The primary category includes feature selection methodologies and the secondary category includes classification methodologies. Feature selection methodologies include PCA, LDA, RFE, univariate feature selection, and genetic algorithm(GA). The detection and classification methodologies are SVM, KNN, logistic regression, decision trees, RF, SGD, XGBoost, artificial neural networks(ANN), and explainable AI.

3.3.1 Preprocessing

The collection of significant attributes from the dataset is critical to the performance of every ML model. The most common KDDCup99 dataset [64] is utilized for assault classification in the execution. There are forty-one distinct attributes in this dataset, including content sort, simple type, and traffic type. It was created using the DARPA'98 IDS assessment software as a foundation.

All the records in KDD dataset are categorized into two ways:

1. Normal intrusion
2. Kind of intrusion (There are four categories of intrusion in the dataset [64][65])
   - *A denial-of-service attack (DOS):* The oldest type of cyber extortion assault is a DoS assault. Essentially, in this assault, the intruder leaves the system very busy, and as a result of that computer or program, the genuine client is denied access.
   - *Remote to Local attack (R2L):* A flaw in a particular version of ncftp, a popular FTP client, is exploited. The repository contains a directory with a very long name; the FTP client then executes (unintentionally) one or more instructions found in the name with the customer's approval. Password guessing, for instance.
   - *Unauthorized entry to local superuser (root) allowances (U2R):* It gains access to the network's core and launches a series of threats and

unauthorized strives. For instance, different buffer overflow assaults, Perl, rootkits, and so on.

- *Probe:* This assault is trying to collect data about a system of PCs with the probable goal of deceiving the surveillance authority. Sending an empty message, for instance, to see if there is a target. To submit such a sample, ping is a simple method. Port scanning, SATAN, SAINT, portweep, and other similar tools are examples.

A cleaning process transforms the raw dataset into a template that can be utilized to train a machine learning algorithm. When creating an effective framework, the information must be in the correct format. Some attributes in the KDD dataset are of the object category. The dataset for training a model should not have an object data type function. Since some of the KDD dataset's attributes are of the object kind, they must be transformed to floats. This is accomplished using the Label Encoder class. From object data type to float type, it encodes the functions.

3.3.2 Feature Selection Methodologies

This part focuses on the justifications of distinct feature selection methodologies that are utilized for the current research. These methodologies play a vital role in the selection of crucial features that would help in classifying the various classes of threats more efficiently.

In this process, the less redundant or more relevant features from a dataset are selected. It is essential to select the relevant features because They have the potential to influence the efficiency and accuracy of the framework [64].

Decreasing redundancy and selecting more suitable features helps to:

- Decreases size of the dataset
- Decreases the risk of overfitting
- Decreases the misleading of the data
- Decreases the time of training the model
- Improves the accuracy of the model

Redundancy can be decreased by dropping irrelevant or partially relevant features from the dataset [65-67]. It is the most important step for almost every framework which uses a dataset having high redundancy or having a large number of columns. Because training the model on irrelevant features may negatively affect the model’s accuracy [68-71].

3.3.2.1 Principal Component Analysis(PCA)

It is widely used in unsupervised learning. The approach of this technique is very simple, but it can make a fair difference between the accuracy of the model trained when applied to all the features. Initially, it calculates the covariance of data points and arranged them in a matrix form. Further, it calculates the Eigen Vector and Eigen Value of that matrix. Then, it arranges all Eigen Vectors according to their decreasing order, the Eigen Scores. Furthermore, it selects the most promising features for training the model. It converts the original dataset into the selected number of Eigen Vectors. PCA is also used in the field of medical science and chemistry. It is also used to reduce the distortion from a graph. Eq. (2) describes the covariance computation of X & Y. Where X and Y are matrices of m, and P is a linear transformation. X is the original data set point, and Y represents the Re-deployment dataset.

$$PX = Y \tag{1}$$

$$cov(X,Y) = \frac{1}{n-1}\sum_{i=0}^{n}(Xi - \bar{x})(Yi - \bar{y}) \tag{2}$$

3.3.2.2 Linear Discriminant Analysis(LDA)

This methodology is a widely utilized feature selection methodology. This technique mainly removes the redundant and dependent features from the dataset. It mostly comprises three major phases. In the primary phase, it calculates the difference between the averages of different classes. This difference is known as Between Class Variance. In the second stage, it calculates the difference between the average and the sample values of each class. This difference is known as Within Class Variance. In the third stage, it selects the features that have greater Between Class Variance and less Within Class Variance. LDA is also broadly used in the field of bioinformatics and chemistry. Consider that the PDF of x with mean vector μi and variance-covariance matrix (same for all populations) is multivariate normal in population πi. For this scenario, the normal function of probability density is calculated as mentioned in Eq. (3).

$$P(X/\pi i) = \frac{1}{(2\prod)^{p/2}|\Sigma|1/2} exp\ [-\frac{1}{2}(X - \mu i)'(X - \mu i)] \tag{3}$$

3.3.2.3 Recursive Feature Elimination(RFE)

The name of this technique is self-explanatory. It works in a loop and removes a few features in each loop. If a dataset has high co-linearity and dependencies, then RFE can be the best algorithm to eliminate these features. RFE first evaluates the importance of the features and then ranked them accordingly. Elimination of the weakest features is performed in this step. RFE class mainly takes two arguments, the classifier and the number of features to be selected. A Logistic Regression classifier is used in the proposed framework as it takes comparatively less time for training the model. RFE trains the model using the classifier provided and calculates the accuracy by eliminating the unwanted features. RFE takes more time to compare to Univariate Feature Selection because it trains the model until the end of the loop.

3.3.2.4 Univariate Feature Elimination

Feature, which has the most stable relationship between the label features of the dataset, can be called an important feature. These features can have a huge impact on the accuracy of the framework. For the selection of these attributes, statistical approaches can be applied. Different statistical approaches can be implemented with the help of a class named SelectKBest. In this framework, the chi2 (chi-squared) test has been implemented for selecting the best features. Chi2 score technique is implemented on the dataset with f features and c classes by using Eq. (4):

$$\chi 2 = \sum_{i=1}^{f}\sum_{j=1}^{c} \frac{(s_{ij} - \mu_{ij})^2}{\mu_{ij}} \tag{4}$$

Where $\boldsymbol{s_{ij}}$ is the ith value of the feature along with the instances. And

$$\mu_{\boldsymbol{ij}} = \frac{\boldsymbol{s_{*j} s_{i*}}}{\boldsymbol{s}} \qquad (5)$$

Where $\boldsymbol{s_{i*}}$ is the ith value of the specific feature, $\boldsymbol{s_{*j}}$ is the number of instances in class j, and s is the number of instances in Eq. (5). The feature selection-based model training and validation is considered as mentioned in fig. 3.1.

3.3.2.5 Proposed Methodology for Feature Selection as a Stack

Input: Set of 41 features from KDD'99 cup dataset

Output: Best selected features subset

Step-1: Select the features having a value greater than or equal to 0.7 and less than or equal to 0.7 and apply Correlation-based feature selection.

Step-2: Calculate Pearson's correlation coefficient using the following Eq. (6).

$$\boldsymbol{Correlation_{coef}} = \frac{\boldsymbol{covariance(x,y)}}{\boldsymbol{(stdv(x) * stdv(y))}} \qquad \textbf{(6)}$$

Step-3: Select the features subset which satisfies the threshold.

Step-4: Repeatedly apply the feature selection from the stack of univariate, RFE, PCA, LDA on the features obtained from step-3.

The following flowchart represents as mentioned in fig. 3.1, the flow of the proposed attribute selection methodology.

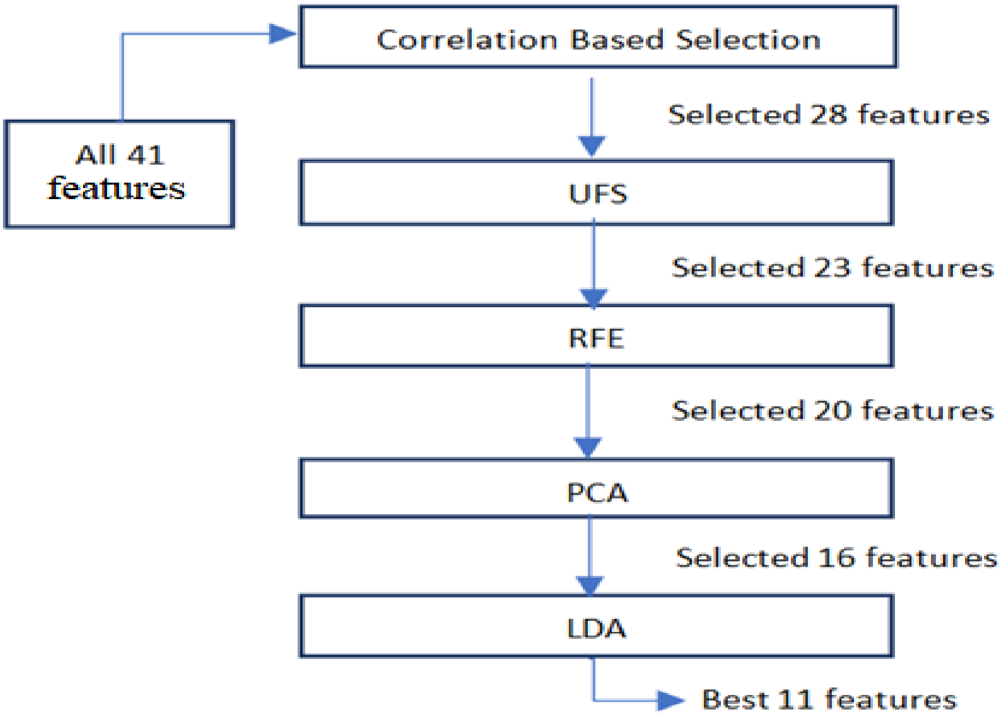


Figure 3.1: Stacked-Based Attribute Selection

For feature selection, five different techniques have been used namely Correlation-based feature selection, Linear Discriminant Analysis (LDA), Univariate Feature Selection, RFE, PCA. All the feature selection methodologies are also used as a stack. Initially, the Correlation-based attribute choosing methodology is applied to the dataset. The availability of a Correlated feature in a dataset can decrease the performance of the model, and also it can affect the accuracy of the model, so these features need to be dropped from the dataset. To make a correlation matrix Pearson’s Correlation Coefficient (PCC) is used.

3.3.2.6 Genetic Algorithm(GA)

The main goal of this methodology is to generate the best outcome in the shortest amount of time. This rationale will have a significant influence on optimization-related issues. The following are the reasons for the requirement of the GA. NP-Hard challenges, or those that need a lot of processing power to solve quickly, can be addressed more effectively with this technique, resulting in near-optimal answers in the least amount of time. Gradient techniques will be utilized in real-world issues to get optimal results, but they will collapse in the final. In such cases, the GA methodology may be used to achieve the best outcomes in a shorter amount of time.

An approach linked to optimization is the GA methodology. It's a method for addressing optimization issues that are both constrained and unconstrained, based on the choosing of an instinctive procedure that promotes root development. It refreshes the collection of various outcomes from the current overall community regularly. At any stage, the organizations are selected as parents arbitrarily from present and global populations and use them to create children for the next iteration. In the next iterations, the species is progressing to an ideal finding. It may be used in evaluating different optimization-related difficulties, which in traditional optimization methods are not a better situation. It may also analyze mixed-integer challenges with integrity-limited elements. It uses 3 main types of rules to create the next generation from the present and general population:

- Selection regulations: Because of these rules, the organizations, known as parents, pick the next generation.
- Crossover regulations: Under these rules, parents create for the succeeding generation, which is a combination of both organizations.
- Mutation regulations: As a result of these rules, certain elements, i.e. parents to generate children, have random modifications applied.

The flow charts of the methodology as described in fig 3.2 show all the above laws and movements.

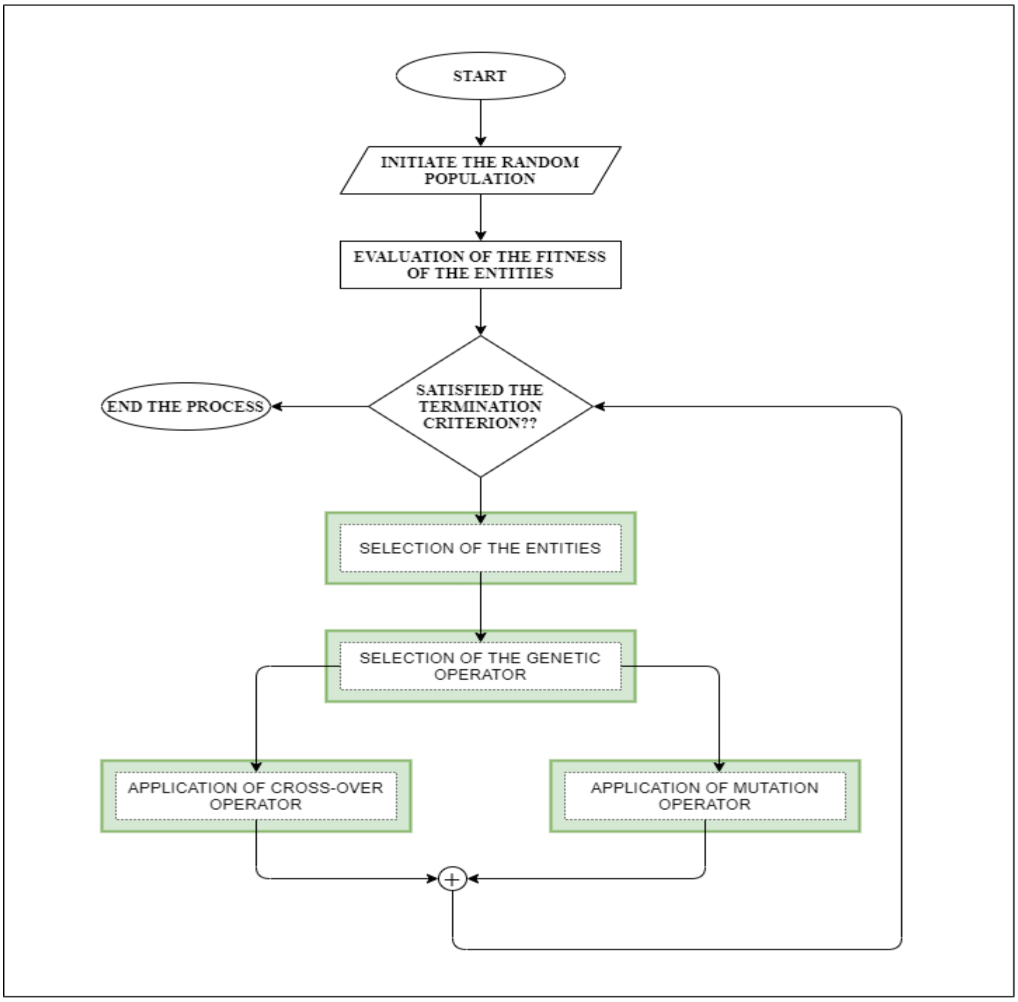

START
INITIATE THE RANDOM POPULATION
EVALUATION OF THE FITNESS OF THE ENTITIES
SATISFIED THE TERMINATION CRITERION??
END THE PROCESS
SELECTION OF THE ENTITIES
SELECTION OF THE GENETIC OPERATOR
APPLICATION OF CROSS-OVER OPERATOR
APPLICATION OF MUTATION OPERATOR

Figure 3.2: The Genetic Algorithm Flow Chart

3.3.3 Classification Methodologies

The research work is majorly based on machine learning for the identification and classification of various threats on the network/device in concern with DDOS. These methodologies are discussed in the following sections.

3.3.3.1 K-nearest Neighbor(KNN)

This algorithm finds the nearest neighbors and differentiates them in a class. It comes under the category of the supervised algorithm. It identifies the closest neighbor by using the Euclidean distance formula. The implementation of this technique is simple to understand. Initially, separate the dataset in the training and testing set, and choose important features that are required to select from the training data. Then, find the distance between all the points by using the Euclidean distance formula and store it in a list. Further, the sort that lists and selects the first n values (number of features needed) from the dataset and then allocates a class to test the points based upon the majority of classes available in the points that have been chosen. Following are the various distance measuring techniques possible in KNN along with their standard formulas as mentioned in Eq. (7 - 9).

Euclidean Distance Function:

$$\sqrt{\sum_{\boldsymbol{i=1}}^{\boldsymbol{f}} (\boldsymbol{X_i} - \boldsymbol{Y_i})^{\boldsymbol{2}}} \tag{7}$$

Manhattan Distance Function:

$$\sum_{\boldsymbol{i=1}}^{\boldsymbol{f}} |\boldsymbol{X_i} - \boldsymbol{Y_i}| \tag{8}$$

Minkowski Distance Function:

$$(\sum_{\boldsymbol{i=1}}^{\boldsymbol{f}} (|\boldsymbol{X_i} - \boldsymbol{Y_i}|)^{\boldsymbol{q}})^{\frac{\boldsymbol{1}}{\boldsymbol{q}}} \tag{9}$$

Where *X* and *Y* are two distinct points and *f* is the number of instance points.

3.3.3.2 Logistic Regression

This is a methodology for categorization that is universal as well as widely utilized. This methodology is very simple to utilize and its output in a linearly separable group is superlative. This is focused on a sample's probability of belonging to a group and these values lie between 0 and 1 and are continuous in nature. The purpose of the logistic regression algorithm is to establish a linear decision boundary that divides two groups from each other. In this methodology, a conditional probability provides this decision boundary that separates the two groups. A threshold function is utilized in this methodology for the decision-making related to identifying a data point belonging to which group, popularly known as a sigmoid function or logistic function and it can be represented as mentioned in Eq. (10).

$$f(x) = \frac{1}{(1+ e^{-z})} \tag{10}$$

3.3.3.3 Support Vector Machine(SVM)

The major aim of this methodology is to determine a hyperplane that separates into various groups in a space that consists of N attributes. Various possible hyperplanes can be chosen to differentiate between the two information points groupings. The major aim of this methodology is to determine a plane that has the highest margin, i.e. the maximum gap among all data points that belong to various classes. Enhancing the width disparity provides some assistance to classify additional trust into possible information items. The data points belong to various classes will be separated with the aid of decision boundaries those are nothing but hyperplanes. The number of features decides the dimension of the hyperplane. The data points nearer to the hyperplane and those points that can impact the position, as well as the hyperplane alignment is termed vectors of support. These support vectors play a vital role in maximizing the gap among the various classifiers. These support vectors helpful in building an SVM-based model.

3.3.3.4 Decision Tree

A general, statistical modeling technique that has implementations covering a variety of distinct fields is Decision Tree Interpretation. Decision trees are usually built by a computational methodology that defines approaches to segment a dataset based on various conditions. It is among the most commonly utilized methodologies for supervised learning and is effective. A non-parametric supervised learning approach is utilized for both classification and regression aspects through Decision Trees. The aim is to construct a framework that forecasts conditional probabilities by studying basic rules of judgment derived from the data characteristics. In general, the principles for choices are in the form of if-then-else sentences. The deeper the tree, the more complicated the laws are and the framework is more suitable. A decision tree is a similar structure that of the tree with various nodes representing the spot where a feature is chosen and a query is asked; edges represent the responses to the queries, and the real outcome or group mark is represented by the leaves. They are utilized for basic linear decision surfaces in non-linear decision-making.

3.3.3.5 Random Forest

As the title of the methodology suggests, this methodology includes a huge volume of Trees of individual choice acting as an ensembling model. Every tree of choice in the random forest churns out a group forecasting and the group with the majority votes turns out the forecasting of the framework. The core idea behind this methodology is collective wisdom, a plain yet strong one. The rationale that this methodology paradigm performs so well in data science as any of the behavior of individual models will be surpassed by a huge volume of relatively uncorrelated frameworks working as a committee. The main aspect is the low association between the frameworks. Much when low-correlation portfolios (such as stocks and bonds) come together to construct unrelated frameworks may provide ensemble forecasts which are more trustworthy than any one of the many projections for a portfolio that is higher than the total of its parts. This magnificent effect has been explained by the trees defend each other (as long as they do not all err in the same direction) from their mistakes. While some trees are wrong, numerous other trees are accurate, since they travel as a cluster in the proper direction. For random forests to function well, the preconditions are:

- In the set of attributes, there is a necessity of a real signal such that models created utilizing such attributes perform better than random speculation.
- The forecasts about the individual trees must have fewer relations within each other.

3.3.3.6 XGBoost

Extreme Gradient Boosting (XGBoost) is a widely used algorithm for the Classification of large datasets with a minimal amount of time. There are many advantages of this algorithm which causes its popularity these days. It performs parallel computing due to which users get their results faster. XGBoost classifier outperformed and accomplished the most noteworthy results when compared with different classifiers. Parameters of the classifier can also be tuned to enhance the results.

$$\boldsymbol{F}(\emptyset) = \boldsymbol{L}(\emptyset) + \boldsymbol{\Omega}(\emptyset) \qquad (11)$$

$\boldsymbol{L}(\emptyset)$ and $\boldsymbol{\Omega}(\emptyset)$,$\emptyset$ refers to the different parameters in Eq. (11), Where $\boldsymbol{L}(\emptyset)$ is a differentiable convex loss function, and $\boldsymbol{\Omega}(\emptyset)$, is a regularized term that castigates complex frameworks [72].

3.3.3.7 Stochastic Gradient Descent(SGD)

Minimization of a function by checking the gradients of loss functions can be obtained by using the Gradient Descent Algorithm. After checking, it updates the weights of the function. To minimize the error by updating the weights, this algorithm is beneficial. These algorithms are also called *optimization algorithms*. The learning rate has to be given as an argument to the classifier to make the changes accordingly. The default value of the learning rate in the classifier is 0.01. The formula to update the weights is mentioned in Eq. (12)[73].

$$\boldsymbol{w} = \boldsymbol{w} + \boldsymbol{\alpha} * \left(\boldsymbol{Y} - \widehat{\boldsymbol{Y}}\right) * \boldsymbol{x} \tag{12}$$

Where x is the input variable, w is the weight, α is the learning rate, Y is the expected outcome, and $\widehat{Y}$ is the predicted outcome.

3.3.3.8 Perceptron

The perceptron algorithm works like a neural cell present in our body. It accepts the training data as a node. It consists of 2 parameters weights and biases and a function called *the activation function*. It runs several times in a loop, and every time it changes the value of weights and biases to minimize the loss between predicted and actual value, and the activation function decides that the particular neuron should be fired for the output layer or not. The below-mentioned Eq. (13) is used for calculating the activation function.

$$\boldsymbol{f}_{\boldsymbol{activation}} = \sum(\boldsymbol{w}_{\boldsymbol{i}} * \boldsymbol{x}_{\boldsymbol{i}}) + \boldsymbol{b} \tag{13}$$

Where *i* is the index number, $\boldsymbol{f}_{\boldsymbol{activation}}$ is the activation function, $\boldsymbol{w}_{\boldsymbol{i}}$is the weight for the i^{th} instance of the data, $\boldsymbol{x}_{\boldsymbol{i}}$ is the input for the i^{th} instance, and $\boldsymbol{b}$ is the bias for the activation function.

The prediction will be equal to one of the activation values that is greater than or equal to zero, and it will be zero if the activation value is less than zero.

3.3.3.9 Artificial Neural Network

Artificial neural networks can be considered as ANN are nothing more than NNs that can assist build DL frameworks. ANN is similar to the human brain NN [74-77]. The operational nodes of ANN are termed neurons, which in humans are linked with each of the neurons. These are organized in a configuration in layers [78-82]. This network may be designed with different layers like Input, Hidden and Output layers [83-85]. The input layer is the one that accepts in different ways, hidden levels are tightly related levels, vital level upon the level that can determine level efficiency by evaluating and manipulating the extraction of functionalities and trends, and the output layer has the most important output. [86-90]. The ANN configuration would look like as indicated in Fig-3.3.

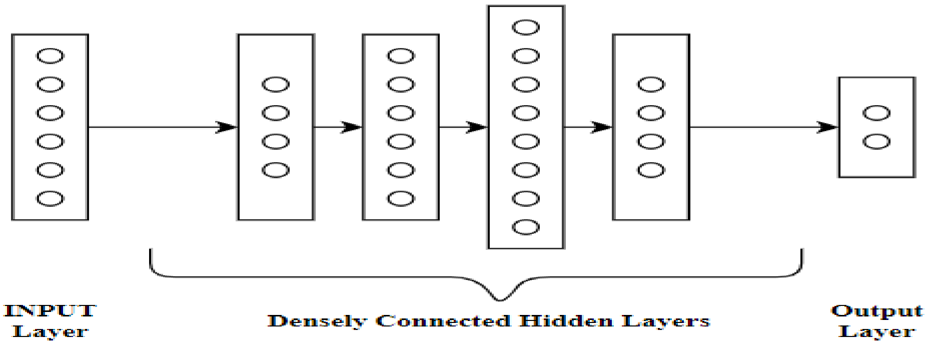


Figure 3.3: Rough representation of ANN

The weight of the information obtained by this network will be included and a weighted total of input values will be evaluated. [91-93]. The function generated is therefore a function of transference, T(x). The transference feature is also transferred to the activated feature, A(x), to get the required results. There are a variety of kernel functions such as linear, threshold, RAMP function, sigmoid, different ReLU functional structures, and softmax, etc. [93-97]. The sort of kernel function used following the study situations selected. As described in Fig-3.4, the operation of the ANN is visible to neurons [98-101].

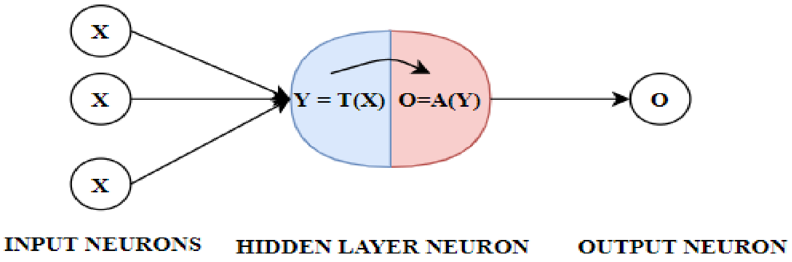


Figure 3.4: Demonstration of working of ANN

Take X1, X2,..., Xn as ANN entry and the appropriate parameters are W1, W2, ... and then two activities take place at each level. The source modification is carried on first, and afterward, the modified outcome is activated, A(X)[101-105]. These are assessed for every neuron on the relevant level utilizing the above-mentioned Eq. (14) and Eq. (15).

$$T(X) = W_1 * X_1 + W_2 * X_2 + \cdots + W_n * X_n + B = \sum_{i=1}^{n}(W_i * X_i) + B \qquad (14)$$

$$Output = A(T(X)) \qquad (15)$$

These networks are presently capable of resolving different dynamic challenges and needs increase with time. A wide range of applications from facial identification to strategic thinking is accountable for NNs. The more real-time situations, the more responsive the NN is. NNs can investigate and detect failures, thus increasing their capacity for effectiveness [106-110]. NNs are frequently significantly chosen for dynamic issue resolution. ANN's prominent implementations include image processing, audio recognition, language processing, translation, tracking, prediction, and anomaly detection [106].

Various methodologies have been incorporated in process of research over the present scenario with the aid of the NSL-KDD dataset[107]. These methodologies are considered right from data analysis with the aid of tools such as MS-Excel, and Tableau. Then, the utilization of machine learning methodologies implemented as mentioned in the previous section along with feature selection methodologies. All these frameworks are utilized for the identification and classification of various threats related to DDOS threats with the aid of the NSL-KDD dataset. One more methodology is implemented based on ANN along with GA[105-109]. GA plays a vital role in the identification of crucial features of the dataset and these features will be utilized as the input for the customized ANN for the identification and classification of various classes in the selected features of the NSL-KDD dataset[118]. These methodologies successful in the implementation of various mentioned methodologies, yet these methodologies can't give any explanation or reason for the generated predictions[120]. This thought drove the concept of explainable AI. An explainable AI-based framework is also generated for better recommendations along with the explanations or reasons for them.

3.3.3.10 Explainable AI

As mentioned earlier, explainable AI generates the interpretability of the model. Interpretability can be categorized into two classes such as locally concentrated interpretation and the other is globally concentrated interpretation[94-96]. The locally concentrated interpretation able to explain the logical reason for the obtained output for the corresponding input given to the model. The globally concentrated interpretation able to understand the structure of the model by looking at the overall structure of the model. The concept SHAP [98] plays a vital role in the enhancement of the interpretability of the IDS. This concept of methodology locally concentrated and globally concentrated interpretations in the same instance and this concept has strong theoretical and mathematical support when compared with other methodologies. The concept of SHAP [99] linked the concept of LIME [99] and the Shapley values [101].

LIME (Local Interpretable Model-Agnostic Explanation) [109-111] concentrates more on learning the local replacement model to evaluate individual forecasts. LIME produces a new modified dataset composed of permutated samples and also determines the accompanying forecasts of the black-box model, and then the interpretable model will be trained on the new modified data. Certain machine learning techniques such as linear regression, logistic regression, decision tree, and random forest are utilized as interpretable models[113-116]. A good local solution to the black box framework forecasts should be a local surrogate framework. And it can be evaluated as represented as mentioned in Eq, (16) follows:

$$\psi(a) = {}^{argmin}_{j \in J}\{\zeta(h, j, k^{a}) + \phi(j)\} \tag{16}$$

The notation in Eq. (15) represents j signifies the model of explanation for a sample of a, J signifies the possible set of explanations, ζ() signifies the loss function, h represents the original model, k^a signifies the weight aspect between the sampled and original data[117-120]. If the correlation between sampled data and original data is higher indicating that the weight will also be higher and vice versa and ϕ(j) signifies the complexity of function j. As per Eq. (15) LIME model trains the interpretable, local surrogate framework j on the obtained new dataset by decrementing the loss function, and then explores the prediction of a sample a by interpreting the local framework ψ(a). Shapley explained the evaluation methodology of obtaining Shapley values [121] and this methodology was utilized in game theory to ascertain that the proportion of each individual of the game offered to the success and this process can be more understandable utilizing the concepts related to predictions of machine learning methodologies[122]. The mean offering of an attribute value to the prediction in all possible combinations can be referred to as Shapley values.

$$\xi_i(g, y') = \sum_{x' \subseteq \{y'_1, y'_2, \ldots, y'_n\} \backslash \{y'_i\}} \frac{|x'|!(N - |x'| - 1)!}{N!} * [g(x' \cup y'_i) - g(x')] \tag{17}$$

The notations in Eq. (17) represents x' represents the subset of attributes that are utilized in the model, y' represents attribute values having a vector and the instances of this explained through y'_i, N represents the number of attributes considered, g(x') represents the prediction for attribute values in x', the evaluation of this prediction value involves masking out the ith attribute[122-126]. By drawing the random instances through simulation or the ith attribute's random values from the dataset. The three properties that abide by the Shapley values such as symmetry property, dummy property, and additivity property, and these properties can be represented as mentioned in Eq. (18) to Eq. (20) respectively.

$$f(x' \cup y'_i) = f\left(x' \cup y'_j\right),\ for\ all\ x' \subseteq \{y'_1, y'_2, \ldots, y'_n\} \backslash \{y'_i, y'_j\} \tag{18}$$

$$f(x' \cup y'_i) = f(x'), for\ all\ x' \subseteq \{y'_1, y'_2, \ldots, y'_n\} \backslash \{y'_i\} \tag{19}$$

$$f(x' \cup y'_i) = f'(x' \cup y'_i) + f^2(x' \cup y'_i) \text{ then } \xi_i(g, y') = \xi_i(f^1, y') + \xi_i(f^2, y') \tag{20}$$

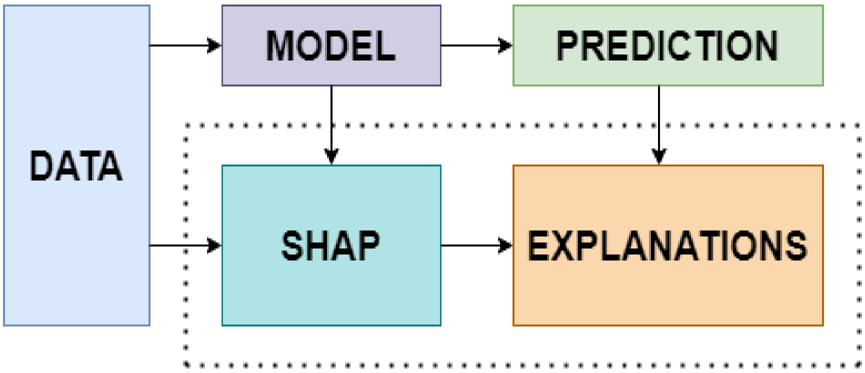


Figure 3.5: Generalized SHAP System

Since there are 2^k possible variations of attribute values, high computing time is needed to evaluate Shapley values. Lundberg[127] proposed SHAP, a combined method for evaluating predictions. It depicts a case x estimate by estimating the relationship between each function and the estimation. The LIME techniques concerning Shapley values can be interpreted with the clarification provided by the linear model. It ties together the two techniques LIME and Shapley principles. Using SHAP values to divide into positive or negative groups, the involvement of each function of the framework can be explained[128]. The key benefits of using SHAP values are that they can be calculated for any framework with only a simple linear model, and each set of data records would have its own set of SHAP values[129]. The following equation can be used to describe an example of the dataset utilizing a given SHAP value.

$$f(C') = \xi_0 + \sum_{i=1}^{N} \xi_i C'_i \qquad (21)$$

The notations in Eq. (21) represents f is known as explanation model, $C'$is known as coalition vector with values 0 and 1 for each of the instances of data, 1 indicates the instances in the new dataset is the same as that of the original dataset, 0 indicates the instances in the new dataset is different from that of the original dataset, that N indicates the size of the maximum coalition, ξ_i is the feature contribution for the attribute i for an instance of the dataset and it is known as Shapley value[123].

<u>*Deep Neural Network:*</u> For the framework to predict different anomalies in the KDD-NSL dataset for the detection of DoS assaults, a DNN is used[130]. Input as a sample can be signified as X that is of the form $\mathbb{R}^n$ and each sample i of the dataset related to an attribute can be represented as x_i, thus the dataset can be characterized as $X =$

$\{x_i\}_{i=1}^{n}$ and the consequent labels are characterized by Y. The classification function based DNN mapping can be represented as $g: \mathbb{R}^n \rightarrow \mathbb{R}^+$[93]. In the DNN framework, several layers are concerned, including input, output, and multiple hidden layers with different neurons in each of these layers. Any of these secret layers' neurons can be triggered, as seen mathematically in Eq (22).

$$h_i^{k+1} = f\left(\sum_j h_j^{(k)} w_{ji}^{(k,k+1)} + b_i^{(k+1)}\right) \qquad (22)$$

The notations in Eq. (6) represents h_i^{k+1} is the activation of (k + 1)th layer of the ith neuron, $w_{ji}^{(k,k+1)}$ is the weight of the connection between jth neuron of kth layer and ith neuron of (k+1)th layers, and $b_i^{(k+1)}$ is the bias of the ith neuron of the (k+1)th layer, $f(.)$ is the activation function. In this framework, the activation function utilized is ReLu and it can be represented as in Eq. (23).

$$f(a) = Max(0, a) \qquad (23)$$

For identification aspects relevant to the given inputs, the Softmax function is utilized in the output level, and this function to activate can be expressed as in Eq (24).

$$f_{Prob}(Y = y_i|X) = \frac{e^{h_i}}{\sum_k e^{h_k}} \qquad (24)$$

The h_i value acquired from the above-mentioned activation function f (.) is represented by the notations in Eq. (7). SHAP values and their descriptions will be created based on the expected category classification. Explainable AI combines the whole mechanism of explainable AI with deep neural networks as a training platform to create an explainable AI framework.

3.3.3.11 Explainable AI-Based Proposed Model

The suggested structure, as well as its flowchart implementation, was discussed in this section to improve the explainability of an IDS system. This interpretable IDS structure, as well as the framework's consistency, are important for any user. As a result, an IDS architecture can be developed, as well as clarity, which is critical at this point. The suggested framework's flowchart is seen in Figure 3.6. This flowchart can be broken down into two parts. The traditional IDS paradigm is on the left, and the right section is used to achieve explainability corresponding to the traditional IDS application's estimation.

In the traditional IDS paradigm, a DNN model is utilized for both preparation and forecasting using the KDD-NSL dataset. The forecast classifiers are compared to the description data, which may serve as a guide as well as a tool for specialists working with intrusion detection systems. The suggested thesis focuses on improving the explainability of the IDS framework's forecasts. As a result, both local and global explainability offer a proper account for the IDS framework's received forecasts. Two techniques are used to generate global explainability. The first approach examines the fundamental characteristics of IDS, while the second methodology describes the relationship among attribute values and their effect on the predicted outcome. The local explainability offers a justification for the output provided by the IDS system as well as the importance of input attributes for the IDS model's predictions.

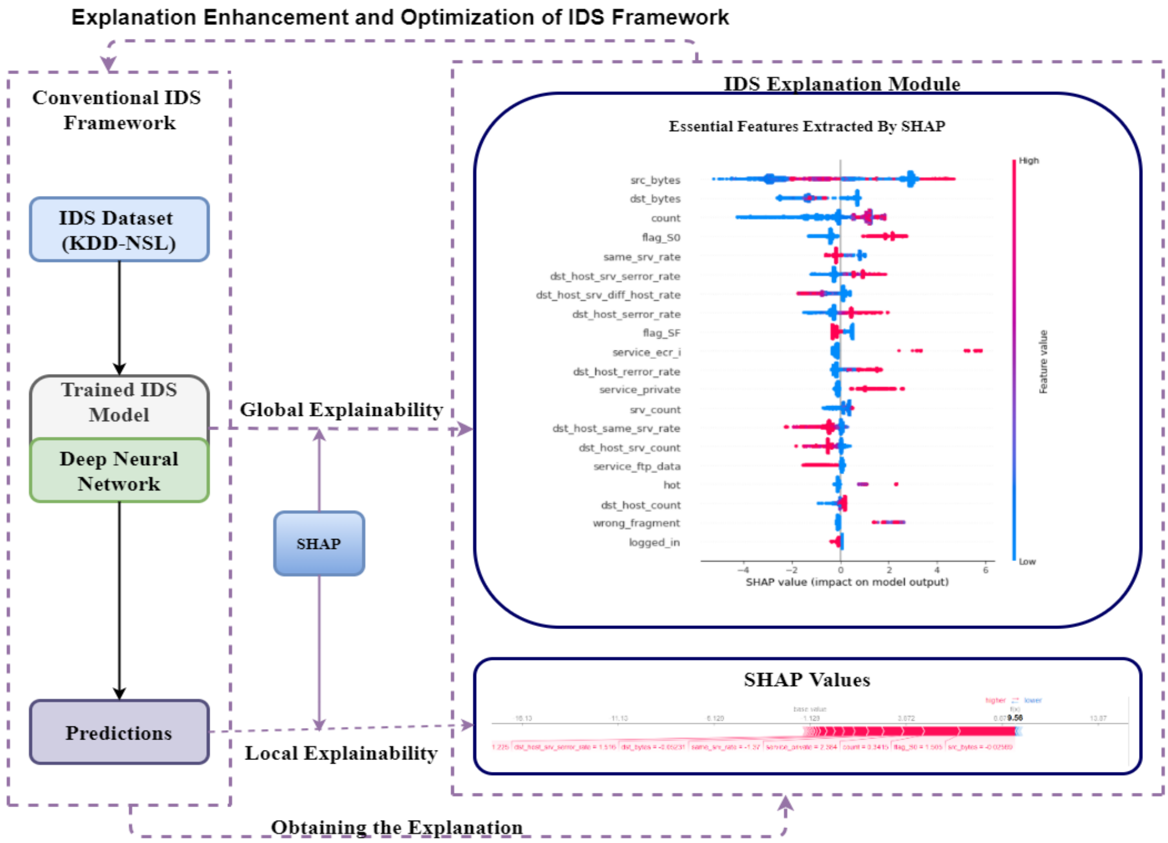


Figure 3.6: The Proposed Framework Flowchart Overview

The suggested structure, as seen in Fig. 3.6, should be used to significantly increase the clarity of the IDS system. With the help of local and global explainability, the experts working on this system will be able to verify the forecasts acquired from the IDS system. Besides, the suggested architecture makes utilizes a deep NN. As a result, experts will change the parameters of the model used in the IDS system to achieve the optimal estimation and favorable interpretation by recognizing the discrepancies between the interpretation and classifiers acquired.

3.4 Performance Evaluation Metrics

Various assessment criteria are taken into account when estimating the success of the system that is used to identify and classify various DDOS risks. These assessment criteria are critical in assessing the application's effectiveness. The NSL-KDD dataset is split into three sections: preparation, testing, and validation. Understanding the true positive, true negative, false positive, and false negative is important since this study of structures deals with classification issues. Accuracy, precision, recall, and F1-score are all calculated using these.

True Positive: The data point that predicted belongs to a positive class as exact as the actual positive class. Such kind of collection of data points obtained under this scenario is called True Positives.

True Negatives: The data point that predicted belongs to a negative class as exact as the actual negative class. Such kind of collection of data points obtained under this scenario is called True Negatives.

False Positives: The data point that predicted belongs to a positive class which is different from the actual negative class. Such kind of collection of data points obtained under this scenario is called False Positives.

False Negatives: The data point that predicted belongs to a negative class which is different from the actual positive class. Such kind of collection of data points obtained under this scenario is called True Positives.

Confusion Matrix: It is a table-like structure that provides information that the size of data points that belong to various classes. From the confusion matrix, the number of true positives, true negatives, false positives, and false negatives may be derived.

Accuracy: It is a metric utilized for identification of the framework trained well or not using the training dataset and thereby using the testing dataset and confusion matrix, accuracy will be calculated as mentioned in Eq. (25). It represents the number of accurately classified data points over the total number of data points. This value lies between 0 and 1 and to obtain in percentage the value multiplied by 100. In the equation, T_N indicates True Negatives, T_P indicates True Positives, F_N indicates False Negatives, and F_P indicates False Positives.

$$Accuracy = \frac{T_N+T_P}{T_N+F_P+T_P+F_N} \tag{25}$$

Precision: It is a metric utilized for identification of the framework trained well or not using the training dataset and thereby using the testing dataset and confusion matrix, precision will be calculated as mentioned in Eq. (26). This value lies between 0 and 1 and to obtain in percentage the value multiplied by 100. Ideally, this value nearer to 1 represents the better efficiency of the model. This metric mainly deals with the identification of false positives high or not. In the equation, T_N indicates True Negatives, T_P indicates True Positives, F_N indicates False Negatives, and F_P indicates False Positives.

$$Precision = \frac{T_P}{F_P+T_P} \tag{26}$$

Recall: It is a metric utilized for identification of the framework trained well or not using the training dataset and thereby using the testing dataset and confusion matrix, precision will be calculated as mentioned in Eq. (27). This value lies between 0 and 1 and to obtain in percentage the value multiplied by 100. Ideally, this value nearer to 1 represents the better efficiency of the model. This metric mainly deals with the identification of false positives high or not. In the equation, T_N indicates True Negatives, T_P indicates True Positives, F_N indicates False Negatives, and F_P indicates False Positives.

$$Recall = \frac{T_P}{F_N+T_P} \tag{27}$$

F1-Score: It is a metric utilized for identification of the framework trained well or not using the training dataset and thereby using the testing dataset and confusion matrix, precision will be calculated as mentioned in Eq. (28). It is the mean value of accuracy and recall and is more accurate than accuracy. This value lies between 0 and 1 and to obtain in percentage the value multiplied by 100. Ideally, this value nearer to 1 represents the better efficiency of the model. In the equation, T_N indicates True Negatives, T_P indicates True Positives, F_N indicates False Negatives, and F_P indicates False Positives.

$$F1_{score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{28}$$

3.5 Summary

This chapter discussed the various aspects in detail of the NSL_KDD dataset and about the various features in that data set and more description will be mentioned in chapter 4. Besides the discussion of datasets, various methodologies that are utilized in the present research work is discussed related to machine learning and artificial neural network. Particularly, the feature selection methodologies are attached to these methodologies to obtain better efficiency in the identification and classification of various threats of DDOS threats. For obtaining the reasoning for the predicted classes explainable AI is also implemented.Overall the proposed methodology has obtained promising results but it still computation time can be improved. The system can be improvised in such the manner that the computation time required to train the model can be minimized.

Chapter 4

Experimental Analysis

4.1 Overview

This chapter mainly focused on the discussion of the various results obtained through the conducted research. This discussion mainly includes the results obtained from the analysis of the NSL-KDD dataset, thereby the results obtained from the feature selection methodology. The main goal of this research is to identify and classify the network threats made through various frameworks. Of them, one is dealing with various machine learning methodologies along with various feature selection methodologies results are compared. The second framework is based on the ANN along with the GA and the results of this framework are compared with the machine learning methodologies log with GA methodology. Finally, the framework is built on an explainable AI concept that generates the output along with the explanation. In every research framework, I tried my best to incorporate the novelty to attain the best results.

4.2 Analysis of NSL-KDD Dataset

This segment discusses the research based on the datasets KDD and NSL-KDD, as well as the relevant recommendations to be taken based on the analysis performed on the dataset in question. In this study, we looked at KDD and NSL-KDD, which are two different datasets. NSL-KDD is removed from the KDD dataset once more. NSL-KDD is a subset of the KDD dataset that accounts for around 20% of the total dataset. NSL-KDD of the results, which was subjected to a comprehensive review in terms of DDOS disruptions as well as normal activities. Apache2, return, ground, Neptune, mailbomb, pod, processtable, smurf, teardrop, udpstorm, and worm are all sub-classes of the

DDOS category. As seen in Table 4.1, a description of the datasets such as KDD and NSL-KDD is given. As seen in fig. 4.1, the same detail can be visualized.

Table 4.1: Summary of the KDD & NSL-KDD datasets

Dataset	**Number of Records**		
	Total	Normal	DDOS
KDD	488735	97277	391458
NSL-KDD	113270	67343	45927

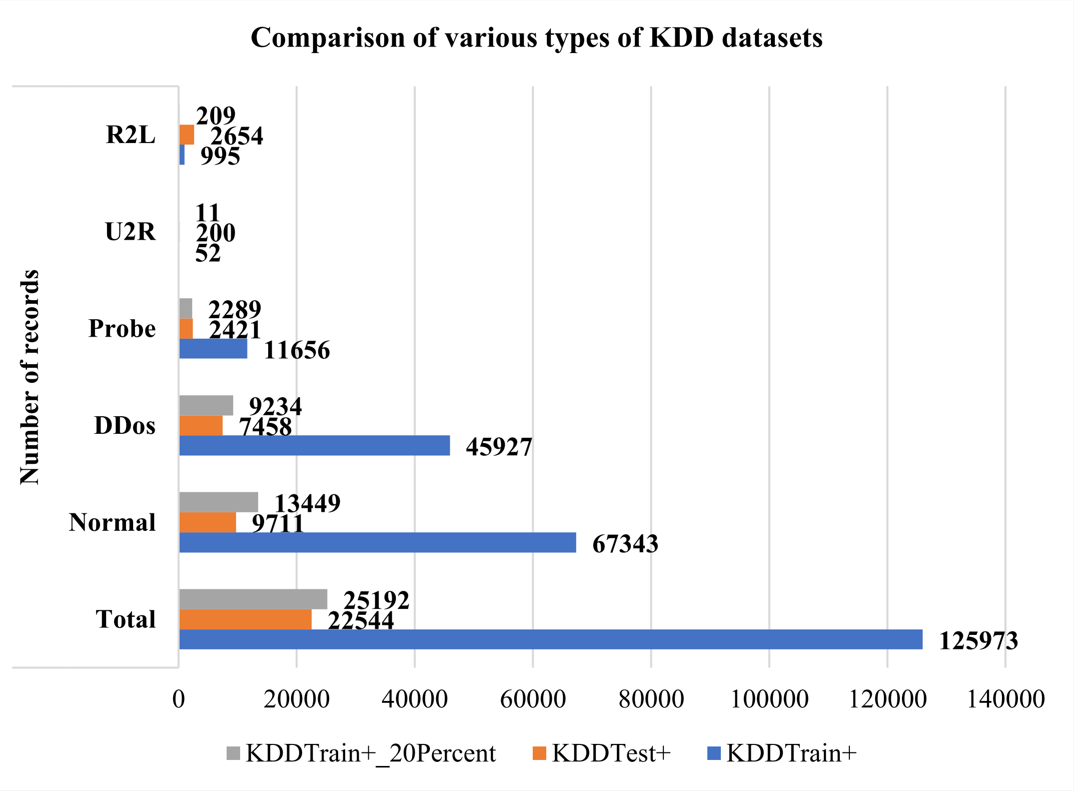
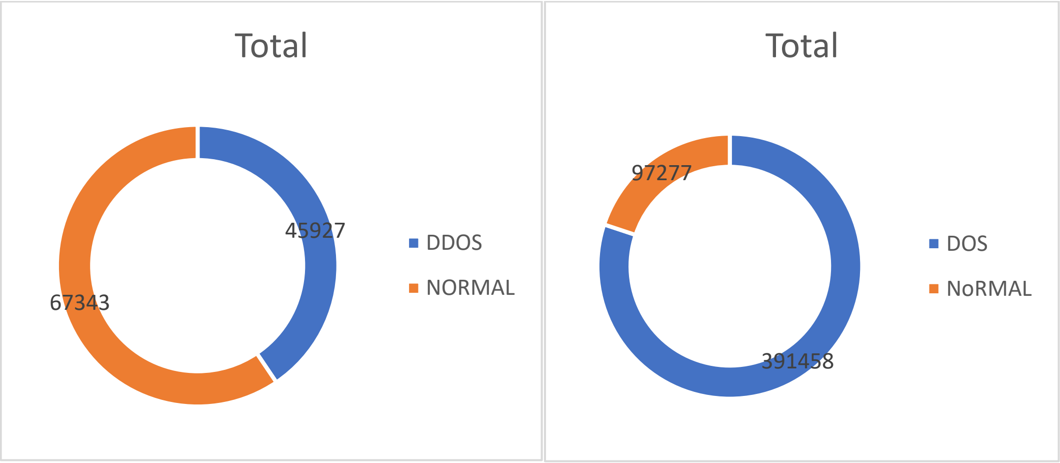


Figure 4.1: Graphical comparison among the various forms of KDD datasets

Figure 1 shows that normal behaviors and DDOS risks account for the majority of any dataset. As a result, it is important to investigate and examine the effects of DDOS attacks on the network and its numerous components. Consider the ratio of DDOS risks in both of these databases, as well as the distribution of typical activities. As seen in table-4.1, the KDD dataset contains 391458 DDOs threats, while the NSL-KDD dataset contains 45927 DDOs attacks. Figures 4.2(a) and 4.2(b) depict the only distributions of DDOS and regular events in the NSL-KDD and KDD datasets, respectively.



(a) (b)

Fig. 4.2: DDOS vs Normal activities distribution for NSL-KDD & KDD Datasets

Fig 4.2 shows that the dataset NSL-KDD was very weak in comparison to all other datasets, while the KDDT dataset was far bigger. Evaluating the previous dataset is insufficient for generalizing the different facets of the network, and analyzing the latter dataset with MS Excel is impossible due to its larger scale. As a result, the NSL-KDD data is of a medium scale, and the guidelines can be applied to a network. As seen in fig. 4.3, the distribution of DDOS and Normal behaviors according to guidelines is based on the dataset NSL-KDD Percent.

Fig 4.4 depicts the distribution of DDOS and Normal behaviors through the protocols in the NSL-KDD dataset. The most significant influence of DDOS attacks can be seen in the TCP protocol, accompanied by UDP, and then the ICMP protocol, which has the least impact on DDOS. However, these merely reflect the number of instances in each of the protocols; a distinction would be more useful if the percentage was taken into account. The ratio is determined using the formula in Eq (29).

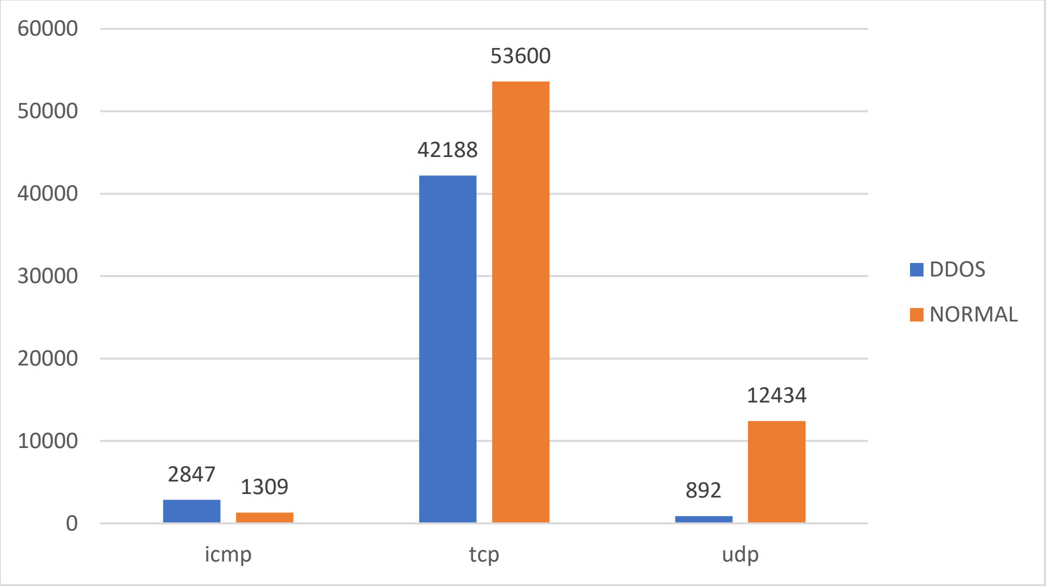


Figure 4.3: Distribution of DDOS and Normal activities in NSL-KDD dataset

$$Ratio_{category} = \frac{Total\ number\ of\ samples\ registered\ under\ a\ category\ and\ a\ protocol}{Total\ number\ of\ samples\ registered\ under\ a\ protocol}$$

(29)

DDOS and Normal operations are represented by category, and TCP, UDP, and ICMP are represented by protocol. The ratios will show how powerful a class (DDOS or Normal) is in a certain protocol. As seen in fig. 4.4, the distribution of such ratios is depicted.

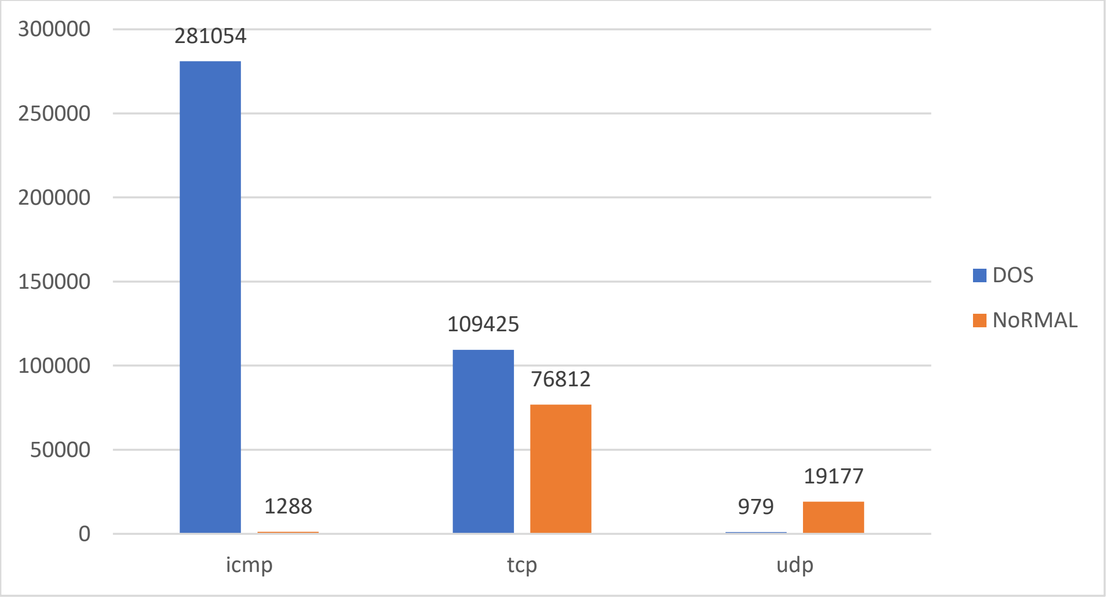


Fig. 4.4: The distribution of proportions of DDOS and Normal activities in the KDD dataset

In contrast to the previously mentioned influence of DDOS on protocols, the DDOS influence was greater in the case of both ICMP and TCP. Even so, the DDOS influence is very minimal in the case of UDP, implying that UDP is much superior to the other two protocols. Consider the situation in which the groups (DDOS and Normal) affect each of the flags by looking at the proportions. The distribution of DDOS and Normal events according to the different flags in NSL-KDD is depicted in fig 4.5. From fig. 4.5, it is clear that DDOS has a greater impact on flags like S0, RSTO, and REJ. The S0 flag indicates that the communication attempt was detected but no response was received, the RSTO flag indicates that the Founder reset the link, and the REJ flag indicates that the link request was denied.

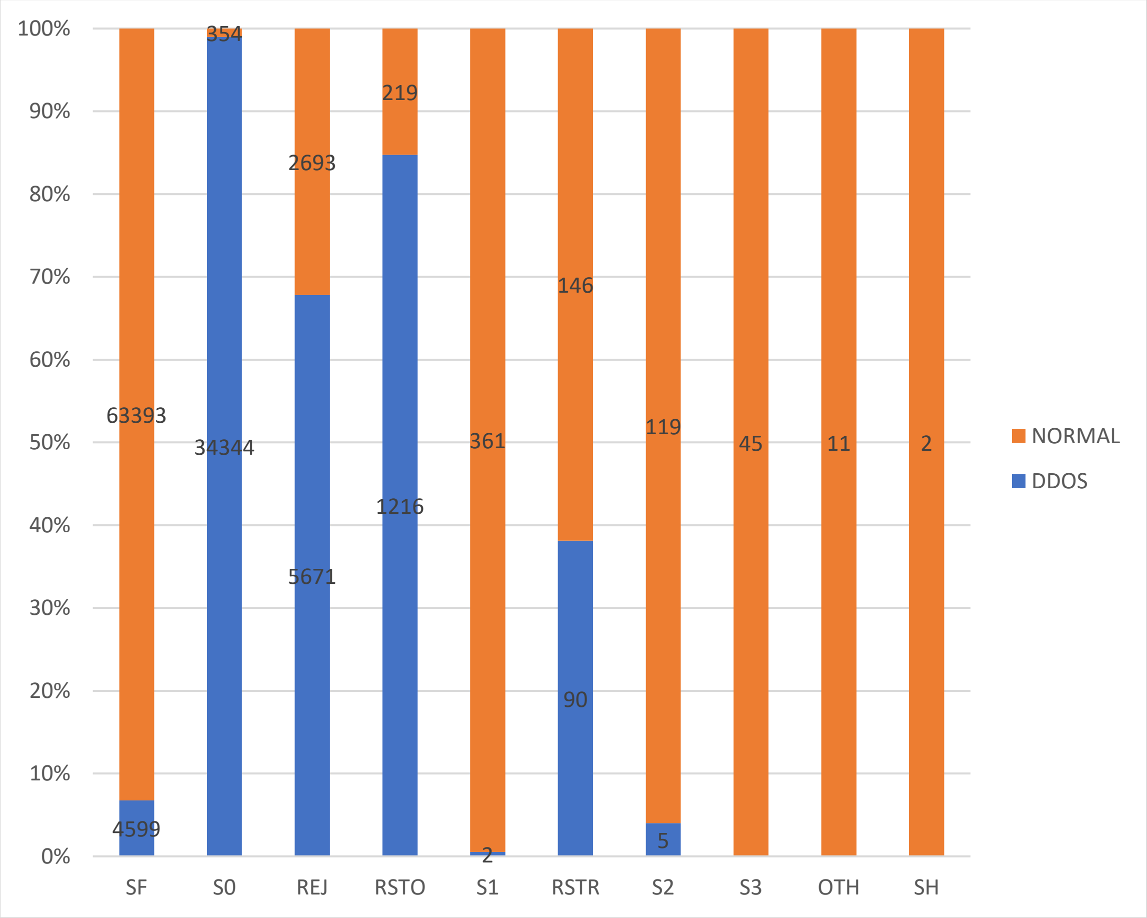


Figure 4.5: Distribution of DDOS and Normal activities as per the various flags in NSL-KDD

The dataset considered, NSL-KDD, further analyzed using the Tableau tool. At first, tried to analyze the distribution of various categories in the NSL-KDD information collection as mentioned in fig 4.6. From this plot, one can identify that the Neptune class is the highest frequent class in the dataset with a percentage of 64.53%, which is followed by normal class, smurf class, satan class, portsweep class, saint class, teardrop class, and so on with corresponding percentages are 15.19%, 9.63%, 8.29%, 0.80%, 0.68%, 0.44%. The remaining classes are almost insignificant.

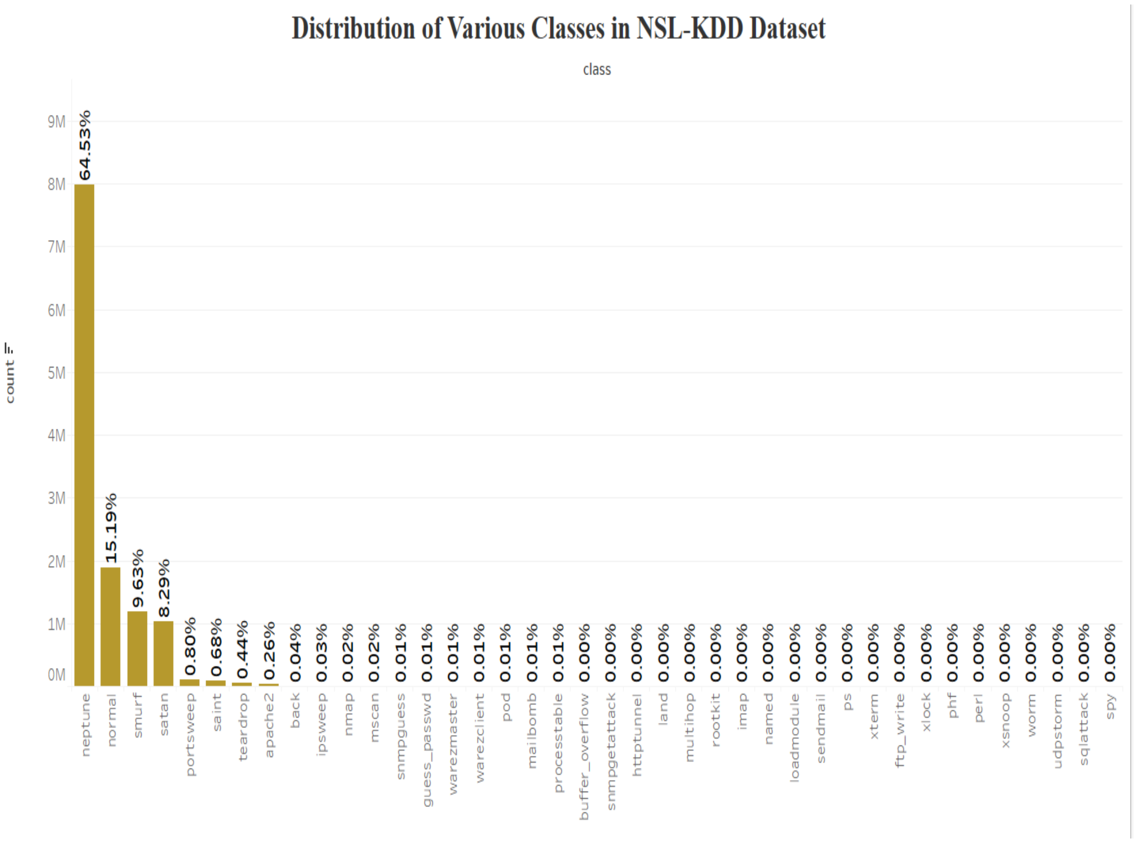


Figure 4.6: Distribution of various classes in NSL-KDD Dataset

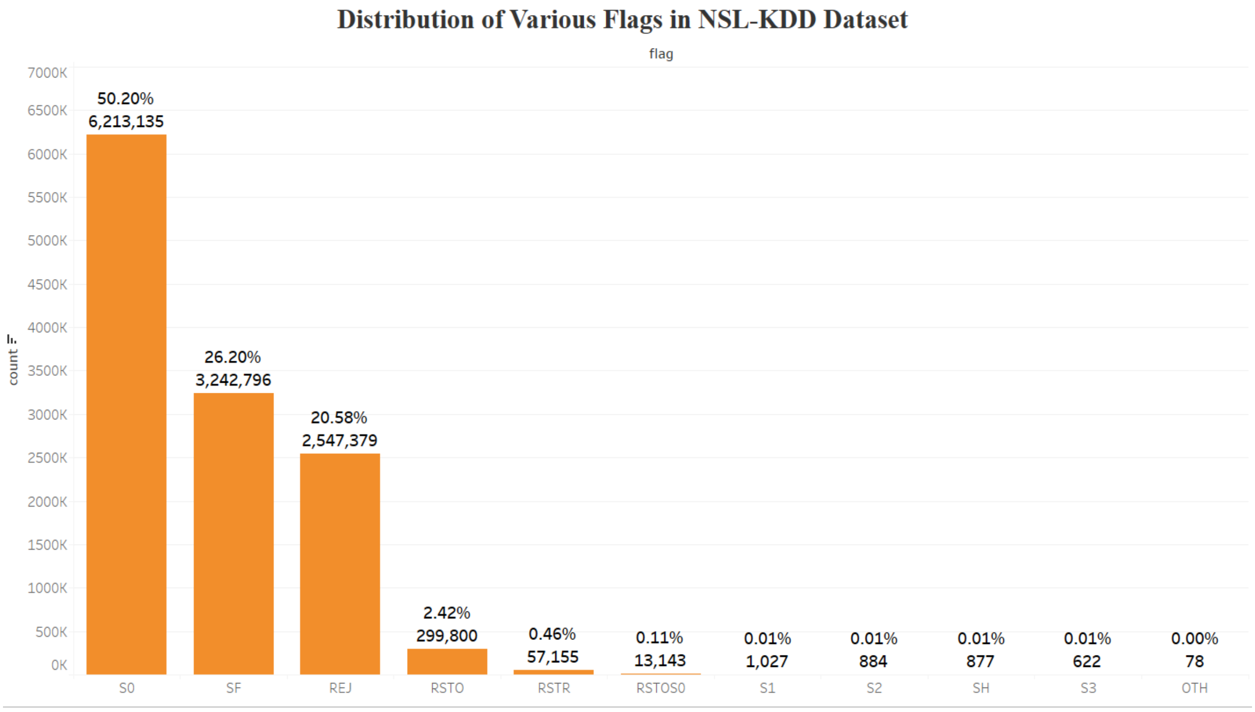


Figure 4.7: Distribution of various flags in NSL-KDD Dataset

The next analysis made is to identify the distribution of various flags in the NSL-KDD dataset as mentioned in fig. 4.7. The major contribution of the flags in the dataset are S0, SF, REJ, RSTO with corresponding percentages are 50.20%, 26.20%, 20.58%, 2.42%. The remaining flags contribute very minimally to the utilized dataset. Then the analysis of the NSL-KDD dataset was further extended to identify the distribution of various protocols as mentioned in Fig. 4.8. From this plot, one can understand that the tcp protocol is a major contribution to the utilized dataset. The tcp contributes about 77.64%, the icmp contributes about 9.75% and the udp contributes about 12.60% to the utilized dataset.

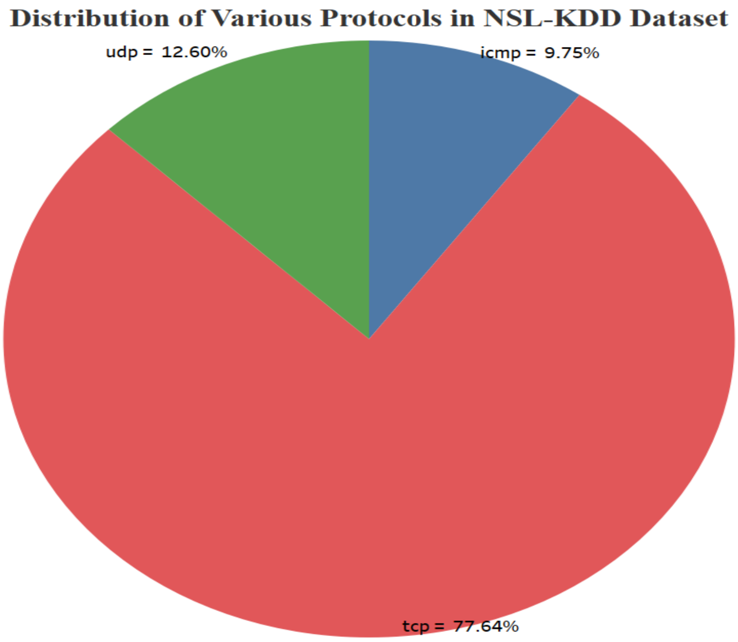


Figure 4.8: Distribution of various Protocols in NSL-KDD dataset

Then the analysis of the NSL-KDD dataset was further extended to identify the distribution of various services as mentioned in fig 4.9. From this plot, one can understand that the significant services in the NSL-KDD are private, ecr_i, domain_u, other, and http. The private service, the ecr_i service, the domain_u service, the other service, and the http service contribute 30.00%, 9.65%, 8.67%, 6.91%, and 4.93% respectively to the dataset. the remaining services are insignificant in the dataset.

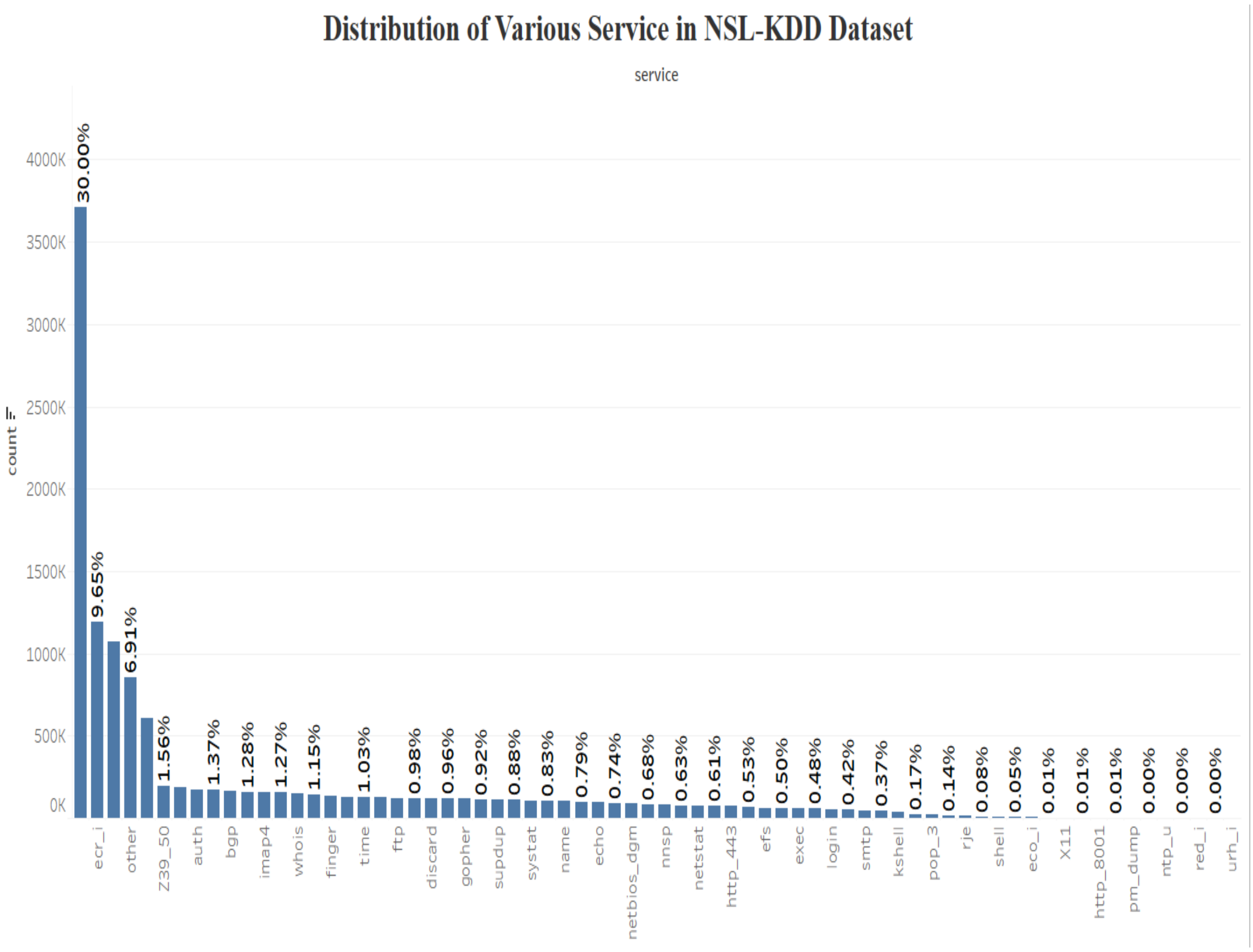


Figure 4.9: Distribution of various services in NSL-KDD Dataset

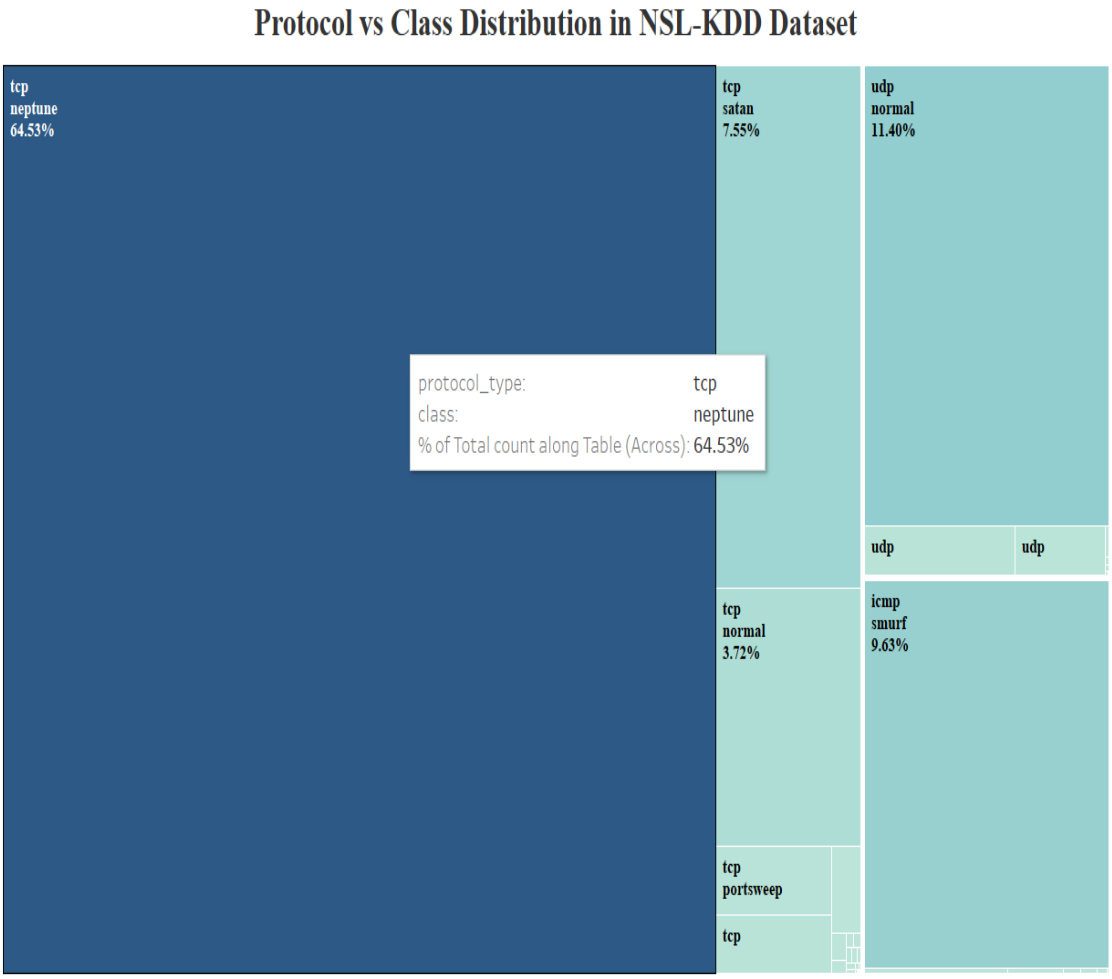


Figure 4.10: Distribution of classes concerning protocol in the NSL-KDD dataset

Then the analysis of the NSL-KDD dataset was further extended to identify the distribution of various classes concerning a protocol in the NSL-KDD dataset as mentioned in fig. 4.10. The tcp protocol the major significant class is the neptune class with a contribution of 64.53%. The icmp protocol the major significant class is the smurf class with the contribution of 9.63%. The udp protocol the major significant class is the normal class with the contribution of 11.40%.

Then the analysis of the NSL-KDD dataset was further extended to identify the distribution of various services concerning a protocol in the NSL-KDD dataset as mentioned in fig. 4.11. The tcp protocol the major significant service is the private service with a contribution of 27.28%. The icmp protocol the major significant service is the ecr_i service with the contribution of 9.65%. The udp protocol the major significant service is the domain_u service with the contribution of 8.67%.

Then the analysis of the NSL-KDD information collection was further extended to identify the distribution of various services concerning a class in the NSL-KDD dataset as mentioned in fig. 4.12. The neptune class protocol the major significant service is the private service with a contribution of 24.45%. The normal class the major significant service is the domain_u service with the contribution of 8.67%. The smurf class the major significant service is the ecr_i service with a contribution of 9.63%. The satan class the major significant service is the other service with the contribution of 5.28%

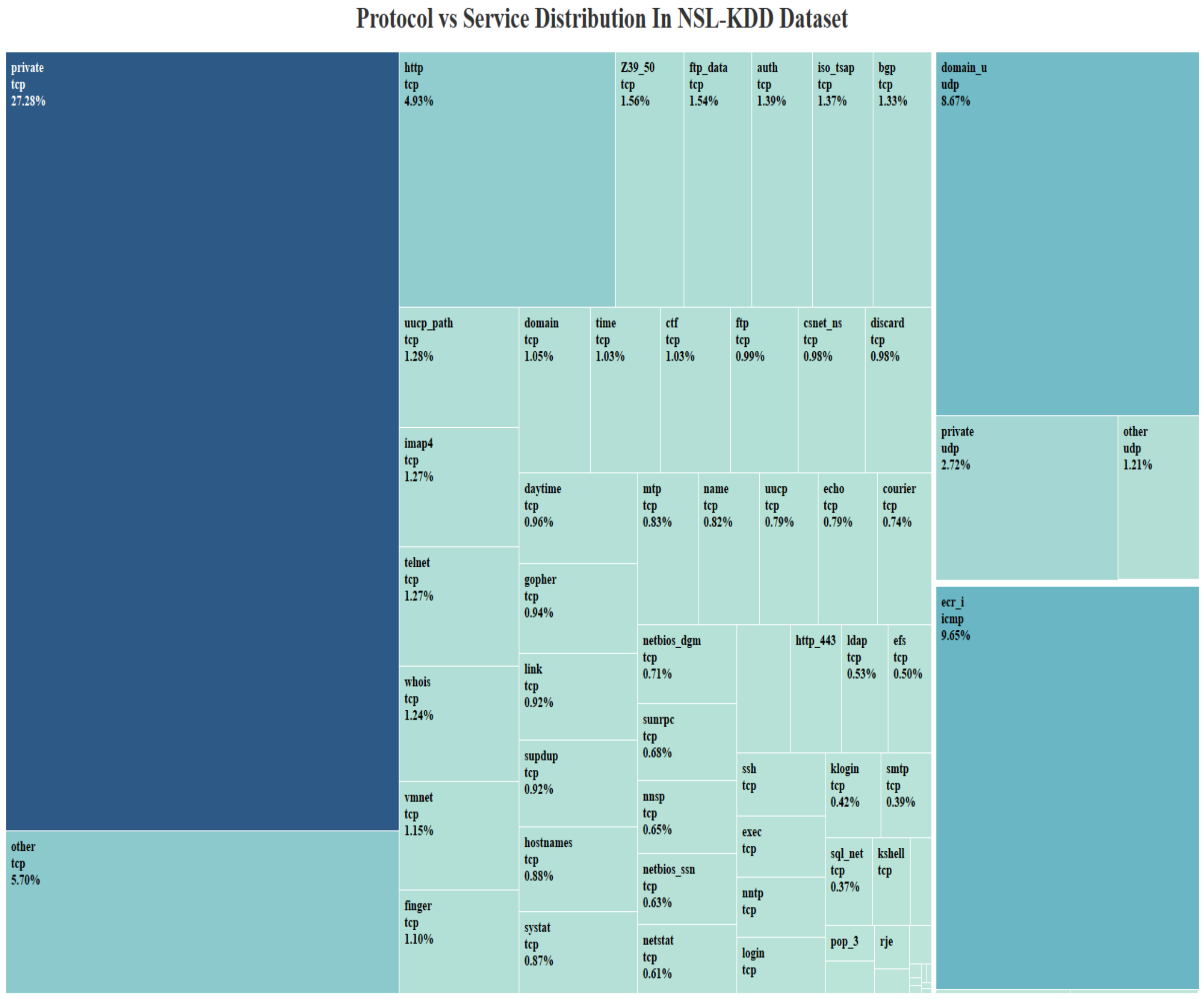


Figure 4.11: Distribution of services concerning protocol in the NSL-KDD dataset

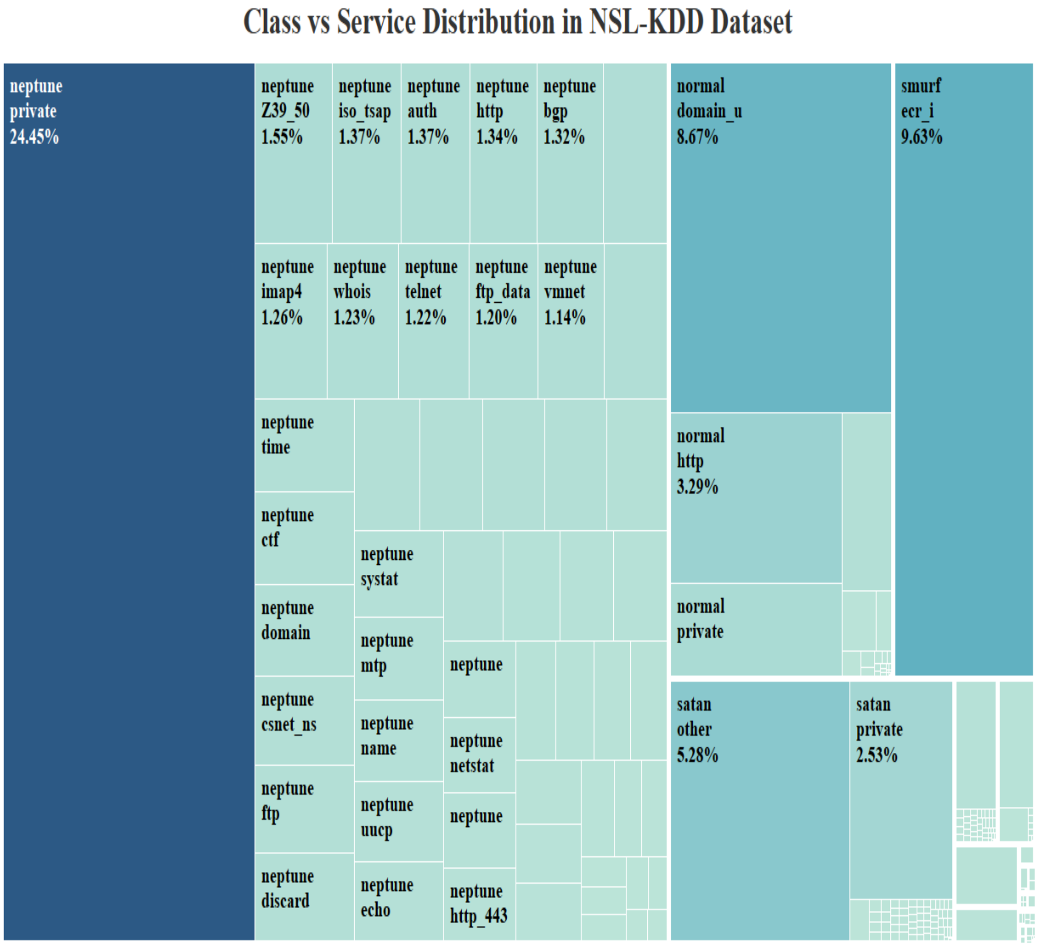


Figure 4.12: Distribution of services concerning class in the NSL-KDD dataset

4.2.1 Recommendations Based on Analysis

Following the guidelines, those proposals were revoked based on the above-mentioned review. The level of assault was moderate in the case of the ICMP protocol, and resources like Eco_i and Urp_i are common, but Ecr_i is critical, as seen in Figure 6. In the case of the TCP protocol, the level of assault was moderate, and services like HTTP, SMTP, IRC, X11, and FTP information are usual, whereas flags like SF and Src byte are always normal, S0 is under full attack, and RSTO is under 80% of probability of being under assault. As previously said, UDP has a very low risk of being targeted. As seen in table-4.2, these descriptions can be more expanded. Similarly, the review was carried out on utilities as well as different flags, as seen in tables 4.3 and 4.4, respectively.

Table 4.2:Analysis of protocol attack

S. No	Protocol_Type	Attack/ Normal	Service	Flag	Continous
1	TCP	Mix	Private→Attack Http→Normal	S0→attack	Src_byte→ normal
2	UDP	Normal (94%)	Private→Attack All Other→Normal	SF→attack	Src_byte→ Balanced
3	ICMP	Mix	Ecr_i→Attack Tim_i→Attack All other→ normal	SF→attack	Src_byte→ attack

Table 4.3: Analysis of various services

Service	Attack/Normal
Http	normal
Private	attack
Domain_u	normal
Smtp	normal
Ftp_data	normal

Table 4.4: Analysis of various flags

Flag	Attack/Normal	Service
SF	Other all Normal	Ecr_i → attack Private → attack
S0	Almost All Attack	http → attack Private → attack
REJ	Balanced	Private → attack
RSTR	Other all Normal	http → attack
RSTO	Almost All Attack	uccp → attack telnet → attack

4.3 Machine Learning Based Framework Along With Feature Selection Results

The association among the attributes can be interpreted utilizing the Pearson correlation coefficient, which was discussed in chapter 3. The coefficients' values are always between -1 and 1. The values of -0.5 and +0.5 levels have demonstrated a strong association in the KDD dataset. A symmetric matrix can be formed by calculating this association and placing certain values in a matrix between each pair of values accessible in the dataset. Correlating characteristics are removed as a result of this discovery. Attributes with a value greater than or equal to 0.7 and a value less than or equal to -0.7 are removed. Until lowering the correlated attributes, there were 41 attributes; after dropping them, only 28 attributes are remaining. For the KDD dataset, the correlation matrix can be expressed as shown in fig. 4.13.

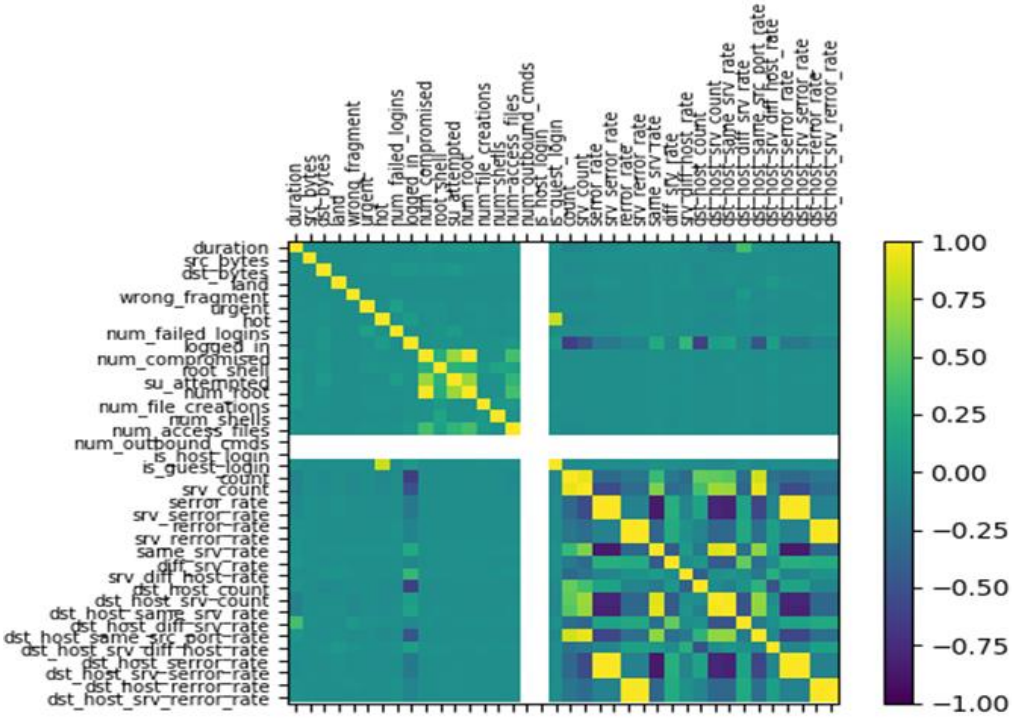
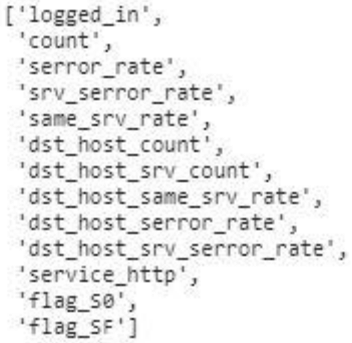


Figure 4.13: Plot of Correlation Matrix

When the Univariate approach is used independently, Figure 4.14 illustrates the 13 chosen attributes of the 41 attributes of the KDD dataset.



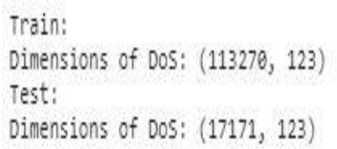
```
['logged_in',
 'count',
 'serror_rate',
 'srv_serror_rate',
 'same_srv_rate',
 'dst_host_count',
 'dst_host_srv_count',
 'dst_host_same_srv_rate',
 'dst_host_serror_rate',
 'dst_host_srv_serror_rate',
 'service_http',
 'flag_S0',
 'flag_SF']
```

Figure 4.14: Features selected using Univariate Feature Selection Technique

Each of the framework's classifiers is trained eleven times with a different number of attributes each cycle. Furthermore, K-Fold cross-validation is used to evaluate the whole analyzed construct. Since the findings obtained by this methodology are often less skewed, it is the most accurate validation methodology. For K-fold cross-validation, the feature cross-validation score() is used with 10 folds. The research is carried out on a device with 16 gigabytes of RAM and an AMD Ryzen 9 4900H with a 6 GB Nvidia GeForce gtx 1660 graphics processor running at 3.30 GHz.

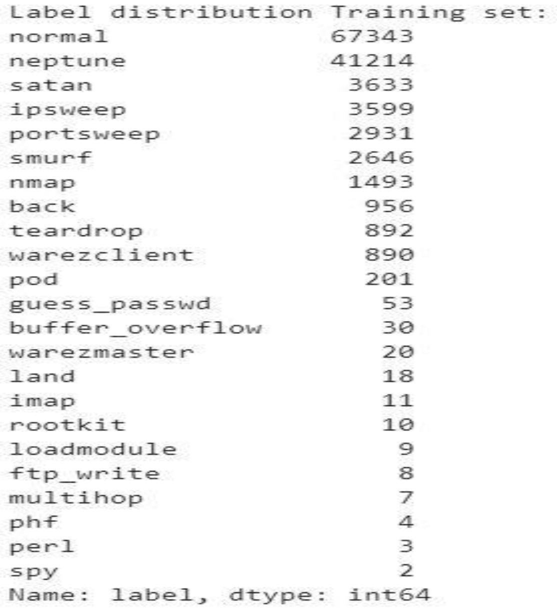
Following a Correlation-based attribute collection on the dataset, 28 attributes are listed, and the framework is trained on these 28 attributes for each classifier. The framework is then conditioned with each of the classifiers after 17 attributes are chosen from the 28 attributes using four-function scaling methodologies. Following that, 11 attributes are chosen from the 28 using four-function scaling methods, and the framework is trained using each of the classifiers. Eventually, all of these approaches are utilized in a stack, such that first, 23 attributes are selected from the 28 attributes utilizing Univariate Attribute Selection, and then 20 best attributes are selected from those 23 attributes utilizing Recursive Attributive Elimination, and from those 20 attributes, 16 attributes are selected using Principal Component Analysis, and from those 16 attributes, 11 attributes are selected utilizing Linear Discretion. Finally, to conduct attribute scaling as a stack, two attribute selection methods, Recursive Attribute Elimination, and Linear Discriminant Analysis are selected. Recursive Attribute Elimination selects the first 17 attributes from a total of 28 attributes, and Linear Discriminant Analysis selects the 11 best attributes from those 17 attributes, before implementing the qualified framework of each of the classifiers. The frameworks are evaluated using K-Fold Cross-Validation. The training set is divided into 10 folds (the default value of the parameter), and the framework is trained on 9 folds before being tested on the last fold. Furthermore, it produces ten various accuracies as a result of the ten various flips, and then it measures the mean of the accuracies to obtain the framework's final accuracy. In Table 4.2, the acquired precision is compared to that of other current methodologies.

The XGBoost classifier is a classification algorithm that combines several tree variations with lower differentiation performance to produce a significant right and limited False Positive item by standard framework iteration. XGBoost, on the other hand, has the potential to scale past billions of good instances by consuming even less capital than current approaches. It can also be calculated on the out-of-core, which saves memory resources on the processor [131]. The train and test dataset examples are depicted in detail in Figure 4.15. The count of mark incidents present in the train and test datasets is seen in Figures 4.16 and 4.17. Figures 4.18 and 4.19 show the first five reference values from the instruction and trial datasets, respectively. Figure 4.20 depicts the entire model. Fig 4.21 and fig. 4.22 show the accurate findings acquired utilizing 17 and 11 attributes, respectively. Figure 4.23 depicts the accuracies of all classifiers, while tables 4.5 and 4.6 show the related effects in tabular form.



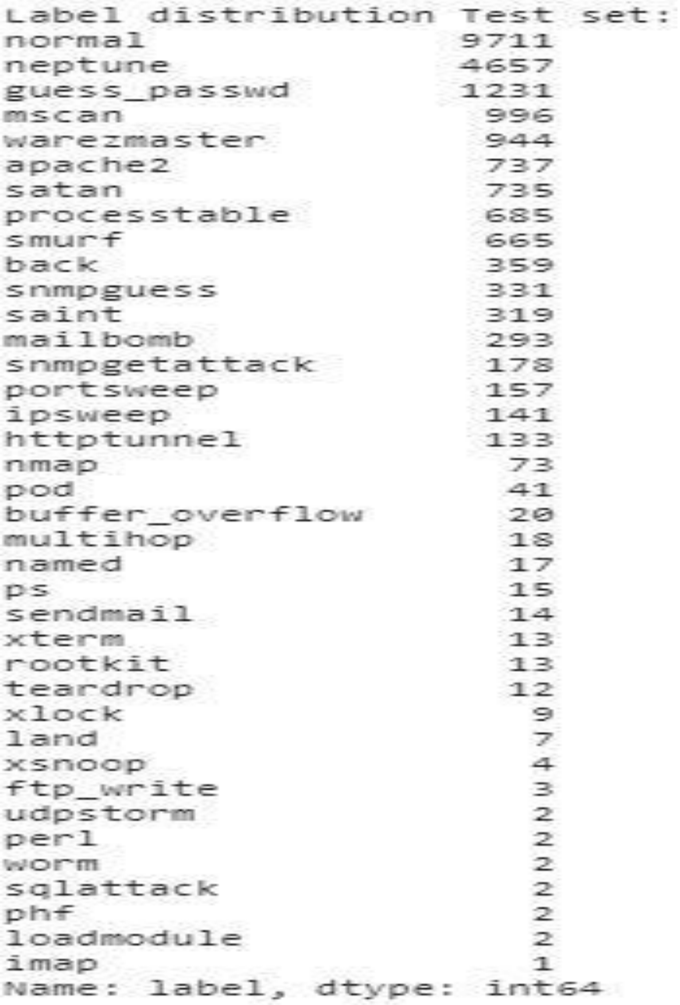
```
Train:
Dimensions of DoS: (113270, 123)
Test:
Dimensions of DoS: (17171, 123)
```

Figure 4.15: Dataset Dimensions



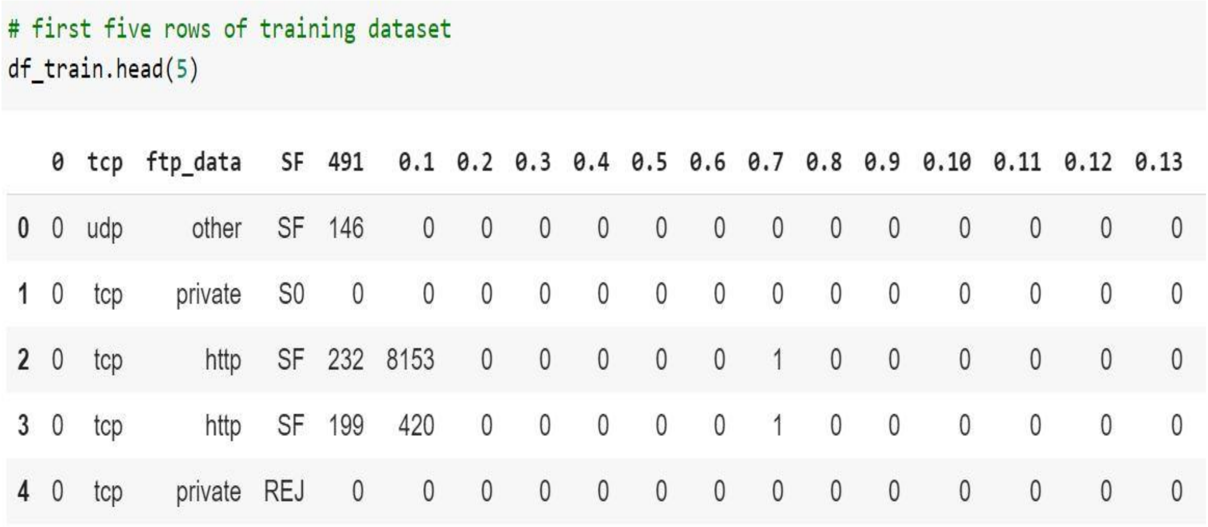
```
Label distribution Training set:
normal             67343
neptune            41214
satan               3633
ipsweep             3599
portsweep           2931
smurf               2646
nmap                1493
back                 956
teardrop             892
warezclient          890
pod                  201
guess_passwd          53
buffer_overflow       30
warezmaster           20
land                  18
imap                  11
rootkit               10
loadmodule             9
ftp_write              8
multihop               7
phf                    4
perl                   3
spy                    2
Name: label, dtype: int64
```

Figure 4.16: Labeled Distribution of Training Data



```
Label distribution Test set:
normal              9711
neptune             4657
guess_passwd        1231
mscan                996
warezmaster          944
apache2              737
satan                735
processtable         685
smurf                665
back                 359
snmpguess            331
saint                319
mailbomb             293
snmpgetattack        178
portsweep            157
ipsweep              141
httptunnel           133
nmap                  73
pod                   41
buffer_overflow       20
multihop              18
named                 17
ps                    15
sendmail              14
xterm                 13
rootkit               13
teardrop              12
xlock                  9
land                   7
xsnoop                 4
ftp_write              3
udpstorm               2
perl                   2
worm                   2
sqlattack              2
phf                    2
loadmodule             2
imap                   1
Name: label, dtype: int64
```

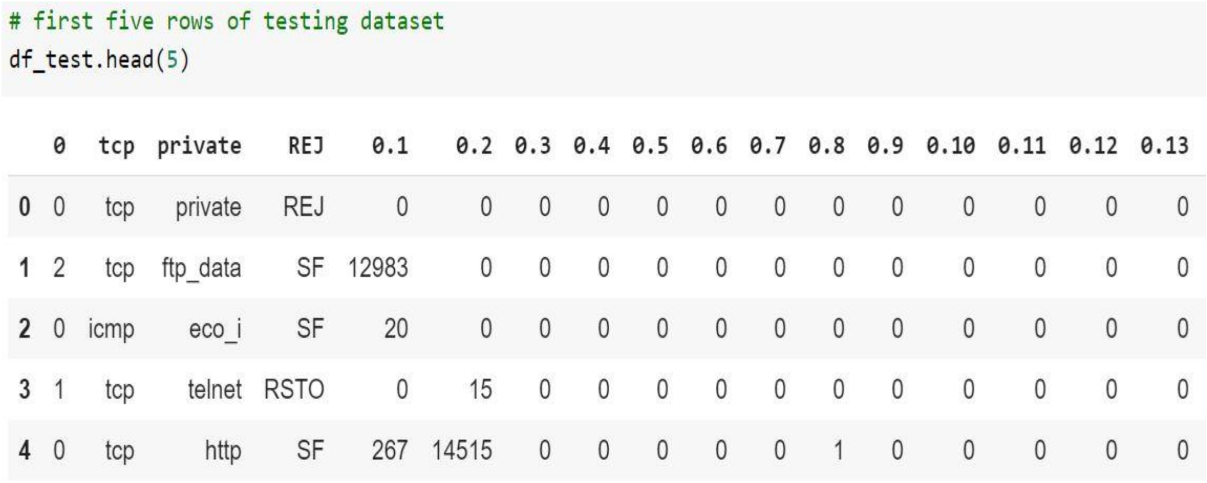
Figure 4.17: Labeled Distribution of Test Data



```
# first five rows of training dataset
df_train.head(5)
```

	0	tcp	ftp_data	SF	491	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.10	0.11	0.12	0.13
0	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0
3	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0
4	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.18: Training Dataset first 5 instances



```
# first five rows of testing dataset
df_test.head(5)
```

	0	tcp	private	REJ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.10	0.11	0.12	0.13
0	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0	0	0
2	0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0	0	0	0
3	1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0	0	0
4	0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0	0	0

Figure 4.19: Testing Dataset first 5 instances

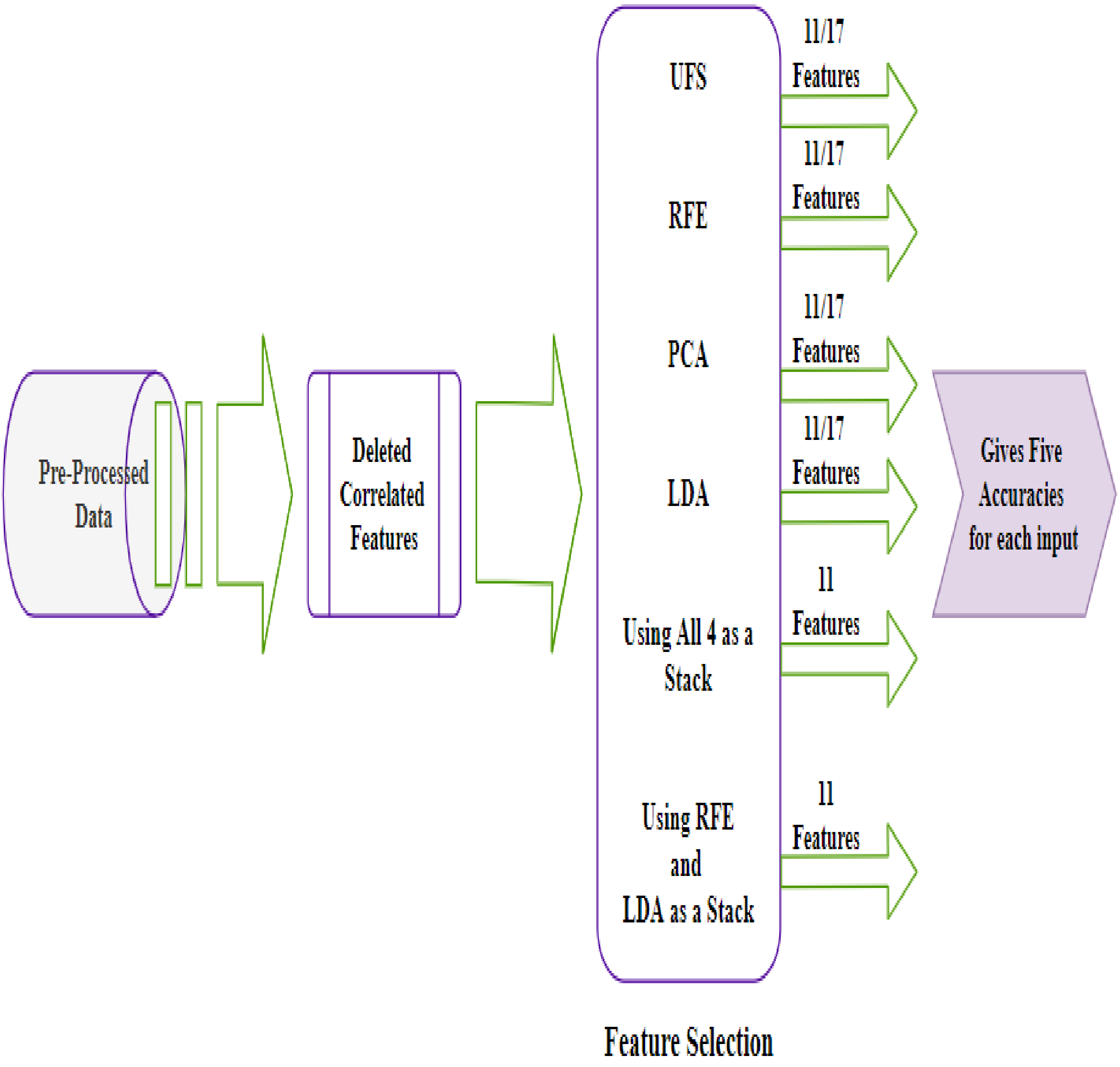


Figure 4.20: Model training and validation

Table 4.5: Accuracies using K-Fold Cross-Validation

Classifier Used	⟶	**SVM**	**Perceptron**	**KNN**	**SGD**	**XGBoost**
↓	**No. of Features**					
After Dropping Correlations	28	77.82%	25.44%	**99.82%**	22.32%	**99.96%**
Univariaite feature Selection	17	63.77%	25.41%	**99.81%**	26.79%	**99.93%**
RFE	17	86.25%	28.43%	**99.87%**	29.34%	**99.94%**
PCA	17	96.5%	23.33%	**99.82%**	12.74%	**99.94%**
LDA	17	99.18%	98.31%	**99.79%**	98.87%	**99.86%**
Univariate Feature Selection	11	91.35%	27.44%	**99.79%**	23.57%	**99.91%**
RFE	11	98.82%	87.54%	**99.11%**	88.84%	**99.43%**
PCA	11	95.93%	23.33%	**99.82%**	7.34%	**99.9%**
LDA	11	99.16%	98.58%	**99.79%**	98.48%	**99.84%**
algorithms as a stack	11	99.13%	98.62%	**99.86%**	98.63%	**99.87%**
RFE and LDA as a stack	**11**	**99.16%**	**98.59%**	**99.81%**	**98.65%**	**99.82%**

Table 4.6: Accuracy comparison with other techniques

Author Name	**Classifiers Used**	**Results**
You et al.[21]	RNN	92.7%
Alrawashdeh et al.[22]	RBM	97.9%
Marwane et al.[27]	C4.5	98.8%
Muhammad Aamir et. al [32]	KNN,SVM,RF	99.66%
Li et al. [42]	AutoEncoder+ *DBN*	92.10%
Gao et al. [43]	*DBN*	93.49%
Proposed Technique	**Stack Based Approach**	**99.87%**

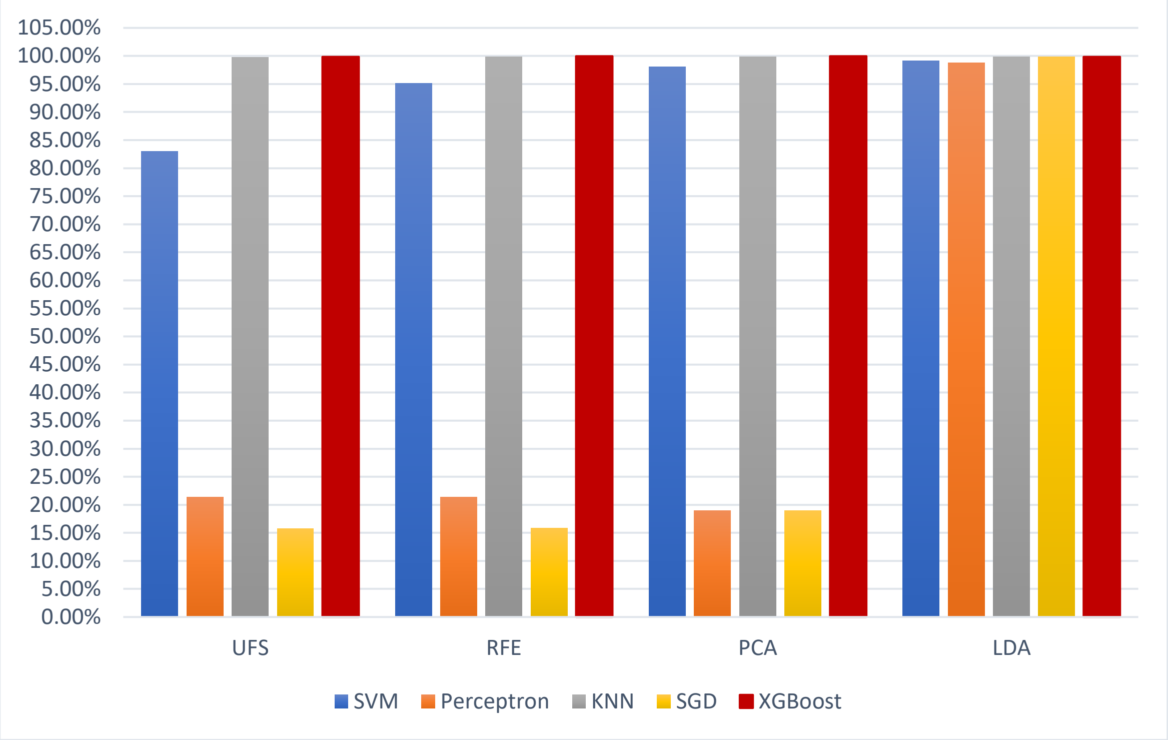


Figure 4.21: Accuracies with 17 features

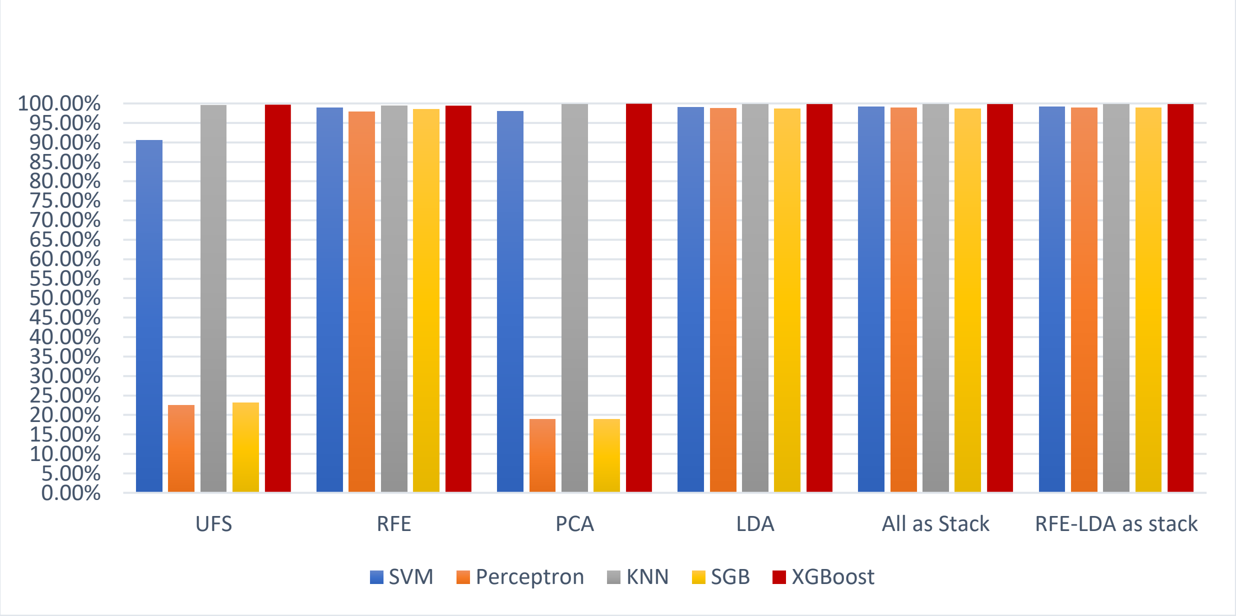
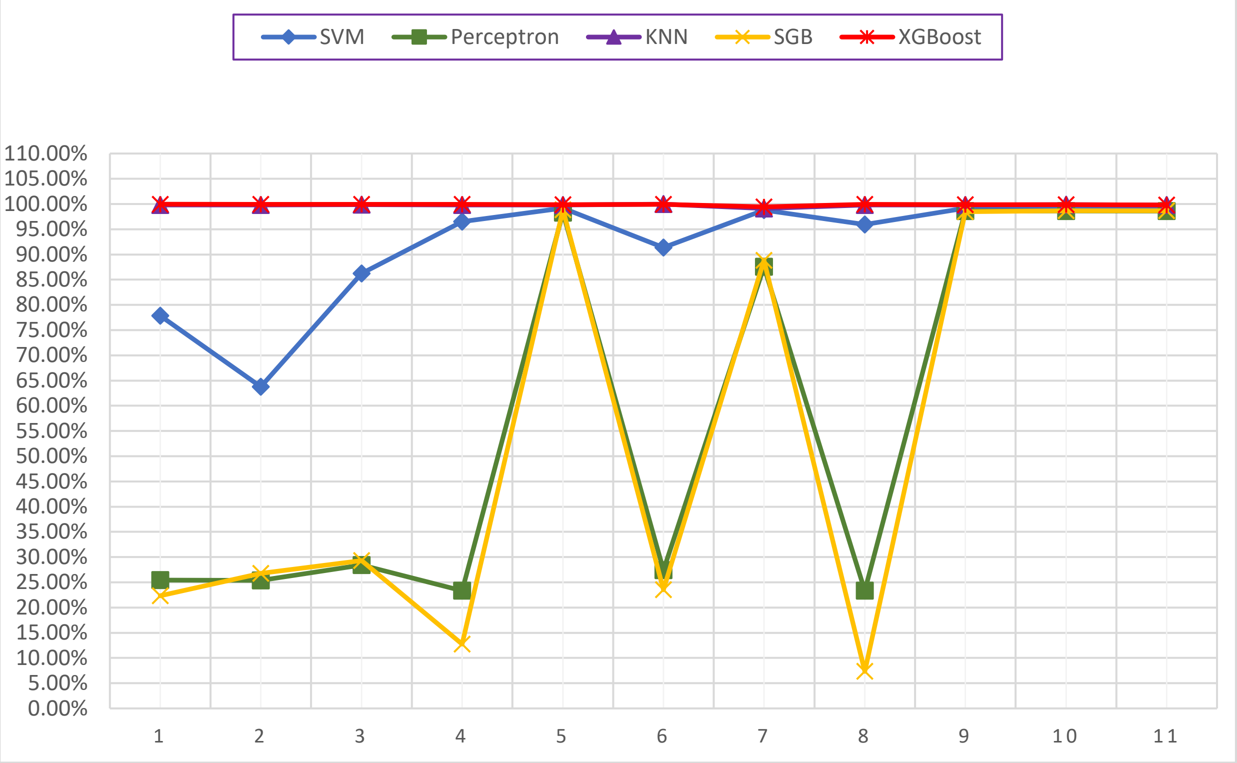


Figure 4.22: Accuracies with 11 features



1: Correlation Matrix (28), 2: UFS(17), 3: RFE(17), 4: PCA(17), 5:LDA(17), 6: UFS(11), 7: RFE(11), 8: PCA(11), 9: LDA(11), 10: All as stack(11),11: RFE-LDA as stack(11)

Figure 4.23: Accuracies of classifiers by using different feature selection techniques

4.4 ANN Based Framework Results

Both repositories classify all risks into four classes: DoS, Probe, U2R, and R2L. There are 22 hazard trends and 41 feature domains in the datasets. Easy features such as protocol type, packet size, and so on, knowledge base features such as the number of logins lost, and so on, and time management features such as the proportion of connections to SYN errors are the three types of features. Both 22 types of threats were present in both the training and research datasets. These assaults are selected at random during the pre-processing stage. Various measures, such as accuracy, recall, precision, F-Measure, and false alarm rate, are used to evaluate the efficacy of the suggested system. The probability of detection by recall metric and the likelihood of a false alarm is widely used to measure the recognizing efficiency on the norm and hazard classes. The performance balance of IDS between ordinary and hazard classes is detected using skewed F-scores. Both of these moves are assessed using four previews from the application on the classification framework's reference dataset[132]. True positive (tp) describes the number of assaults that are correctly identified, false positive (fp) describes the number of legitimate incidents that are identified as attacks, true negative (tn) describes the number of legitimate attacks that are correctly identified, and false negative (fn) describes the number of assaults that are identified as attacks. Tables 4.7, 4.8, and 4.9 show the comparable findings acquired in a tabular format.

Table 4.7: Results obtained using various Machine Learning Techniques

Algorithm	Accuracy with all features (**Random Forest**)	Accuracy with Feature Subset (80)
Random Forest + Genetic Algorithm	96.01 %	95.83%
Logistic Regression + Genetic Algorithm	**96.50**	**99.97**
Decision Tree + Genetic Algorithm	95.66	82.28
XGBoost+ Genetic Algorithm	95.74	82.28

Table 4.8: Result obtained using ANN on KDD Dataset

Learning Rate	Epochs	Batch Size	Accuracy	Recall	False Positive rate (10^{-5})	Precision	F-measure
0.001	**20**	**36**	**99.99**	**1.0**	**1.3497**	**94.28**	**97.05**
0.001	5	20	99.99	1.0	2.6994	88.57	93.93
0.01	20	10	99.99	1.0	2.6994	88.57	93.93
0.01	15	36	99.99	1.0	2.6994	88.57	93.93
0.01	20	20	99.99	96.87	2.6995	88.57	92.53
0.001	15	10	99.99	96.87	2.6995	88.57	92.53
0.001	10	36	99.99	96.87	2.6995	88.57	92.53
0.01	5	10	99.99	1.0	2.6994	88.57	93.93
0.1	10	15	99.99	1.0	2.6994	88.57	93.93
0.01	20	36	99.99	96.87	2.6995	88.57	92.53

Table 4.9: Result obtained using ANN on NSL-KDD Dataset

Learning Rate	Epochs	Batch Size	Accuracy	Recall	False Positive rate (10^{-5})	Precision	F-measure	Time (Sec)
0.01	10	36	91.88	95.25	0.1026	85.58	90.16	65.39
0.001	20	36	89.90	97.63	0.1425	78.67	87.13	129.83
0.01	20	20	89.74	87.48	0.0846	89.14	88.30	234.13
0.001	**10**	**36**	**91.85**	**97.03**	**0.1126**	**83.80**	**89.93**	**65.57**
0.01	10	20	91.15	89.43	0.0749	90.32	89.87	115.93
0.001	20	20	88.65	97.74	0.1594	75.63	85.27	228.88
0.01	20	36	89.91	89.92	0.1009	86.47	88.16	129.24
0.001	10	20	89.81	93.40	0.1241	82.37	87.54	116.23

4.5 Explainable AI-Based Results

This section primarily focuses on the dataset, the output of the modeled IDS, and finally, the explanation of the acquired findings from the viewpoint of the suggested framework in terms of local and global explainability. The objective of this suggested framework is to get the description from IDS that corresponds to the expected category.

Accuracy, precision, recall, and F1-score are the assessment criteria used to assess the suggested IDS platform's success. The proportion of correctly classified examples to the total test range is known as accuracy. Precision is known as the proportion of examples classified as an assault to the total number of examples classified as an assault. The proportion of examples that are marked as an assault to all examples of the type of assault is known as recall. The F1-score is calculated by taking into account both accuracy and recall. A deep neural network with a learning rate of 0.001, epochs of 20, and a batch size of 36 was used in the suggested system training model. This system has a 99.99 percent accuracy, 94.28 percent precision, 100 percent memory, and an F1-score of 97.05 percent.

Figure 4.24 shows a rundown of the SHAP values derived using the proposed IDS framework. The obtained table, as seen in fig. 4.25, can be used to illustrate the understanding of the acquired information. The provided table, as seen in fig. 4.26, can be used to illustrate the understanding of the acquired information. This graph aids in the identification of important functions, with src bytes, flag S0, count, and service private being the top four. As seen in Fig. 4.26, the overall rationale for the acquired outcomes can be described. The top 20 critical attributes listed for DoS hazard, as well as their accompanying feature values, are seen in this diagram. Depending on the Shapley values, the color reflects feature values ranging from low to high. The Shapley values are plotted on the X-axis, while the properties are plotted on the Y-axis in Fig. 4.26. If the red color level rises, so does the feature rating. On the other hand, as the density of the blue color improves, the feature value reduces. Overlap points are reverberated in the y-axis direction, showing the distribution as a consequence of the Shapley values. The characteristics are arranged in this manner.

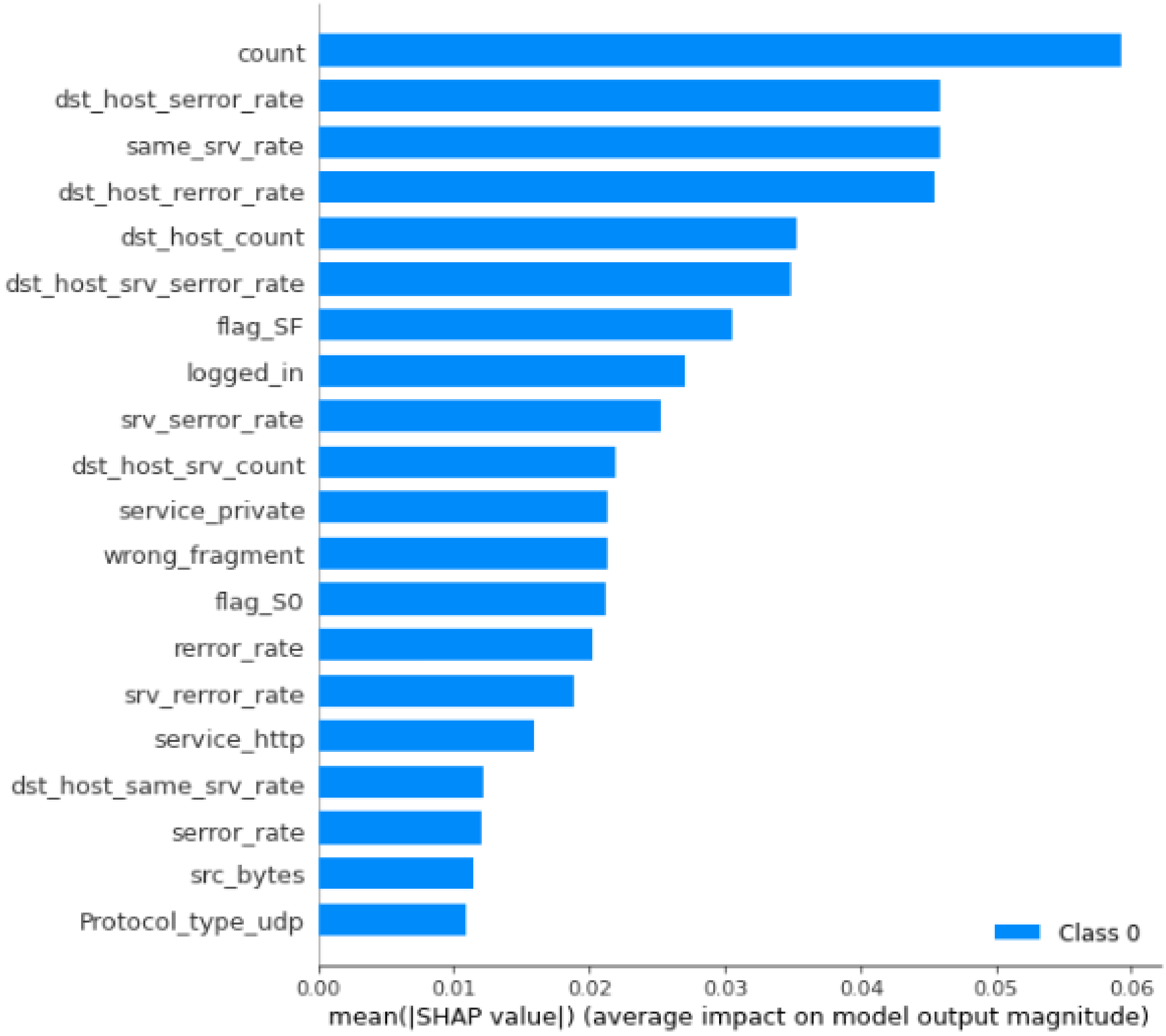


Figure 4.24**:** The Distribution of SHAP Values

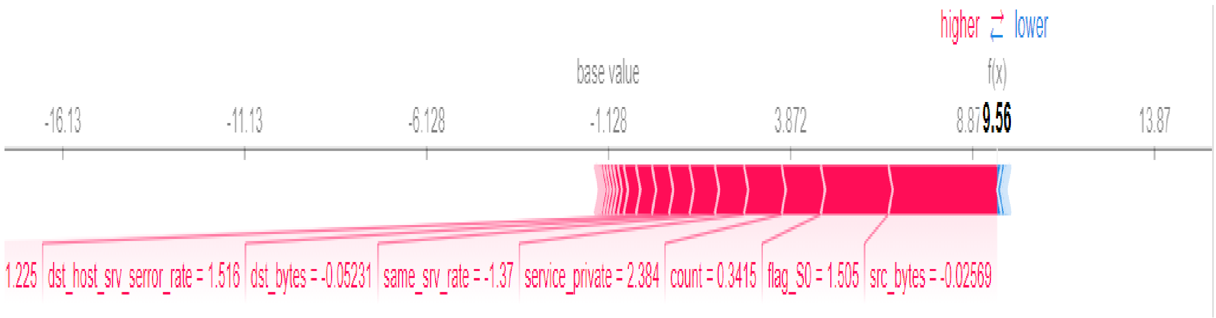


Figure 4.25: Interpretation of the deep neural network classifier

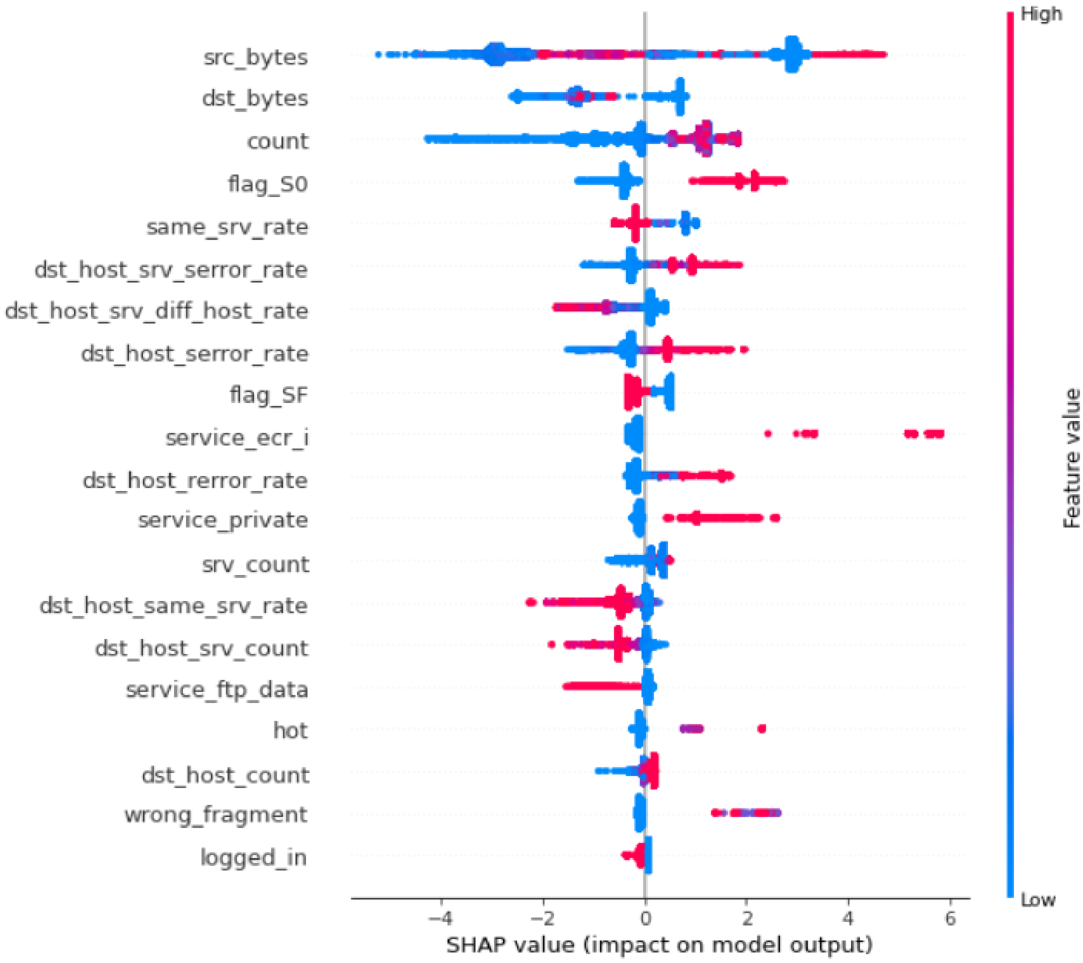


Figure 4.26**:** Top 20 attributes of DoS attack

4.6 Summary

The present conducted research work intended to present the recommendation by analyzing the considered dataset i.e, NSL-KDD which will be helpful for the generation of the IDS framework. Also, the research is intended to generate a novel framework for the identification and classification of various threats related to DDOS threats. As the ML algorithms along with feature selection algorithms able to generate the output but not the associated explanation. Thereby, to resolve that particular aspect explainable AI is considered along with SHAP values for the generation of explanation associated with the obtained outcome of the framework. The novel framework was generated through the above-mentioned methodologies for the identification and classification of threats along with the explanations associated with them.

Chapter 5

Conclusion and Future Works

Various frameworks are used to assess, identify, and classify the whole research study. One of the suggested implementations looked at the KDD and NSL-KDD datasets to see if the DDOS assault affected different areas like protocols, facilities, and flags. When the whole dataset is considered, the generalization would be more efficient. For the Bigdata case, it will be a successful and powerful problem. The simulation of successful IDS based on DDOS assaults can be represented due to these guidelines. This research can be extended to include other types of assaults, such as the Probe assault, R2L assault, and U2R assault, in addition to the effective guidelines. If appropriate guidelines on any of these types of assaults can be identified, the most appropriate intrusion detection can be developed.

One of the most difficult problems that WSN face is how to manage data while communicating across nodes, as well as the various assaults that can occur while doing so. The proposed model is trained several times with different numbers of attributes using various methodologies, and the outcomes are compared. The model is evaluated using the K-Fold cross-validation method to achieve more accurate data. Using XGBoost with the best 28 attributes selected after implementing the Correlations-Based attribute filtering process yields the highest precision of 99.96 percent. Furthermore, when applied as a stack, it achieves 99.87 percent accuracy with 11 attributes. XGBoost, on the other hand, gives more than 99.8% accuracy in nearly all 11 scenarios, according to K-fold cross-validation. The acquired outcomes also show that using RFE and LDA in combination with the correlation methodology improves precision. Both KNN and XGBoost have reasonable efficiencies, but XGBoost outperforms KNN in terms of overall efficiency. In the future, Nature Inspired methodologies could be utilized for optimization, and for preparation, Ensemble methodology or Deep Learning methodology could be utilized.

The amount of traffic information generated by IoT systems has increased significantly. In comparison, the traffic load on Internet networks has increased dramatically, promoting the flow of traffic across the systems. Millions of packets per second can be processed using the new backend interfaces. As a result, IDS must execute the packet analysis in a matter of nanoseconds. The existence of a system that can analyze large quantities of information at high speeds is a requirement in the field of intrusion detection. The architecture is compared using ANN on the datasets KDD'99 and NSL-KDD in this article. NSL-KDD is a dataset that solves the problem of multiple extraneous information, which is one of KDD'99's problems. In the same system, NSL-KDD had lower accuracy than KDD'99, but the recognition rate per category was higher. The recognition of attacks that occur more often in everyday life, such as DoS, was detrimental, while identifying deadly attacks was beneficial, thanks to the removal of redundant information and the use of selected information to enhance the detection of hazardous attacks. In the case of IDSs, it is essential to accomplish both objectives. As a result, work on these datasets' interpretation and technique growth will progress. Using parallel deep learning to reduce IDS time for very large datasets may be a fruitful avenue for potential research.

When analysts analyze and forecast different aspects depending on the dataset in question, there is no explanation for that specific prediction. When it comes to machine learning, computer vision, and natural language processing, the situation is close. It is critical to determine the cause for the specific forecast since this improves the model's interpretability. It is extremely important when dealing with information protection issues. In light of this, a model based on the NSL-KDD dataset was suggested to improve the predictability of the forecast. Various assessment parameters, such as accuracy, precision, recall, and F1-score, were used to assess the framework's results. The suggested framework has a 99.99 percent accuracy. Aside from estimation, the suggested framework improved the interpretability of the acquired forecasts by utilizing both local and global explainability, which would be beneficial to IDS specialists.

The research that is being conducted may be improved. First and foremost, additional data sets should be used to demonstrate the framework's suitability for network IDSs. Second, while SHAP has simple computations to directly convert machine learning models compared to the Shapley approximation, it is still impractical to use in real-time. Eventually, the SHAP mechanism can analyze more complex assaults such as

Advanced Persistent Threats (APT). This study adds to our understanding of IDS interpretability. Additional work in the future will focus on experimenting with additional databases, running the device in real-time, and explaining potential risks.

Computation time is one of the major concerns in a large datasets like KDD and NSL-KDD. To overcome this issue, nature inspired algorithm can be used for optimization, and training, either ensemble technique or deep learning algorithms can be used.

Chapter 6

Bibliography

[1] M. Abomhara, G.M. Køien, Security and privacy in the Internet of Things: Current status and open issues, in Proceedings of the IEEE International Conference on Privacy and Security in Mobile Systems, 2014, pp. 1-8.

[2] T.M. Behera, S.K. Mohapatra, U.C. Samal, M.S. Khan, M. Daneshmand, A.H. Gandomi, Residual EnergyBased Cluster-head Selection in WSNs for IoT Application, IEEE Internet of Things Journal 6(3) (2019)5132-5139.

[3] J. Rivera, R. van der Meulen, Gartner says the Internet of Things installed base will grow to 26 billion units by 2020, 2013, (https://www.gartner.com/newsroom/id/2636073).

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, Internet of Things: A survey on enabling technologies, protocols, and applications, IEEE Communication Surveys & Tutorials 17(4)(2015) 2347–2376.

[5] C.W. Tsai, C.F. Lai, M.C. Chiang, L.T. Yang, Data mining for internet of things: A survey, IEEE Communications Surveys & Tutorials 16 (1) (2013) 77-97.

[6] P.I.R. Grammatikis, P.G. Sarigiannidis, I.D. Moscholios, Securing the Internet of Things: challenges, threats, and solutions, Internet of Things 5 (2018) 41-70.

[7] M. Aly, F. Khomh, M. Haoues, A. Quintero, S. Yacout, Enforcing Security on Internet of ThingsFrameworks: A Systematic Literature Review, Internet of Things (2019) 100050.

[8] J. Landford, R. Meier, R. Barella, X. Zhao, E. Cotilla-Sanchez, R.B. Bass, S. Wallace, Fast sequence component analysis for attack detection in synchrophasor networks, in Proceedings of the 5thInternational Conference on Smart Cities and Green ICT Systems, 2016, pp. 1-8.

[9] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[10] https://anti-ddos.pro/en/ddos-attacks.html

[11] https://securelist.com/ddos-attacks-in-q2-2011/36394/

[12] https://www.technologyreview.com/the-download/610405/github-just-suffered-the-worlds-biggest-ddos-attack-it-barely-blinked/

[13] https://blog.apnic.net/2018/04/03/the-ddos-threat-landscape-in-2017/

[14] Chandola, V., Banerjee, A. and Kumar, V. (2009), "Anomaly detection: a survey", ACM Computer Survey, Vol. 41 No. 3, pp. 1-58.

[15] Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K. and Stanley, H.E. (2000), "PhysioBank, physiotoolkit, and PhysioNet: Components of a new research resource for complex physiologic signals", Circulation, Vol. 101 No. 23, pp. e215-e220, available at: http://circ.ahajournals.org/cgi /content/full/101/23/e215.

[16] Chuanhuang Li, Yan Wu, Xiaoyong Yuan, Zhengjun Sun, Weiming Wang, Xiaolin Li, Liang Gong, Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN, Int J Commun Syst. 2018, 31:e3497, https://doi.org/10.1002/dac.3497.

[17] Khamparia, A., Pande, S., Gupta, D., Khanna, A. and Sangaiah, A. (2020), "Multi-level framework for anomaly detection in social networking", Library Hi-Tech, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/LHT-01-2019-0023

[18] Partha Ghosh, et al. "An efficient hybrid multilevel intrusion detection system in a cloud environment." IOSR Journal of Computer Engineering 16.4 (2014): 16-26.

[19] Hariharan. M, Abhishek H. K and B. G. Prasad, DDoS Attack Detection Using C5.0 Machine Learning Algorithm, I.J. Wireless, and Microwave Technologies, 2019, 1, 52-59 Published Online January 2019 in MECS, DOI:10.5815/ijwmt.2019.01.06

[20] Bhuvaneswari Amma N.G., S. Selvakumar, Deep Radial Intelligence with Cumulative Incarnation Approach for Detecting Denial of Service Attacks, Neurocomputing (2019), DOI: https://doi.org/10.1016/j.neucom.2019.02.047

[21] Muhammad Aamir, Syed Mustafa Ali Zaidi, Clustering-based semi-supervised machine learning for DDoS attack classification, Journal of King Saud University –Computer and Information Sciences, Production and hosting by Elsevier, https://doi.org/10.1016/ j.jksuci. 2019.02.0031 319-1578/_2019.

[22] G. Dayanandam, T. V. Rao, D. Bujji Babu, and S. Nalini Durga, DDoS Attacks—Analysis and Prevention, © Springer Nature Singapore Pte Ltd. 2019 H. S. Saini et al. (eds.), Innovations in Computer Science and Engineering, Lecture Notes in Networks and Systems 32, https://doi.org/10.1007/978-981-10-8201-6_1

[23] K. Narasimha Mallikarjunan, A. Bhuvaneshwaran, K. Sundarakantham, and S. Mercy Shalinie, DDAM: Detecting DDoS Attacks Using Machine Learning Approach, Springer Nature Singapore Pte Ltd. 2019, N. K. Verma and A. K. Ghosh (eds.), Computational Intelligence: Theories, Applications, and Future Directions—Volume I, Advances in Intelligent Systems and Computing 798, https://doi.org/10.1007/978-981-13-1132-1_21.

[24] J. Cui, M. Wang, Y. Luo, et al., DDoS detection and defense mechanism based on cognitive-inspired computing in SDN, Future Generation Computer Systems (2019), https://doi.org/10.1016/j.future.2019.02.037.

[25] Omar E. Elejla1, Bahari Belaton1, Mohammed Anbar2, Basim Alabsi2, and Ahmed K. Al-Ani, Comparison of Classification Algorithms on ICMPv6 Based DDoS Attacks Detection, Springer Nature Singapore Pte Ltd. 2019, R. Alfred et al. (eds.), Computational Science and Technology, Lecture Notes in Electrical Engineering 481, https://doi.org/10.1007/978-981-13-2622-6_34.

[26] Mohamed Idhammad, Karim Afdel, Mustapha Belouch Semi-supervised machine learning approach for DDoS detection, Springer Science+Business Media, LLC, part of Springer Nature 2018, Applied Intelligence, https://doi.org/10.1007/s10489-018-1141-2.

[27] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi, A Deep Learning Approach to Network Intrusion Detection, IEEE Transactions On Emerging Topics In Computational Intelligence, Vol. 2, No. 1, February 2018.

[28] Olivier Brun, Yonghua Yin, Erol Gelenbe, Deep Learning with Dense Random Neural Network for Detecting Attacks against IoT-connected Home Environments, Procedia Computer Science 134 (2018) 458–463, Published by Elsevier Ltd.

[29] Rohan Doshi, Noah Apthorpe, Nick Feamster, Machine Learning DDoS Detection for Consumer Internet of Things Devices, 2018 IEEE Symposium on Security and Privacy Workshops, DOI 10.1109/SPW.2018.00013.

[30] Alekhya kaliki1, K Munivara Prasad, Machine Learning based Application Layer DDoS attack detection using Firefly Classification Algorithm, International Journal of Pure and Applied Mathematics, Volume 118 No. 17 2018, 635-645.

[31] Chuanhuang Li, Yan Wu, Xiaoyong Yuan, Zhengjun Sun, Weiming Wang, Xiaolin Li, Liang Gong, Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN, Int J Commun Syst. 2018, 31:e3497, https://doi.org/10.1002/dac.3497

[32] Muna AL-Hawawreh, Nour Moustafa, Elena Sitnikova, Identification of malicious activities in the industrial internet of things based on deep learning models, Journal of Information Security and Applications, © 2018 Elsevier Ltd. https://doi.org/10.1016/j.jisa.2018.05.002 2214-2126

[33] Xiaoyong Yuan, Chuanhuang Li, Xiaolin Li, DeepDefense: Identifying DDoS Attack via Deep Learning, 978-1-5090-6517-2/17/$31.00 ©2017 IEEE.

[34] Jin Kim, Nara Shin, Seung Yeon Jo and Sang Hyun Kim, Method of Intrusion Detection using Deep Neural Network, 978-1-5090-3015-6/17/$31.00 ©2017 IEEE.

[35] Indraneel, S., Praveen Kumar Vuppala, V., HTTP Flood attack Detection in Application Layer using Machine learning metrics and Bio-inspired Bat algorithm, Applied Computing, and Informatics (2017), DOI: https://doi.org/10.1016/j.aci.2017.10.003.

[36] Marwane Zekri, Said El Kafhali, Noureddine Aboutabit, and Youssef Saadi, DDoS Attack Detection using Machine Learning Techniques in Cloud Computing Environments, 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), IEEE, doi:10.1109/CloudTech.2017.8284731.

[37] Hossein Hadian Jazi, Hugo Gonzalez, Natalia Stakhanova, Ali A. Ghorbani, Detecting HTTP-based Application Layer DoS attacks on Web Servers in the presence of sampling, Computer Networks (2017), DOI: 10.1016/j.comnet.2017.03.018

[38] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for the Internet of Things, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.08.043

[39] S.A.R. Shah, B. Issac, Performance comparison of intrusion detection systems and application of machine learning to Snort system, Future Generation Computer Systems (2017), https://doi.org/10.1016/j.future.2017.10.016

[40] E. Hodo, X. J. A. Bellekens, A. Hamilton, C. Tachtatzis, and R. C. Atkinson, Shallow and deep networks intrusion detection system: A taxonomy and survey, Submitted to ACM Survey, 2017, [Online]. Available: http://arxiv.org/abs/1701.02145

[41] Q. Niyaz, W. Sun, and A. Y. Javaid, A deep learning-based DDOS detection system in software-defined networking (SDN), Submitted to EAI Endorsed Transactions on Security and Safety, In Press, 2017, [Online]. Available: http://arxiv.org/abs/1611.07400

[42] H.-W. Lee, N.-R. Kim, and J.-H. Lee, "Deep neural network self-training based on unsupervised learning and dropout," Int. J. Fuzzy Logic Intell. Syst., vol. 17, no. 1, pp. 1–9, Mar. 2017. [Online]. Available:http://www.ijfis.org/journal/view.html?doi=10.5391/ IJFIS.2017.17.1.1

[43] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep learning approach for network intrusion detection in software-defined networking," in Proc. Int. Conf. Wireless Netw. Mobile Commun., Oct. 2016, pp. 258–263.

[44] Manuel S. Hoyos Ll, Gustavo A. Isaza E, Jairo I. Vélez and Luis Castillo O, Distributed Denial of Service (DDoS) Attacks Detection Using Machine Learning Prototype, S. Omatu et al. (eds.), DCAI, 13th International Conference, Advances in Intelligent Systems and Computing 474, © Springer International Publishing Switzerland 2016, DOI: 10.1007/978-3-319-40162-1_4

[45] Elike Hodo, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis, and Robert Atkinson, Threat analysis of IoT networks Using Artificial Neural Network Intrusion Detection System, 978-1-5090-0284-9/16/$31.00 ©2016 IEEE.

[46] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," PLoS One, vol. 11, no. 6, Jun. 2016, Art. no. e0155781.

[47] Y.Wang, W.-D. Cai, and P.-C.Wei, "A deep learning approach for detecting malicious JavaScript code," Security Commun. Netw., vol. 9, no. 11, pp. 1520–1534, Jul. 2016.

[48] C. Garcia Cordero, S. Hauke, M. Muhlhauser, and M. Fischer, "Analyzing flow-based anomaly intrusion detection using replicator neural networks," in Proc. 14th Annu. Conf. Privacy, Security. Trust, Auckland, New Zealand, Dec. 2016, pp. 317–324.

[49] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proc. 9th EAI Int. Conf. Bio-Inspired Inf. Commun. Technol., 2016, pp. 21–26. [Online]. Available: http://dx.doi.org/10.4108/eai.3-12-2015.2262516

[50] S. Potluri and C. Diedrich, "Accelerated deep neural networks for an enhanced intrusion detection system," in Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom., Berlin, Germany, Sep. 2016, pp. 1–8.

[51] L. You, Y. Li, Y. Wang, J. Zhang, and Y. Yang, "A deep learning-based RNNs model for an automatic security audit of short messages," in Proc. IEEE 16th Int. Symp. Commun. Inf. Technol., Qingdao, China, Sep. 2016, pp. 225–229.

[52] K. Alrawashdeh and C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016, pp. 195–200.

[53] B. Dong and X. Wang, “Comparison deep learning method to traditional methods using for network intrusion detection,” in Proc. 8th IEEE Int. Conf. Commun. Softw. Netw., Beijing, China, Jun. 2016, pp. 581–585.

[54] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, “Deep learning and its applications to machine health monitoring: A survey,” Submitted to IEEE Trans. Neural Netw. Learn. Syst., 2016. [Online].Available: http://arxiv.org/abs/1612.07640

[55] Alkasassbeh, M., Al-Naymat, G., Hassanat, A.B., Almseidin, M.: Detecting distributed denial of service attacks using data mining techniques. Int. J. Adv. Comput. Sci. Appl., 7(1) (2016).

[56] Tesfahun, A., Bhaskari, D.L.: Intrusion detection using random forests classifier with SMOTE and feature reduction. In: 2013 International Conference on IEEE Cloud & Ubiquitous Computing & Emerging Technologies (CUBE) pp. 127–132 (2013).

[57] Sang Min Lee, Dong Seong Kim, Je Hak Lee, Jong Sou Park, Detection of DDoS attacks using optimized traffic matrix, Computers and Mathematics with Applications, Elsevier Ltd, doi:10.1016/j.camwa.2011.08.020.

[58] F. Falcini, G. Lami, and A. M. Costanza, “Deep learning in automotive software,” IEEE Softw.,vol.34,no.3,pp.56-63,May2017.[Online].Available:http://ieeexplore.ieee.org/ document/7927925/

[59] A. Luckow, M. Cook, N. Ashcraft, E.Weill, E. Djerekarov, and B. Vorster, “Deep learning in the automotive industry: Applications and tools,” in Proc. IEEE Int. Conf. Big Data, Dec. 2016, pp. 3759–3768. [Online]. Available: http://ieeexplore.ieee.org/document /7841045/

[60] Z. Liang, G. Zhang, J. X. Huang, and Q. V. Hu, “Deep learning for healthcare decision making with EMRs,” in Proc. IEEE Int. Conf. Diro informat. Biomed., Nov. 2014, pp. 556–559.

[61] S. P. Shashikumar, A. J. Shah, Q. Li, G. D. Clifford, and S. Nemati, “A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology,” in Proc. IEEE EMBS Int. Conf. Biomed. Health Information, FL, USA, 2017, pp. 141–144.

[62] H. Lee, Y. Kim, and C. O. Kim, “A deep learning model for robust wafer fault monitoring with sensor measurement noise,” IEEE Trans. Semicond. Manuf., vol. 30, no. 1, pp. 23–31, Feb. 2017.

[63] R. Polishetty, M. Roopaei, and P. Rad, “A next-generation secure cloud-based deep learning license plate recognition for smart cities,” in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016, pp. 286–293.

[64] Md Al Mehedi Hasan, et al. "Feature selection for intrusion detection using random forest." Journal of information security 7.03 (2016): 129.

[65] Praveen Bhanodia, Babita Pandey, Devendra Kr Pandey, Aditya Khamparia, “A Comprehensive Survey of Link Prediction in Social Networks: Techniques, Parameters and Challenges”, Expert System with Applications, Vol. 124, pp. 164-18, 2019.

[66] Aditya Khamparia, Karan M Singh.: “A Systematic Survey on Deep Learning Architectures and Applications”, Expert System, https://doi.org/10.1111/exsy.12400, 2019.

[67] Utkarsh Agrawal, Jatin Sharma, Rahul Singh, Deepak Gupta, Ashish Khanna, Aditya Khamparia.: “Hybrid Wolf-Bat algorithm for optimization of connection weights in multi-layer perceptron”. In: *ACM Transactions on Multimedia Computing Communications and Applications* (DOI: https://doi.org/10.1145/3350532.

[68] Ashish Kumar Luhach, Aditya Khamparia, Ravindra Sihag, Raj Kumar, “Honey Bee Optimization-based Sink Mobility Aware Heterogeneous Protocol for Wireless Sensor Network”, Scalable Computing: Practice and Experience, Vol. 20, Issue 4, Page 591-598, 2020.

[69] Muhammad Aamir, Syed Mustafa Ali Zaidi, Clustering-based semi-supervised machine learning for DDoS attack classification, Journal of King Saud University –Computer and Information Sciences, Production and hosting by

Elsevier, February 2019, https://doi.org/10.1016/j.jksuci.2019.02.0031 319-1578/_2019

[70] Z. Chen, F. Jiang, Y. Cheng, X. Gu, W. Liu and J. Peng, "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-Based Cloud," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Shanghai, pp. 251-256, 2018.

[71] Hariharan. M, Abhishek H. K and B. G. Prasad, DDoS Attack Detection Using C5.0 Machine Learning Algorithm, I.J. Wireless, and Microwave Technologies, 2019, 1, 52-59 January 2019 in MECS, DOI:10.5815/ijwmt.2019.01.06

[72] HerveNkiama, Syed ZainudeenMohd Said, and Muhammad Saidu. "A subset feature elimination mechanism for the intrusion detection system." International Journal of Advanced Computer Science and Applications 7.4 (2016): 148-157.

[73] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[74] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016, pp. 1135–1144.

[75] L. S. Shapley, “A value for n-person games,” Contributions to the Theory of Games, vol. 2, no. 28, pp. 307–317, 1953.

[76] R. Mitchell, I.R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACMComputing Surveys 46(4) (2014) 55.

[77] Holzinger, Andreas. "From machine learning to explainable AI." In 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), pp. 55-66. IEEE, 2018.

[78] Ignatiev, Alexey. "Towards trustable explainable AI." In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI, pp. 5154-5158. 2020.

[79] Wang, Maonan, Kangfeng Zheng, Yanqing Yang, and Xiujuan Wang. "An Explainable Machine Learning Framework for Intrusion Detection Systems." IEEE Access 8 (2020): 73127-73141.

[80] Margaliot, M. and Marcelloni, F., “Intelligence : Why, When,” IEEE Comput. Intell. Mag., vol. 14, no. February, pp. 69–81, 2019.

[81] Wang, Danding, Qian Yang, Ashraf Abdul, and Brian Y. Lim. "Designing theory-driven user-centric explainable AI." In Proceedings of the 2019 CHI conference on human factors in computing systems, pp. 1-15. 2019.

[82] Lundberg, Scott M., Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. "From local explanations to a global understanding with explainable AI for trees." Nature machine intelligence 2, no. 1 (2020): 2522-5839.

[83] Ding, Yalei, and Yuqing Zhai. "Intrusion detection system for NSL-KDD dataset using convolutional neural networks." In Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 81-85. 2018.

[84] Gurung, Sandeep, Mirnal Kanti Ghose, and Aroj Subedi. "Deep learning approach on network intrusion detection system using NSL-KDD dataset." International Journal of Computer Network and Information Security (IJCNIS) 11, no. 3 (2019): 8-14.

[85] Shone, Nathan, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. "A deep learning approach to network intrusion detection." IEEE transactions on emerging topics in computational intelligence 2, no. 1 (2018): 41-50.

[86] Protić, Danijela D. "Review of KDD Cup'99, NSL-KDD and Kyoto 2006+ datasets." Vojnotehnički glasnik 66, no. 3 (2018): 580-596.

[87] Amarasinghe, Kasun, Kevin Kenney, and Milos Manic. "Toward explainable deep neural network-based anomaly detection." In 2018 11th International Conference on Human System Interaction (HSI), pp. 311-317. IEEE, 2018.

[88] A. Barredo et al., “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” Inf. Fusion, vol. 58, no. December 2019, pp. 82–115, 2020.

[89] Pande, Sagar, Aditya Khamparia, Deepak Gupta, and Dang NH Thanh. "DDOS Detection Using Machine Learning Technique." In Recent Studies on Computational Intelligence, pp. 59-68. Springer, Singapore, 2020.

[90] Hajimirzaei, Bahram, and Nima Jafari Navimipour. "Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm." ICT Express 5, no. 1 (2019): 56-59.

[91] Kwon, Donghwoon, Kathiravan Natarajan, Sang C. Suh, Hyunjoo Kim, and Jinoh Kim. "An empirical study on network anomaly detection using convolutional neural networks." In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), pp. 1595-1598. IEEE, 2018.

[92] Thomas, Rajesh, and Deepa Pavithran. "A Survey of Intrusion Detection Models based on NSL-KDD Data Set." In 2018 Fifth HCT Information Technology Trends (ITT), pp. 286-291. IEEE, 2018.

[93] Pande, Sagar Dhanraj, and Aditya Khamparia. "A Review on Detection of DDOS Attack Using Machine Learning and Deep Learning Techniques.".

[94] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems, 2017, pp. 4765–4774.

[95] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD-cup 99 data set,” in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE, 2009, pp. 1–6.

[96] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, 2016, pp. 1135–1144.

[97] L. S. Shapley, “A value for n-person games,” Contributions to the Theory of Games, vol. 2, no. 28, pp. 307–317, 1953.

[98] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD cup 99 data set,” in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications. IEEE, 2009, pp. 1–6.

[99] S. Revathi and A. Malathi, “A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection,” International

Journal of Engineering Research & Technology (IJERT), vol. 2, no. 12, pp. 1848–1853, 2013.

[100] L. Dhanabal and S. Shantharajah, “A study on NSL_KDD dataset for intrusion detection system based on classification algorithms,” International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, no. 6, pp. 446–452, 2015.

[101] M. Abomhara, G.M. Køien, Security and privacy in the Internet of Things: Current status and open issues, in Proceedings of the IEEE International Conference on Privacy and Security in Mobile Systems, 2014, pp. 1-8.

[102] T.M. Behera, S.K. Mohapatra, U.C. Samal, M.S. Khan, M. Daneshmand, A.H. Gandomi, Residual EnergyBased Cluster-head Selection in WSNs for IoT Application, IEEE Internet of Things Journal 6(3) (2019)5132-5139.

[103] J. Rivera, R. van der Meulen, Gartner says the Internet of Things installed base will grow to 26 billion units by 2020, 2013, (https://www.gartner.com/newsroom/id/2636073).

[104] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, Internet of Things: A survey on enabling technologies, protocols, and applications, IEEE Communication Surveys & Tutorials 17(4)(2015) 2347–2376.

[105] C.W. Tsai, C.F. Lai, M.C. Chiang, L.T. Yang, Data mining for internet of things: A survey, IEEE Communications Surveys & Tutorials 16 (1) (2013) 77-97.

[106] P.I.R. Grammatikis, P.G. Sarigiannidis, I.D. Moscholios, Securing the Internet of Things: challenges, threats, and solutions, Internet of Things 5 (2018) 41-70.

[107] M. Aly, F. Khomh, M. Haoues, A. Quintero, S. Yacout, Enforcing Security on Internet of ThingsFrameworks: A Systematic Literature Review, Internet of Things (2019) 101

[108] J. Landford, R. Meier, R. Barella, X. Zhao, E. Cotilla-Sanchez, R.B. Bass, S. Wallace, Fast sequence component analysis for attack detection in

synchrophasor networks, in Proceedings of the 5thInternational Conference on Smart Cities and Green ICT Systems, 2016, pp. 1-8.

[109] Pande, Sagar, Aditya Khamparia, Deepak Gupta, and Dang NH Thanh. "DDOS Detection Using Machine Learning Technique." In Recent Studies on Computational Intelligence, pp. 59-68. Springer, Singapore, 2021.

[110] Pande, Sagar Dhanraj, and Aditya Khamparia. "A Review on Detection of DDOS Attack Using Machine Learning and Deep Learning Techniques." Think India journal 22, no. 16 (2019): 2035-2043.

[111] Pande, Sagar, and Ajay B. Gadicha. "Prevention mechanism on DDOS attacks by using multilevel filtering of distributed firewalls." International Journal on Recent and Innovation Trends in Computing and Communication 3, no. 3 (2015): 1005-1008.

[112] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.

[113] E. Ramentol, Y. Caballero, R. Bello, F. Herrera, SMOTE-RSB*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory, Knowledge and information systems 33(2) (2012) 245-265.

[114] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P. J. Kennedy, Training deep neural networks on imbalanced data sets, IEEE international joint conference on neural networks, 2016, 4368–4374.

[115] M. Mohammadi, A. Al-Fuqaha, S. Sorour, M. Guizani, Deep learning for IoT big data and streaming analytics: A survey, IEEE Communications Surveys & Tutorials 20 (4) (2018) 2923-2960.

[116] F.J. Pulgar, F. Charte, A.J. Rivera, M.J. del Jesus, Choosing the proper autoencoder for feature fusion based on data complexity and classifiers: Analysis, tips, and guidelines, Information Fusion 54 (2020) 44-60.

[117] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. Manzagol, Stacked denoising autoencoders: learning useful representations in a deep network with

a local denoising criterion, Journal of machine learning research 11 (2010) 3371–3408.

[118] Y. Fan, C. Zhang, Z. Liu, Z. Qiu, Y. He, Cost-sensitive stacked sparse auto-encoder models to detect striped stem borer infestation on rice-based on hyperspectral imaging, Knowledge-Based Systems 168(2019) 49-58.

[119] H.H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K.R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks, IEEE transactions on Emerging Topics in Computing 7(2) (2016) 314-323.

[120] R. Kozik, M. Choraś, M. Ficco, F. Palmieri, A scalable distributed machine learning approach for attack detection in edge computing environments, Journal of Parallel and Distributed Computing 119 (2018)18-26.

[121] Q. Zhang, L. T. Yang, Z. Chen, P. Li, A survey on deep learning for big data, Information Fusion 42 (2018)146-157.

[122] S.H. Khan, M. Hayat, M. Bennamoun, F.A. Sohel, R. Togneri, Cost-sensitive learning of deep feature representations from imbalanced data, IEEE transactions on neural networks and learning systems 29(8)(2017) 3573-3587.

[123] Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem, Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches, Internet of Things, Volume 7,2019, 100059, ISSN 2542-6605.https://doi.org/10.1016/j.iot.2019. 100059.

[124] O. Brun, Y. Yin, E. Gelenbe, Y.M. Kadioglu, J. Augusto-Gonzalez, M. Ramos, Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments, Procediacomputer science 134 (2018) 458-463.

[125] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for the internet of things, Future Generation Computer Systems 82 (2018) 761–768.

[126] Q. Feng, Y. Zhang, C. Li, Z. Dou, J. Wang, Anomaly detection of spectrum in wireless communication via deep auto-encoders, The Journal of Supercomputing 73(7) (2017) 3161–3178.

[127] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT, Sensors 17(9) (2017).

[128] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, Feb. 2018, DOI: 10.1109/TETCI.2017.2772792.

[129] B.B. Zarpelao, R.S. Miani, C.T. Kawakami, S.C. de Alvarenga, A survey of intrusion detection in the Internet of Things, Journal of Network and Computer Applications 84 (2017) 25-37.

[130] W.W. Ng, G. Zeng, J. Zhang, D.S. Yeung, W. Pedrycz, Dual autoencoders features for the imbalanced classification problem, Pattern Recognition 60 (2016) 875-889.

[131] M. Nijim, H. Albataineh, M. Khan, D. Rao, FastDetict: A data mining engine for predicting and preventing DDoS attacks, in Proceedings of the IEEE International Symposium on Technologies for homeland security, 2017, pp. 1-5.

[132] Y.-A. Chung, H.-T. Lin, S.-W. Yang, Cost-aware pretraining for multiclass cost-sensitive deep learning,2015, [Online]. Available: https://arxiv.org/abs/1511.09337.

[133] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: a comprehensive review, Journal of Network and Computer Applications 36(1) (2013) 16-24.

[134] R. Mitchell, I.R. Chen, A survey of intrusion detection techniques for cyber-physical systems, ACMComputing Surveys 46(4) (2014) 55.

[135] Tiwari, P. and Melucci, M. (2018a), “Towards a quantum-inspired framework for binary classification”, The 27th ACM International Conference on Information and Knowledge Management, October 22–26, Torino and New York, NY, pp. 4, available at https://doi.org/10.1145/3269206.3269304

[136] Su, Tongtong, Huazhi Sun, Jinqi Zhu, Sheng Wang, and Yabo Li. "BAT: deep learning methods on network intrusion detection using NSL-KDD dataset." IEEE Access 8 (2020): 29575-29585.

[138] Pradeep Mohan Kumar, K., M. Saravanan, M. Thenmozhi, and K. Vijayakumar. "Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks." Concurrency and Computation: Practice and Experience 33, no. 3 (2021): e5242.

[139] Belgrana, Fatima Zohra, Nacéra Benamrane, Mohamed Amine Hamaida, Abdellah Mohamed Chaabani, and Abdelmalik Taleb-Ahmed. "Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features." In 2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pp. 23-29. IEEE, 2021.

[140] Singh, Kuldeep, Lakhwinder Kaur, and Raman Maini. "Comparison of Principle Component Analysis and Stacked Autoencoder on NSL-KDD Dataset." In Computational Methods and Data Engineering, pp. 223-241. Springer, Singapore, 2021.

List of Research Publications

1. Sagar Pande, Aditya Khamparia, Deepak Gupta,(2021)**Feature Selection And Comparison Of Classification Algorithms For Network Intrusion Detection System**, Journal of Ambient Intelligence and Humanized Computing, Journal, SCI, Impact Factor=7.104 **(Published)**
2. Sagar Pande, Aditya Khamparia, Deepak Gupta, (2020). **Multi-level framework for anomaly detection in social networking**, Library Hi-Tech, 2020. SCI, Impact Factor: 1.256 **(Published)**
3. Sagar Pande, Aditya Khamparia, Deepak Gupta,(2021) **An intrusion detection system on healthcare system using machine and deep learning**, World Journal of Engineering, Scopus & ESCI, Impact Factor: 1.2 **(Published)**
4. Sagar Pande, Aditya Khamparia, Deepak Gupta, (2021) **DDOS Detection Using Machine Learning Technique**, Recent Studies on Computational Intelligence. Studies in Computational Intelligence, vol 921. Springer, Scopus. **(Published)**
5. Sagar Pande, Aditya Khamparia, (2019) **A Review on Detection of DDOS Attack Using Machine Learning and Deep Learning Techniques**, Think India Journal, UGC-Care. **(Published)**
6. Sagar Pande, Aditya Khamparia, Deepak Gupta,(2021) **Recommendations for DDOS Attack based Intrusion Detection System through data analysis**, ICICC-2021, Conference, Scopus. **(Presented)**
7. Sagar Pande, Aditya Khamparia, Deepak Gupta,(2021) **Recommendations for DDOS Attack using Tableau,** International Conference On Data Analytics & Management Conference (ICDAM-2021), Scopus. **(Presented)**
8. Sagar Pande, Aditya Khamparia, and Deepak Gupta,(2021) **Explainable Deep Neural Network Based Analysis on Intrusion Detection Systems,** Special Issue on: Explainable AI (XAI) for Web-based Information Processing**,** Journal, SCI, Impact Factor=4.787. **(Communicated)**