An optimised machine learning approach for detecting a network attack in an Intrusion detection system

ADITYA YADAV

Under the Supervision of

Dr. PANKAJ KUMAR KESERWANI

PROJECT REPORT

Submitted to

NATIONAL INSTITUTE OF TECHNOLOGY SIKKIM

for the award of the Degree of

Bachelor of Technology

in

Computer Science and Engineering

June 2021

Certificate by Supervisors

This is to certify that the project report entitled **"An optimised machine learning approach for detecting a network attack in an Intrusion detection system"** is being submitted by Mr. Aditya Yadav (Roll No. B170071CS), a student in the Department of Computer Science and Engineering, National Institute of Technology Sikkim, for the award of the degree of Bachelor of Technology (B.Tech). It is an original work carried out by him under my supervision and guidance. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Dr. Pankaj Kumar Keserwani

Certificate by Student

I hereby declare that the work presented in the report entitled **"An optimised machine learning approach for detecting a network attack in an Intrusion detection system"** is a bonafide record of the work done by me under the supervision of Dr. Pankaj Kumar Keserwani, Department of Computer Science and Engineering, and that no part thereof has been presented for the award of any other degree.

- I have followed the guidelines provided by the institute in writing the report.
- The report does not contain any classified information.
- Whenever I have used materials from other sources, I have given due credits to those by citing them in the text of the report and giving the details in the references.
- Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credits to the sources by citing them in the text of the report and giving their details in the references.

Dated: 29-06-2021 Mr. Aditya Yadav
Place: NIT SIKKIM Roll No.: B170071CS

Acknowledgment

I wholeheartedly thank our Project Guide Dr. Pankaj Kumar Keserwani, Department of Computer Science and Engineering, National Institute of Technology Sikkim for his extreme support and guidance throughout the project work.

My special thanks to our head of department Dr. Pratyay Kuila, Department of Computer Science and Engineering, National Institute of Technology Sikkim, who allowed me to carry out this project work and provided his guidance whenever necessary.

I am grateful to my parents who gave me persistent encouragement and inspiration throughout my being.

Finally, I would like to thank everyone who has directly or indirectly helped me in the successful completion of this project and throughout.

Aditya Yadav

Contents

List of Tables

List of Equations

List of figures

Chapter 1: Introduction

1.1 Abstract

As a consequence of the growing use of emerging technologies such as IoT, cloud computing, artificial intelligence, edge computing, and quantum computing, network security is degrading as a result of the continual growth in the number of cyber-attacks. However, at the same time, because of the growing reliance of individuals and businesses on the internet, as well as their concerns about the security and privacy of their online activities, cyber-security has gotten a lot of attention [1]. To guard against harmful online activity, many prior machine learning (ML)-based network intrusion detection systems (NIDSs) have been created.

An intrusion detection system (IDS), a critical cyber security technology, keeps track of the condition of the network's software and hardware. Existing IDSs still confront problems in increasing detection accuracy, lowering false alarm rates, and identifying novel threats, despite decades of research [2]. Many researches have concentrated on creating IDSs that employ machine learning approaches to tackle the challenges mentioned above. With great accuracy, machine learning algorithms can automatically detect the key distinctions between normal and anomalous data. Machine learning algorithms are also capable of detecting unknown attacks due to their high generalizability [3].

Despite the application of numerous supervised and unsupervised learning methods in the field of machine learning to improve the efficacy of IDSs, present intrusion detection algorithms still struggle to reach high performance [4]. First, the classification process of an IDS is hampered by a large amount of redundant and irrelevant data in high-dimensional datasets.

To that end, this study proposes an optimized Machine learning-based framework to detect the network attacks in a computer network. More specifically, the proposed framework consists of using chi-square as a feature selection method and randomized search cv function as a hyper-parameter for model optimization to tune the parameters of a random forest (RF) classifier. The proposed framework is evaluated using a CICIDS 2017 dataset [5] [6]. Experimental results show that the proposed optimized framework reduced the feature set size by up to 50%. Moreover, it has also achieved a high detection accuracy, precision, recall, and F-score compared to the default classifier. This highlights the effectiveness and robustness of the proposed framework in detecting the network attacks.

1.2 Introduction

The recent increase in day-to-day usage of the number of network devices, advancement in existing and new technologies and services leading to a steady increase in cyber-attacks by hackers performing malicious attacks to modify or steal the data from victim systems. To overcome these all the cyber-security network-related attacks there is an increasing demand for protective measures like Intrusion Detection System (IDS) which is one of the essential elements of network security that helps to detect the attacks by analyzing network traffic packets or operating system network logs. In Fig1.showing continuous increase in cyber-attacks count over past years and this is increased about 200%. Therefore, current scenario needs faster development in the field of cybersecurity, especially in network intrusion detection and prevention systems.

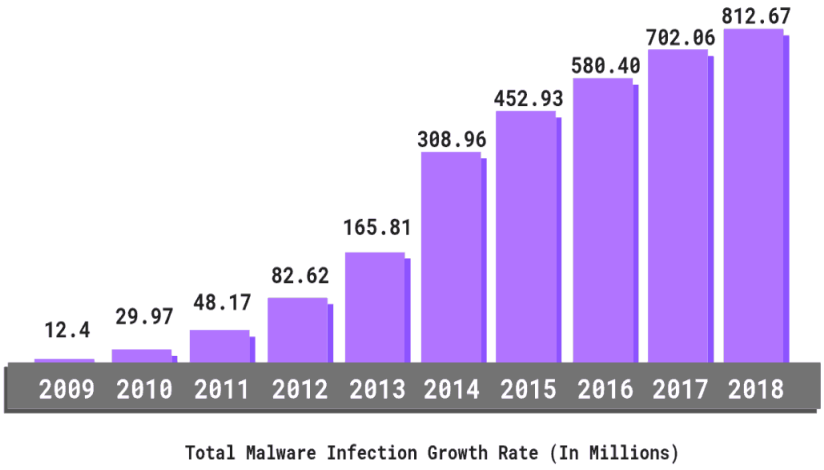

Figure 1 Cyber-attacks growth in recent years

To mitigate the risk of cyber-attacks IDS is one effective solution. Intrusion detection is a process of detecting and preventing intrusions activity on any network or system. It is one of the significant ways of preventing the computer system from intrusions. In general intrusion detection systems are software applications or hardware security programs that verify that anything occurring in the system over a network is malicious or normal in behavior [7]. There are five types of intrusion detection systems: 1) Host-based intrusion detection system 2) Network-based intrusion detection system 3) Protocol-based intrusion detection system 4) Application protocol-based intrusion detection system 5) Hybrid intrusion detection system [8]. Each of the above-mentioned intrusion detection systems has a varied procedure in detection and providing security to the data, and each of them has its pros and cons. Initiating alert to organizations when any malicious or intrusive activity is detected within network or system devices so network and host-based intrusion detection system play a vital role in cybersecurity. One of the most common problems of these intrusion detection systems is handling the large amounts of attack alerts

and the way how to handle them. Another major problem that arises due to the recent emerging of new technologies is the upcoming issues of “Big Data”, “IoT”, “Fog & Edge Computing”, “Cloud Computing” is their large network data logs [9] [10]. These emerging technologies produce a large amount of data that is multidimensional with numerous features is needed to be reduced to get an efficient intrusion detection system. It is now very surged need to implement these enormous data to be considered not because of the increasing number of tuples, but also for the number of features in each tuple that may lead to generate more false positives so redundancy of dataset classification will increase.

For any intrusion detection system, selecting a subset of features to reduce the set of non-contributing features in classification is called the pillar of an intrusion detection system. The efficiency and accuracy of a dataset have a major dependency on the performance of the classifier model on that dataset. If the dataset split into training components is less redundant and has more accurate logs then the testing component will have more improvised model performance. Therefore, it is one of the most inevitable steps to get better accuracy performance choosing a less redundant dataset for testing and training components of the system model, and gathering the network logs dataset is a big quandary as a dataset that is publicly available for use is very diverted from real scenario. There are some well-known available datasets on network attacks mostly used by the researchers are such as NSL KDD, UNSW NB-15, CICIDS 2017 datasets [11] [12] [13].

In our implementing CICDS dataset, there are a set total of 79 features. To make our dataset fit for applying machine learning classification algorithms for gaining high accuracy dataset needs to be first pre-processed, cleaning of redundant data, encoding of categorical features, scaling of features with very huge values, feature reduction should be performed first to increase the efficiency and get training time reduced.

Machine learning approaches helped us to analyze learn from very huge datasets and make correct predictions overtime on continuous learning from each outcome feature has now reduced human intervention to a greater extent in the real-time data science world.

To get the proper intrusion detection system it involves feature elimination and classification of attack category for enhancing the performance of the model to get better metric value on accuracy, precision.

The rest of the study is organized as follows. In Chapter II, the related survey of works done by various researchers is discussed briefly. In Chapter III, the proposed model is discussed and analyzed in-depth for algorithms implemented from dataset selection to performance evaluation. Different phases of intrusion detection systems are described in detail with a detailed analysis of the performance evaluation of our proposed model. In Chapter IV Conclusion and future work are briefly mentioned.

1.3 Background Study & Motivation

Intrusion detection systems (IDS) have been used in the past to safeguard networks against intruders [14]. This entails monitoring the network and identifying network attacks using attack signatures; when traffic matches a known predefined attack pattern, it indicates that an attack has occurred. Traditional IDS are good at detecting known attacks, but they're useless against unknown violent attacks, leaving the network susceptible to zero-day vulnerabilities. Furthermore, the introduction of new attack patterns demands the updating of the signature database that contains attack definitions.

1.3.1 Motivation

Machine learning techniques as the basis of attack detection are an alternate and complementary option. Over the last decade, machine learning has exploded in popularity, with applications in fields as diverse as health care, product suggestion, and email spam filtering [15].

Motivation to perform a detailed study in network security topic of intrusion detection systems is:

1) Continuously developing IoT networks, cloud computing has become an increasingly valuable target of malicious attacks due to the increased amount of valuable user data they contain. In response, network intrusion detection systems have been developed to detect suspicious network activity with more accuracy.

2) Traditional network security solutions may not be directly applicable due to the differences in the advancement of IoT structure and behavior in recent times.

3) Healthy class discussions with our project supervisor about recent issues, challenges, and solutions by machine learning and deep learning methods to solve the problem arouse interests in the field and motivated to do some contribution.

4) During this project, I have read many research and journal papers related to intrusion detection systems on various methods to solve this problem each having its pros and cons led me to work in this field more efficiently.

1.3.2 IDS an overview

Detection of intruders is the detection and prevention of intrusive activity on any network or system. These intrusion attacks may lead to serious damage to networks and systems. IDS keeps an eye on network systems to alert the admin if any malicious or intrusive activity is noticed [16]. It is a radar-like software application continuously searching for fraud activity or policy violations.

Types of Intrusion Detection System:

The intrusion detection system is of 5 types:

1. **Network Intrusion Detection System:**

A network intrusion detection system is a hardware or software-based system that detects malicious traffic on the network. In a network intrusion detection system, there are some points on the network which inspect the congestion from incoming packets to all network devices. It checks the packets for if there is an attack by the pre-determined or known existing database. If it matches to attack an alert is sent to the administrator [17].

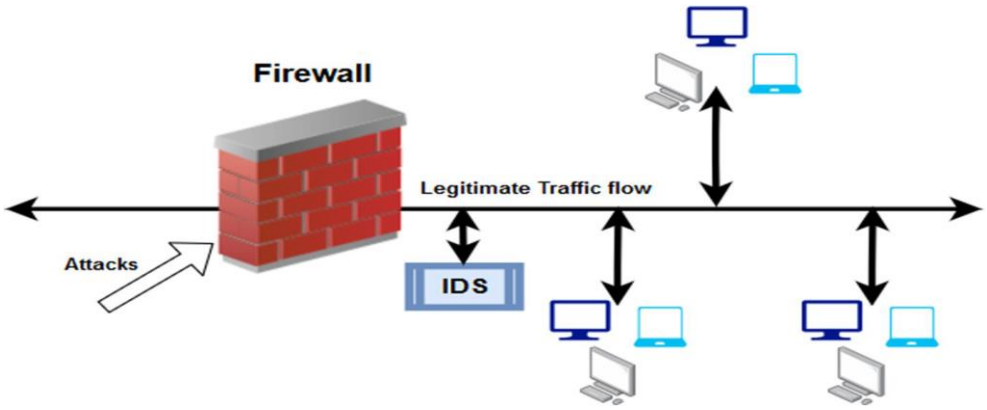

Figure 2 Network Intrusion Detection System

2. **Host Intrusion Detection System:**

A host-based intrusion detection system is an application that inspects for any malicious activity on a network and reports to the administrator if any suspicious activity is detected by operating through data on individual computers on the whole network system. Host-based IDS analyzes the network by capturing screenshots of individual hosts information logs [18].

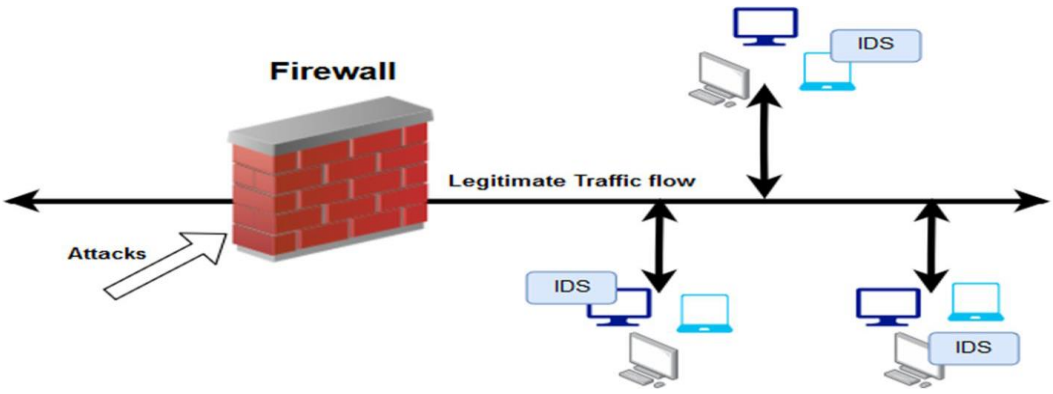

Figure 3 Host-based Intrusion Detection system

3. **Protocol-based Intrusion Detection System:**

   A protocol-based intrusion detection system is installed on the front end of the webserver which performs its operation by analyzing the protocol system used between the host network device and the webserver. Protocol-based IDS keeps an eye on the front end of a web server HTTP, HTTPS protocol stream [19].

4. **Application Protocol-based Intrusion Detection System:**

   Application Protocol-based Intrusion Detection System also monitors and analyses the dynamic network behavior from server installed application. But it typically performs its operation from the application protocol of a group of servers [20].

5. **Hybrid Intrusion Detection System :**

   If any intrusion detection system has two or more machine learning approaches are combined to perform as a single algorithm for classification then it is called a hybrid intrusion detection system. In this individual system, data is mixed with the data on the network system to get full network logs of the system [21].

Detection methods of IDS:

1. **Signature-based Method:**

   A signature-based intrusion detection system is based on detecting the attack or any malicious activity by matching the known attack pattern from existing data of pre-defined attacks. These attack patterns which are known in the database are called signatures. When any network behavior is matched with these known signatures then the intrusion detection system issues an alert alarm to the administrator that a malicious attack is happening over a network [22].

2. **Anomaly-based Method:**

   An anomaly-based intrusion detection system is based on detecting any unknown attack or malicious activity which is not known or its behavior is not in the existing database of the intrusion detection system. In an anomaly-based intrusion detection system, machine learning models are trained based on existing attack behavior and identifies for the attacks based on that available information and predictions [23].

Comparison of IDS with Firewalls:

Intrusion detection system it monitors and issues only alert if any malicious activity is detected however in the firewall it blocks any unauthorized access and also disables users to access any malicious content on the system. The major difference is that firewall can protect if any intrusive activity is happening inside the system but an intrusion detection system can alert only over a network.

- A firewall does not restrict or alerts for permitted network traffic areas while an intrusion detection system keeps an eye over the complete network.

- To function a firewall no human intervention is required but for an intrusion detection system, an administrator is required to handle the attacks when alerted by the system.

1.3.2 Machine learning an overview

Machine learning refers to algorithms that can explain data automatically with little or no human intervention [24]. Preparing the model's inputs (features), finishing the training, and eventually testing the model to evaluate how well it works are all part of a machine learning project's work flow. In most cases, the input comes from a dataset that is relevant to the problem.

There are several forms of learning. We connect a feature vector in training with a label in supervised learning, so the model learns how to predict a proper label based on the provided feature vectors. As a result, when the system is deployed and new feature labels are provided to it, it can anticipate what the associated label will be [24].

Unsupervised learning is when you learn from an unlabeled dataset. The training dataset is used to try to discover patterns or groupings in the data, and when a new feature vector is supplied to it, it reports which group it belongs to [24].

Other types of learning techniques include semi-supervised, reinforcement, and deep learning. Because they are not used in this work, the specifics of how they operate have been omitted.

Machine learning solves many different types of problems; we can group these problems into the following tasks:

- Classification outputs a predicted label from some predefined set.
- Regression outputs a predicted real-valued numerical result.

- Clustering groups objects such that objects within the same cluster have similar properties compared to that of other clusters. There may or may not be a predefined number of clusters to fit the training data.

Both classification and clustering are types of approaches which are suitable for this project. Classification aims to predict a category and assign the labels benign or attack. While clustering aims to divide the data into groups that are similar to each other. Here, we cluster the data into normal and anomalous cases.

Challenges

There are certain difficulties with machine learning. The choice of a suitable dataset is the first major difficulty. The dataset must be relevant to the job at hand and large enough to allow the model to learn.

Another common problem in machine learning is overfitting, which happens when your model does not generalize well to fresh data. When you over-refine your model, it may look that it has learned all of the noise in the dataset, which means that although the performance appears to be improving, it really degrades when tested on new data [25].

1.4 Problem Statement

For high-dimensional data spaces, the machine learning-based intrusion detection systems have decreased their performance and a very large increase in execution time of the model as compared to low dimensional data spaces in which it has higher accuracy and efficiency in detecting network attack. Therefore, it is an urgent need for implementing an appropriate feature reduction method with proper classification methods but some of the features which have a lesser dependency on target outcome variable can be eliminated so that it does not possess a great impact on the classification process and results in reducing time. There are many networks intrusion dataset that works upon any network attacks-related datasets like UNSW-NB15, NSL KDD, CICIDS, etc. which includes many irrelevant features or features having least dependency on outcome variable so it drastically decreasing the rate of intrusion detection and increasing false alarm rate. Here the aim is to propose a relevant feature extraction technique to obtain more accurate and decrease false alarm rate. Apply a proper classification algorithm to train a model which detects the attacks in a network by following network log data. Compare with other feature extraction and classifier algorithms of previously done researches with your proposed method.

1.5 Objectives

1. To identify the most significant features extraction mechanism of Intrusion detection on any network attacks dataset.

    - ✓ To achieve this objective a statistical-based feature selection algorithm chi-square is applied which selects the k best features having a top feature score (chi-square).

2. Comparing Machine learning-based approaches with the reduced feature set and all features by measuring respective performance metric score.

    - ✓ To achieve this objective after feature extraction classification on all attack labels, binary classification and multi-class classification is performed on the dataset. Linear support vector machine classifier, Naïve Bayes classifier, Random Forest classifier, Decision tree classifier is applied and compared the performance of each model.

3. Propose a method with better accuracy and performance scores on testing set than standard machine learning approaches.

    - ✓ To propose a better classification algorithm a randomized random forest classifier is proposed with different hyper-parameters selected by randomized search cv algorithm.

4. Obtain better accuracy and decreased false alarm rate with strongest and reduced features by Chi-square algorithm proposed.

    - ✓ To achieve this objective by our proposed trained model and used feature selection algorithm has shown accuracy of 99.98% in overall attack categories and has decreased the false alarm rate which results in a better intrusion detection system approach based on chi-square feature selection and proposed randomized random forest classifier.

1.6 Summary of Chapter

In this chapter, a discussion to introduction of Intrusion detection is done. It is a process of detecting and preventing intrusions activity on any network or system. The intrusion detection system is of five types NIDS, HIDS, Hybrid IDS, Protocol IDS, Application protocol IDS. There are two types of detection methods of IDS signature-based and anomaly-based detection. Signature-based detection can detect only pre-known attacks in the database while an anomaly-based intrusion detection system is capable of handling the unknown attacks on basis of information or trend learn from the huge network attacks log data. Due to continuous growth in network devices, IoT, cloud computing data files, and information security a primary-concern. In some recent reports of increasing cyber-attacks continuously as shown in Figure 1, there is a need for improvement in this field. Apart from seeing these incidents news in-class discussion with our project supervisor motivated me to give some contribution in this field. Machine Learning (ML), Deep Learning (DL), and nature-inspired approaches are applied to design and develop more efficient and intelligent security solutions to fight against fraudsters or attackers. Cloud computing has become popular among the business community due to its cost effectiveness and other advantages. Security is also one of the paramount of the cloud environment, and hence efforts are made to propose enhanced NIDS. My problem is to find the best minimal subset of features to get better accuracy on the classification of attacks category. For this problem some objectives are defined to achieve that are selecting relevant features subset, applying proper classification model, comparing the metric score such as accuracy precision-recall f score with that of previously done researches. To perform this work CICIDS 2017 dataset is chosen for our works. CICIDS 2017 dataset has 80 features and 12 types of attacks mainly listed into 7 categories DOS, Probe, DDoS, Brute force, Benign, Botnet, Web attack. Each attack category has some attacks from the dataset which is properly listed in Table 1. for the CICIDS dataset. I have discussed the relevant background knowledge required for this dissertation, my success criteria and project requirements, how I planned to implement the project and my starting point.

Chapter 2: Literature Survey

2.1 Related Tools & Technologies

Technology and tools used for this study are:

1. **Python 3.8.5:** Python is one of the most powerful dynamic programming languages. Python is also called an interpreted scripting language. The programming language in our implementation of machine learning algorithms is python's 3.8.5 version.

2. **Anaconda Software:** Anaconda is software used for machine learning, deep learning, data science for python and R programming languages. It has a collection of individual software packages named Jupyter Notebook, Spyder, Anaconda navigator, etc.

3. **Jupyter Notebook:** Jupyter Notebook is an open-source software also known as a python notebook included in Anaconda software which is capable of creating python notebooks mostly used for data science machine learning, deep learning. Jupyter notebook has filed with the extension ".ipynb".

4. **Machine Learning:** Machine Learning is the field of study in which machines are trained on basis of known patterns and are made able to think like humans. Machine learning is now used up in almost all fields to reduce human intervention and a better way of prediction and analysis.

5. **Python Libraries:** Some most common python libraries used in machine learning are used such as Pandas, numpy, sci-kit learn, matplotlib, etc.

2.2 Related Works

In most of the previous researches, the intrusion detection systems proposed approaches to select the relevant subset of features to improve accuracy rate and reduce false alarm rate. IDS attracts the attention of the scientific community as a critical instrument in computer-based systems for guaranteeing cyber security. Although many methods have been presented to improve the performance of IDS, we only include related work that comes within the ML based IDS, employs feature selection or an optimized classifier algorithm, or focuses on hybrid techniques in this area.

Rafael G. Mantovani et al. [26] investigated the approaches of random search and grid search. They wanted to fine-tune the Support Vector Machine classifier's (SVM) hyper-parameters. They used a large - scale dataset to conduct their experiment, and then compared the performance of Randomized Search with four other methods: Ant Colony Optimization, Evolutionary Algorithms, Grid Search Method, and Estimation of Distributed Algorithm. The results of this study show that the predictive power of the SVM classifier combined with Random Search is comparable to the other four methods employed, with the benefit of the model's cheap computing cost.

Ahlam Alrehili and Kholood Albalawi [27] conducted an ensemble-based customer review sentiment analysis in 2019. The suggested technique combined five classifiers: Random Forest, Naive Bayes, SVM, bagging, and boosting, into a voting system. The authors run six distinct scenarios to compare the proposed model's results against five other classifiers. They remove stop words from unigrams (with and without stop words), bigrams (with and without stop words), and trigrams. The Random Forest classifier has the greatest accuracy of 89.87 percent out of all of them.

In each file of the CICIDS2017 dataset that contains a DDoS attack, SMOTE was used to improve the sensitivity of the arrangement for minority classes. By using SMOTE with a minority oversampling class of 200 percent, the researchers were able to raise the number of minority class occurrences (DDoS) in training data from 29285 to 87855. The AdaBoost classifier was used to determine the performance metrics of the training data, along with other feature selection methods such as PCA and EFS [28]. With an accuracy of 81.83 percent, precision of 81.83 percent, recall of 1, and F1 Score of 90.01 percent, the results show that their suggested approach exceeds prior literature.

To manage the unbalanced distribution of minority class instances in the CICIDS2017 dataset, a uniform distribution-based balancing (UDBB) technique was developed [29]. The researchers consolidated all of the data files from the Monday CICIDS2017 dataset into a single file. Their research contrasts the imbalanced scenario (with CICIDS2017's original distribution) to the balanced case (with CICIDS2017's original distribution) (after applying the uniform distribution-based balancing

approach). Random Forest, Bayesian Network, LDA, and QDA classifiers with 10 features were used to compute training data performance measures. The accuracy of RF, NB, LDA, and QDA after using UDBB was 98.8 percent, 97.6 percent, 95.7 percent, and 98.9 percent, respectively. RF, NB, LDA, and QDA haves F measures as 98.8 percent, 97.7 percent, 95.7 percent, and 99.0 percent, respectively.

A review of network intrusion detection system methods, types, and technologies, as well as their benefits and drawbacks, was done by Khraisat et al. [30]. The different machine learning approaches that have been recommended for detecting zero-day attacks are shown. The recommended techniques, on the other hand, have difficulty providing information about any undiscovered new threats, are less accurate, and have a high false alarm rate. On the most common public datasets such as NSL KDD, CICIDS, and others, a comprehensive comparison of various previously known research outcomes is evaluated and compared. Recent research were summarised, and contemporary approaches for enhancing NIDS performance as a solution to NIDS difficulties were investigated.

An empirical study was conducted to show that the algorithm indeed lives up to what it claims presents an FS using Genetic Algorithm with Support Vector Machine (SVM) for mining the medical dataset [31], results from the work show that models built with reduced features give a higher diagnoses rate and lower miscalculation rate. Findings from [32] showed improvement in the performance of the proposed model by metric score values accuracy, precision, recall, f-score with reduced selected features, and understanding ability of the learning process. Empirical evaluation of the consistent feature selection measures with wrapper method shows consistency in method performing efficiently more than the wrapper method.

A. Alazab et al. [33] A good intrusion detection system must hold an essential functionality to give its attacks classification results more accurate and efficient gives its classification of attacks results more accurate and efficient. The dataset that used in this experiment is NSL-KDD. In this research, the focus was to train multiple functions to achieve highly accurate results. Training and testing time of the proposed model was a concern to be addressed in their research. To address this problem, proposed a feature selection technique based on values of information gain with J48 classification, which makes it capable to detect several attack categories with high accuracy and low false alarm rate. Dataset has been divide into five parts according to the attack categories (DoS, Probe, R2L, U2R, Normal). In this experiment, the final accuracy which was achieved for attacks classification is 98.20% for Normal, 99.60% for Probe, 99.70% for R2L, 97.2% for DoS, 92.50% for U2R. Execution time that reduced from 1.73ms to 0.3ms for 41 features to 12 proposed relevant features after feature extraction.

Bahareh Abolhasanzadeh et al. [34] in this research proposed an intrusion detection system by experimenting on the NSL-KDD dataset. In this study, there are four feature selection methods are used and compared with the ANN classification algorithm are Principal Component Analysis (PCA), Factor Analysis, Kernel-PCA, and Bottleneck (Autoencoder).

1. Principal component analysis (PCA) by name it is clear that there is an analysis of principal components. Hence, this method is a linear feature extraction technique based on computing the combinations of the possible features that are giving the largest variance and selects that subset of features. It can be applied only to the linearly separable data.

2. Kernel-PCA is another modified PCA that uses kernels to extract the optimal features. It uses PCA but to separate the decision boundaries non linearly function is used for high dimensional features dataset.

Factor analysis is a statistics-based linear method that extracts features based on factors depending on the number of features observed.

The subset of features extracted from the above-proposed feature extraction methods is now used for attack classification using an artificial neural network classifier. The results that came out after this experiment have an accuracy of 86.78% for PCA, 88.31% for Kernel PCA 88.53% for factor analysis feature selection algorithms. To improve this achieved accuracy an auto-encoder is used named Bottleneck autoencoder so the highest accuracy rate evaluated from this study was 91.46%.

Omar et al. [35] In this study, a hybrid intrusion detection system is proposed that uses a combination of supervised and unsupervised machine learning algorithms, such as K-means, fuzzy C-means, and gravitational search clustering algorithms, to obtain an optimal subset of features by clustering groups of features with similar importance. To improve the accuracy of the attack categorization in this work, a hybrid classifier integrating support vector machine and gravitational search method is suggested. As its name suggests, the gravitational search algorithm is based on the laws of gravity and motion. Using a k-1 segregator, the K-mean clustering technique divides all features into k subsets and partitions them according to their feature significance. Fuzzy c-means clustering is a type of clustering in which parts of the data are included in several clusters. This clustering technique is one of the most widely used and significant. Despite this, the total accuracy attained was poor, and further work is needed.

2.3 Summary

In this chapter, a description of some tools and technologies which are used in this work and study were done by other researchers in the field of intrusion detection system their works are discussed with the accuracy of their proposed algorithm attained and any improvement suggested. The tools and technologies used are Python 3.8.5 programming language is used for my work. Some of the famous python libraries are also used like pandas, numpy, matplotlib, sci-kit-learn, etc. Pandas library is used for reading datasets in CSV format. Numpy is used for handling numerical data in our dataset. Matplotlib is used for graphical representation of results dataset prediction and evaluation of our algorithms. Scikit-learn is one of the most famous used library modules for machine learning concepts in python sci-kit learn is used for data pre-processing and also enables us to use all classifier regression algorithms directly by importing them from the module. Several machine learning algorithms for classification are discussed in the above section are Naïve Bayes, Support vector machine, Decision tree classifier, K-nearest neighbor classifier, and random forest classifier. Many proposed works of other researches discussed with many different algorithms some of them are as in A.Alzaab [33] in this study information gain based feature selection is proposed and for classification of attacks J48 classifier, the method is proposed. PCA, Kernel-PCA-based feature selection is proposed by Bahareh Abolhasanzadeh [34].

Chapter 3: Proposed Work and Implementation

3.1 Proposed Work

In this section, a framework to solve the problem statement a work is proposed to achieve all the goals and objectives is depicted in following flow-diagram Figure 4. and brief later.

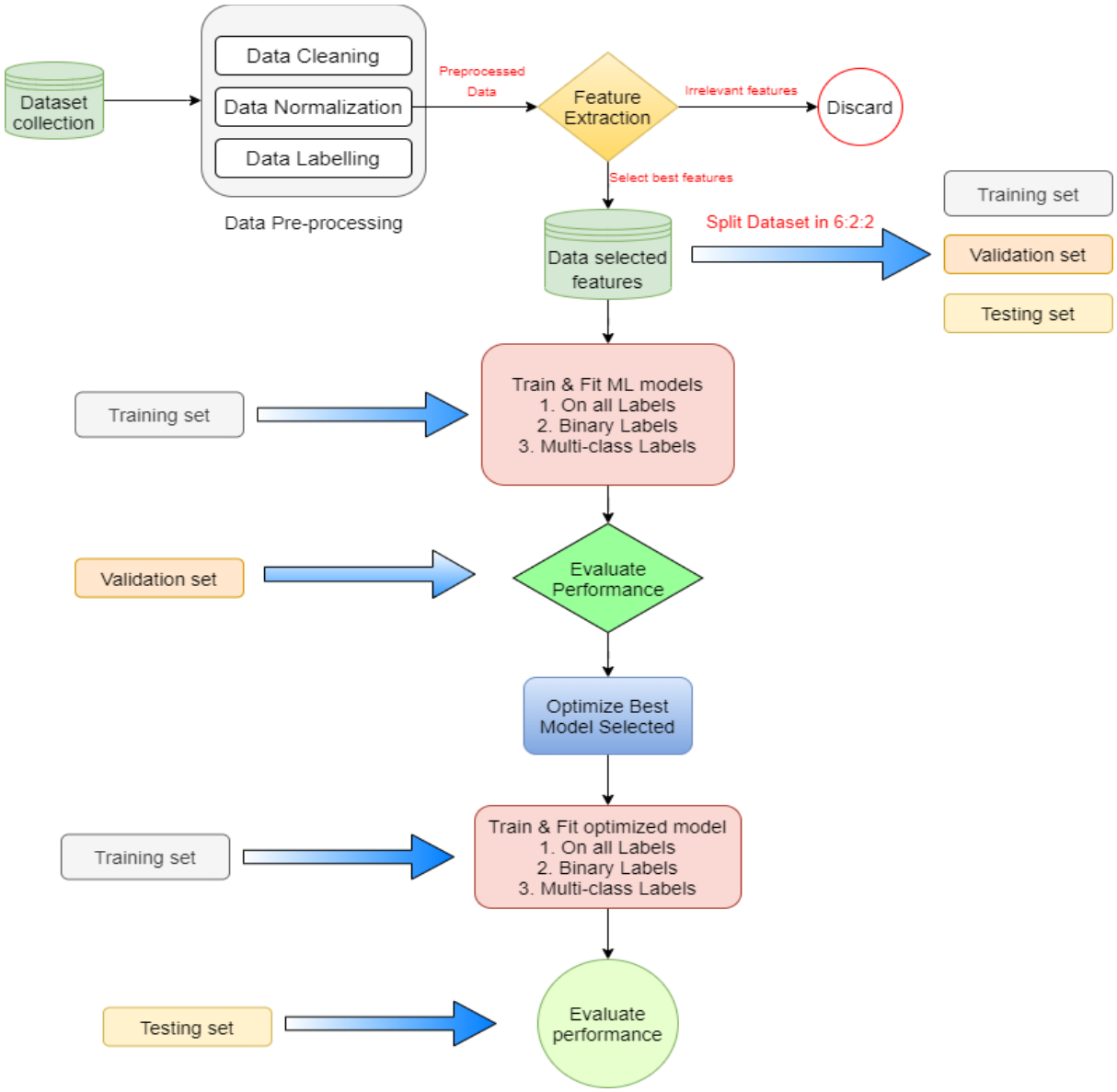

Figure 4 Proposed Work Flow Chart

3.1.1 Data Collection and Pre-processing

Data gathering and pre-processing are the first steps in any network intrusion detection system. The dataset used in this work is CICIDS 2017 dataset. Data pre-processing is an important stage since it increases data quality and the enhances the overall performance of the proposed approach. Data preprocessing includes data cleaning, data encoding, data labelling, dataset splitting and data normalization:

Data Cleaning

Data cleansing, also known as data scrubbing, is an important stage in the preprocessing process since it reduces redundancy and noise. Furthermore, the information gathered may be insufficient, erroneous, duplicate attributes, redundant attribute or contain null, infinite values. All discrepancies are removed during data cleansing. The Python environment is utilised to complete the task in this project.

Data Normalization

Normalization of a feature or characteristics in data is a process that reduces the upper and lower bounds of numerical values in data to a smaller range, usually [-1,1] or [0,1], also the information values are preserved. Normalization is necessary because significant variances in distinct data attributes might cause problems when applying classification algorithms. Normalization allows new values to be generated within a given range without changing the overall distribution or outcomes. In this work, data normalization is performed by min-max scaler [36].

For example, if attributes have values -10, 234, -657, 987, 213 then to normalize this data, first check for the largest digit value and then divide each by the largest number of that digits for this example largest is 987 so divide by 1000 each then, normalized data will be -0.01, 0.234, -0.657, 0.987, 0.213. This method is called decimal standard scaling.

Data Encoding

Some categorical attributes in typical datasets contains some values in the form of strings that are challenging for machines to read and analyze. To improve the categorization relevance of these features, numerous encoding methods should be used to turn these sorts of data features into numerical data types. There are a variety of encoding techniques available, including one-hot encoders, label encoders, binary encoding, and so on. One-Hot-Encoding is a binary representation in which all values are zero except for one, which has a value of one. This method of variable encoding is used to extract binary features from a set of categorical features.

If a categorical feature has n distinct values, one hot encoding creates n binary datatype columns for each distinct value of the feature.

Splitting the dataset

The dataset is divided into subsets so that the machine learning model's performance can be evaluated using additional data that was not used to train the model. Another reason is to choose the train-test split assessment process, in addition to dataset size, is computational efficiency. The CICIDS 2017 dataset is divided into three components in this study: training set, validation set, and testing set, in the ratio of 6:2:2 respectively. The dataset is stratified using the y attribute, which is our target variable 'Label,' such that proportion of attacks is consistent across all three datasets.

Attack Labelling

Attack labelling is a step to be conducted in any network intrusion detection dataset to execute experiment as label attribute present in dataset has multiple attack classes. To accomplish binary classification labelling has to be done into two categories one is normal and other is attack all the attack classes except normal i.e. benign in our dataset are clumped in a single label called attack. Labeling is done in groups while doing multi-class classification. For example, the highly benchmarked dataset NSL KDD comprises 22 attacks classified into four attack categories: DoS, Probe, R2L, and U2R. Similarly, the attack classes in the CICIDS 2017 dataset are classified into seven attack categories.

3.1.2 Feature Selection

Feature selection is a difficult operation that involves extracting the right network feature set. The process of finding the features that contribute the most to identifying the optimal class prediction to the target variable outcome is known as feature selection. The adoption of an optimal feature selection technique reduces data size, training time, and detection time while also protecting against overfitting, resulting in total network intrusion detection system performance improvement [37]. High-dimensional data is vast in size and contains redundant and useless elements that slow down any system's performance. Many ways of dealing with the curse of high dimensionality have been presented by the scholars [ref]. However, a better strategy for feature selection optimization remains to be developed. For example, if there are 10 features with their different ratio of contribution to target variable outcome as 0.92, 0.2, 0.11, 0.84, 0.97, 0.05, 0.79, 0.123, 0.976, 0.01 so the feature with very low contribution can be removed from the dataset for classification here only 5 features can be taken into consideration which has their dependency score above 0.7, as it has multiple benefits for improvement in model performance.

There are many feature extraction algorithms like chi-square, genetic algorithm, recursive feature elimination, Pearson's correlation coefficient, ANOVA F-test, etc. [38] [39] In this work, chi-square function with select k best feature technique for extracting relevant features is used.

3.1.3 Classification algorithms

There are many standard supervised and unsupervised machine learning algorithms. Implemented at some of the machine learning algorithms and evaluated its performance score on a proportion of the dataset reserved for model validation.

Support Vector Machine

For classification, an SVM utilizes an optimum hyperplane, which is a line in 2D space, a plane in 3D space, and so on. In other words, the hyperplanes are created by the SVM during its training phase to optimize the margins between classes [40].

Decision Tree

This classifier builds a tree from top to bottom using a flowchart-like structure. The internal node in the tree represents a "test" on an attribute, while the branch represents the test's result. After computing all characteristics, the leaf node is equal to a class label or decision as a result [41].

Naïve Bayes

This classifier is based on the Bayes' Theorem, which states that several algorithms from the same family can be employed since they have a common principle. Each pair of categorized characteristics is independent of the other in this case. This classifier is straightforward and efficient. It can quickly create an ML model and forecast. Because it predicts using probability, it's also known as a probabilistic classifier. Every classifier in supervised learning must go through a training and testing phase. After optimizing the feature set, which picks the appropriate set, the dataset is created. Training and testing data for selected characteristics are split into two groups [42].

KNN

k-NN is a straightforward supervised learning technique that performs well on big training sets. It is assumed that data points with comparable properties would provide similar results. The k-NN technique uses the majority neighbors among nearest neighbor to estimate an unknown data sample. It is resistant to noisy training data and works well with a large number of instances. The distance between each instance and all training data samples takes longer to compute [43]. K-NN classifies using a majority vote on neighboring locations, thus it should perform well if we believe that similar sorts of attack feature vectors are close together.

K-means clustering

K means cluster is an unsupervised clustering technique that divides objects into K disjoint clusters based on their feature values (where K is a positive integer specifying the number of clusters). It groups data based on their Euclidean distance, grouping items into the same cluster if their feature values are comparable [44].

Random Forest

Random forest classifier creates several decision trees from the dataset provided as input during training to operate on the unknown data. Each decision tree votes for a certain class based on its training for unknown input, and the majority vote determines the class of unknown data. It can successfully cope with large and complex datasets, making it suitable for an IoT network [45].

Model Training & validation

All above mentioned algorithms are implemented to train the classifier model on training data. Each classifier model is fitted in three categories first one on all attack labels in dataset, second one is for binary classification, third one is for multi-class classification by grouping all attack classes into some categories. After training the model on training data model's performance is evaluated and compared on basis of various performance metric as explained in later section is used to find out the best performing model on our validation data.

Optimization of model by selecting the best performed model

Among all above machine learning classifier algorithms model is validated by its performance scores and analyzed to select one of the best performed model. The model once selected is used to optimize in order to increase accuracy and overall model performance. Optimization of model refers to tuning of hyper-parameters of algorithm so that it works on a selective way to achieve improvement in classification of network attacks in an intrusion detection system [46].

After optimization of model again apply the tuned classification algorithm in each category as above to train model on training set and analyze the proposed model performance by studying detailed confusion matrix and performance scores.

3.1.4 Performance evaluation

The testing and validation datasets are now being used to forecast and evaluate our model based on several performance metrics like as accuracy, precision, recall, f-1 score, and a confusion matrix. The

number of right and erroneous predictions generated by the model in comparison to the actual outcomes (target value) in the data is shown in a confusion matrix.

True positives occur when a forecast of a result is attacked, and in actuality, it is also an attack, implying that positive is anticipated to be positive. True negatives occur when the result forecast is normal, and the result is normal in fact, implying that negative is anticipated for negative. False positives occur when a normal result is expected, but the result is actually attacked, i.e., a negative is forecasted as a positive. False negatives occur when the expected outcome is attack but the actual outcome is normal. The confusion matrix is used to compute all of the aforementioned performance metrics.

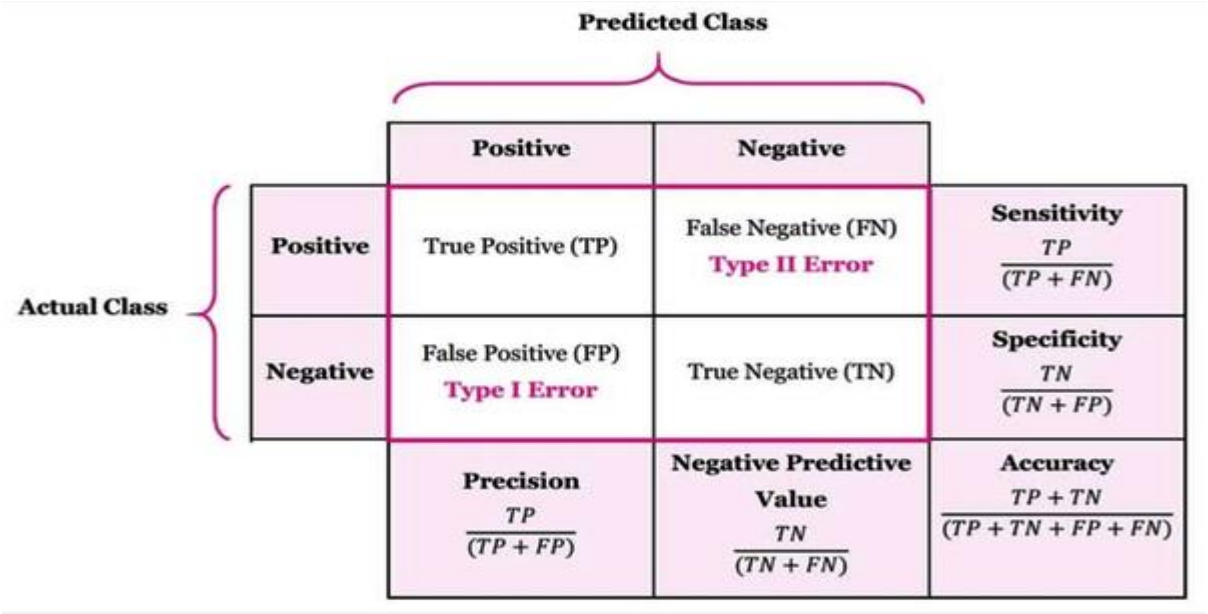

Figure 5 Performance evaluation by Confusion Matrix

- Accuracy is calculated by dividing true forecasts by total predictions and calculating the percentage of accurate predictions.

- Precision is defined as the fractions of an assault that are predicted true in reality for all attacks.

- The recall is defined as the fractions of prediction that attack, and it is also an attack with complete attack anticipated in reality.

- The Precision and Recall numbers are used to compute the F1 Score. In general, the F1 score is more important in determining the metric score for comparison. The harmonic mean of accuracy and recall is used to get the F1 score.

3.2 Implementation & Results

In this section, Implementation of proposed work done on CICIDS 2017 dataset is explained in detail. Methods used in implementing the proposed work is depicted in a flow chart in below Figure 6.

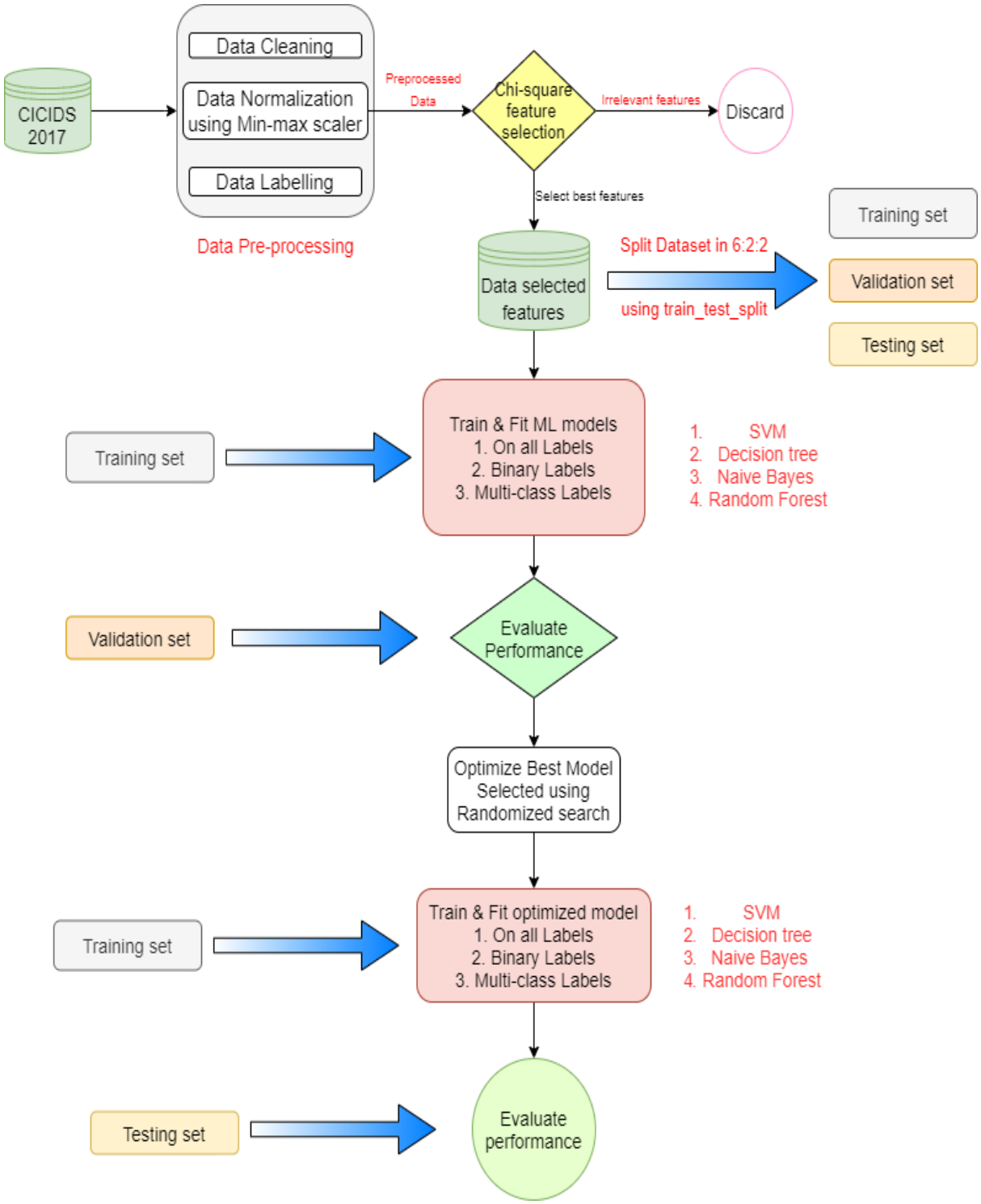

Figure 6 Flow chart of implemented work

3.2.1 Data Collection and Pre-processing

The need for a dataset to run an intrusion detection system is the most important task to do first. A benchmark network data should be utilized to verify the efficacy and better performance of our work while creating a network intrusion detection system. A network intrusion detection systems dataset CICIDS 2017 is utilized in this study to detect network attacks.

Analyzing CICIDS-2017 dataset

This dataset provides the realistic background of twenty-five users' network traffic events. Users' profiles were specified, with protocols such as Secure Shell (SSH), HTTP, HTTPS, email, and FTP included. This data collection includes a variety of common assault scenarios. It is available in Packet Capture (PCAP) and CSV file formats to the general public [47]. To complete our task in this work, we used eight CSV files, which we combined into a single csv file called cicids.csv.

The CICIDS-2017 dataset was generated from actual traces of benign and fourteen types of attacks extracted from network traffic data, with a total of 2,827,876 entries in the dataset. The benign traffic has 2,271,320 records, whereas the assault records have 556,556 records [48]. The CICIDS-2017 dataset has total 79 attributes. This dataset is one of the most popular, and it contains fresh attacks as a distinctive feature. The distribution of fifteen class labels in the CICIDS 2017 dataset is shown in Table 1. below.

Table 1 All attack label counts in CICIDS 2017 dataset

Attack labels in CICIDS 2017 dataset	Attack Count
Benign	2,271,320
DoS Hulk	230,124
Port Scan	158,804
DDoS	128,025
DoS Golden Eye	10,293
FTP Patator	7,935
SSH Patator	5,987
DoS Slowloris	5,796
DoS Slowhttptest	5,499
Bot	1,956
Web attack: Brute Force	1,507
Web attack: XSS	652
Infiltration	36
Web attack: SQL Injection	21
Heartbleed	11

All the attack labels of CICIDS 2017 dataset are briefed below:

- DoS Hulk: The attacker uses the HULK tool to launch a DoS assault on a web server.
- Port Scan: In this attack, an attacker hits packets with shifting destination ports in an attempt to capture information such as the operating system and active services on the targeted computer.
- Distributed Denial of Service (DDoS): In this assault, the attacker uses many computers under the control of a server system to launch a distributed DoS attack on the victim workstation.
- DoS Golden Eye: The attacker uses the Golden Eye tool to launch a DoS assault.
- FTP Patator: In this case, the attacker uses FTP Patator to execute a brute force attack on the FTP login password.
- SSH Patator: In this case, the attacker uses SSH Patator to guess the Secure Shell (SSH) login password using a brute force attacks.
- DoS Slow Loris: In this assault, the attacker employs the Slow Loris tool.
- DoS Slow HTTP Test: In this test, the attacker uses an HTTP request to flood a server with HTTP requests.
- Botnet: In this case, the attacker uses trojans to gain control of target workstations in order to create a network from which to launch attacks remotely. Bot is the name of the hampered machine.
- Web Attack: Brute Force: The attacker uses a trial-and-error technique to get access to user personal data such as passwords and Personal Identification Numbers (PINs).
- XSS Web Attack: In this assault, the attacker sends malicious scripts to trustworthy websites to carry out the attack.
- Infiltration: In this type of infiltration, the attacker employs tactics to penetrate and gain unauthorized access to a system within a network.
- Web Attack: SQL Injection: In this attack, the attacker attempts to access or change private data by inserting SQL queries into an entry field on data-driven applications.
- Heart Bleed: The attacker uses the OpenSSL protocol to root harmful material in OpenSSL memory and get unauthorized access to confidential information.

Data Cleaning

In this work, Dataset is very large so that it may contain redundant and duplicate data which has no contribution toward the targeted outcome. Removal of these redundancies is called data cleaning.

After loading and analyzing the dataset it is checked that whether the dataset has any duplicate columns or not if there are any duplicate columns present it is dropped. On applying I found that one attribute named 'Fwd_header_length is duplicate repeated twice so one instance of it is removed. Now dataset is remained with 78 attributes.

Now dataset is checked for any 'Null', 'NaN', 'infinite' values for any attribute is there or not if present it is replaced with 'NaN' and further all tuples having any attribute value 'NaN' is dropped.

Now, the dataset is checked for any attribute if there which have only one unique value throughout the dataset. As only one unique value for an attribute makes it redundant for classification problems because it has no contributions to the outcome. On finding this I found that there are 8 attributes which have only one unique value so this is also removed. Now dataset is remained with only 70 attributes.

Label counts of each attack is then analyzed as shown in Table 1. So, it came in notice that last 3 categories of attacks i.e. Infiltration, Web attack: SQL Injection, Heartbleed are very rare as compared to others in dataset so, I neglected those tuples as 20-30 attacks in 28 lakhs of logs. However, it may not impact a large in the performance but may have a very little.

After this dataset becomes cleaned, and ready for pre-processing which involves Data normalization and Data encoding, Data splitting operations are performed.

Data Normalization

The normalization of feature or attribute is done to limit the numerical values of data in a range (usually 0-1) without affecting range differences of actual values or losing the information. Its main purpose is to make the numerical data into a smaller range so that computation becomes faster and efficient without any erroneous data. For example, if the first column values range from 0 to 1, and the other column values range from 10,000 to 10,00,000, the variance in the two columns can lead to problems in modeling and analysis. Some most common methods of data normalization are Decimal scaling, Min-max normalization, Z-score normalization. In this work I implemented Min-max normalization. Normalization helps in the generation of new values within the specified range without affecting the general distribution and results. Eq. (1) is used to normalize the values of various attributes present in the datasets.

$$X' = a + \frac{(x - \min(x))\ (b - a)}{\max(x) - \min\ (x)}$$

Equation 1 Min-max Normalization equation

Above equation represents the way of doing min-max scaling of numerical features where, X = feature value, a = lower boundary range of feature, b = higher boundary of feature value, Min(x) and Max(x) denotes the minimum and maximum value of all tuples in that feature attribute respectively.

Data Encoding

Dataset is checked for the data type of all the attributes present in CICIDS 2017 dataset the attributes with integer and float data type needs no change but if any attribute is found with object datatype i.e. categorical value then before proceeding feature selection and model training it must be encoded. Here in this work, CICIDS 2017 dataset on verifying I found no attribute categorical other than 'label' hence, there is no requirement for applying any methods to encode the data. Now our data is ready for feature selection process.

3.2.2 Feature Selection

Extracting the appropriate network feature set is referred to as feature selection and is a challenging task. The use of an optimized feature selection technique reduces data size, training time, and detection time and safeguards against overfitting, leading to overall performance enhancement of the NIDS system [49]. Importance of feature selection in any intrusion detection system is that number of input attributes is reduced so there is steep decrease in computational cost of model training and, in some circumstances, it led to increase the model's performance.

There are two ways of feature selection methods:

Supervised methods: Supervised methods are those which uses output attribute for removal of irrelevant features. Supervised methods are further divided into three types:

1) Wrapper: Look for feature subsets that perform well.
   Example: Recursive feature elimination.
2) Filter: Subsets of characteristics should be chosen depending on their link to the objective.
   Example: Statistical methods, Feature importance methods.
3) Intrinsic methods: Algorithms that choose features automatically during training.
   Example: Decision trees.

Unsupervised methods: Unsupervised methods are those which do not uses output attribute for removal of irrelevant features.

Example: Correlation based. There are many feature selection methods available like ANOVA F-test, correlation coefficient, Chi-square, etc. Various optimization techniques can be used for feature

selection but for feature selection in this work on CICIDS 2017 dataset chi-square scoring function with select k best method is used.

Chi-square Feature selection method

In statistics, the chi-square test is used to determine if two occurrences are independent. For categorical characteristics in a dataset, the Chi-square test is employed. We calculate the Chi-square between each feature and the target and choose the features with the highest Chi-square scores [50]. It examines if the sample's connection between two categorical variables reflects their true association in the population.

Chi square scores is calculated as:

$$X^2 = \sum \frac{(Observed\ frequency\ - Expected\ Frequency)^2}{Expected\ Frequency}$$

Equation 2 Chi-square equation

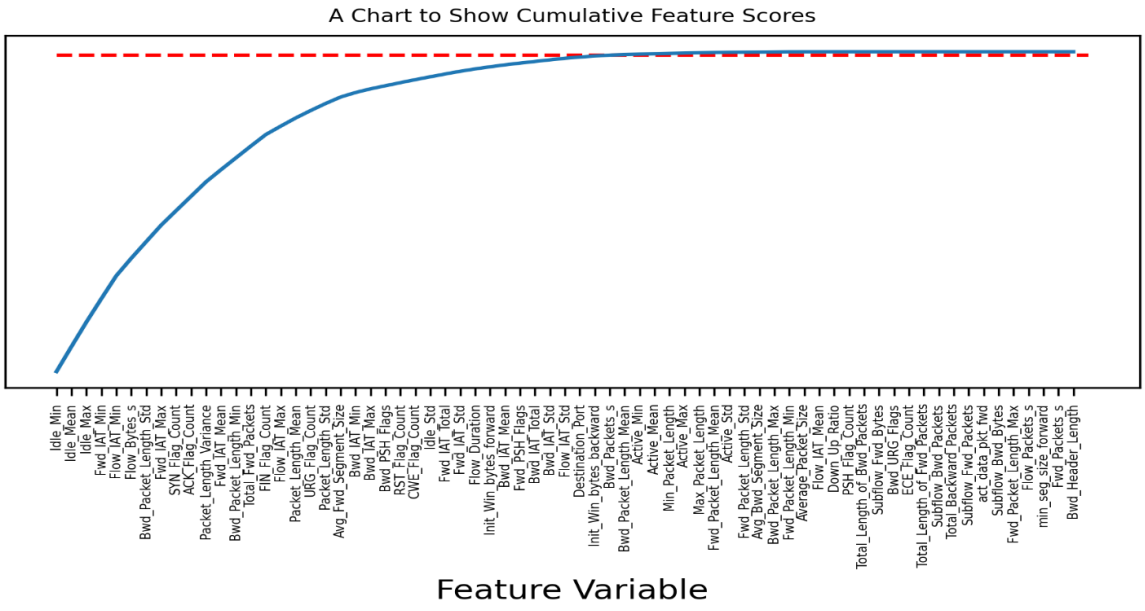

Figure 7 Cumulative feature scores using chi-square

In our work, after calculating feature scores a graph is plotted is shown in Figure 7. showing cumulative feature importance of all the features it is observed that only approx. 40 features out of all features are only more relevant which can give almost 99% of information so it is best to select only those features which also reduces the training time of model. By Select K best method using chi-square as feature scoring parameter 40 most relevant features were selected for building model and evaluation by above mentioned method and feature selected are shown in snapshot of code below:

Feature Selection

Selecting K-best features by using chi2 scoring function for features

```
In [28]: features = SelectKBest(score_func=chi2, k=x_train.shape[1])

         #fit features to the training dataset
         fit = features.fit(x_train, y_train.Label)

In [29]: # perform selectkbest with k=40

         features = SelectKBest(score_func=chi2, k=40)
         fit = features.fit(x_train, y_train.Label)

         x_train = fit.transform(x_train)
         x_test = fit.transform(x_test)

In [30]: new_features = dataset_copy.columns[features.get_support(indices=True)]

In [31]: print('Number of features selected :',len(new_features))
         new_features

         Number of features selected : 40

Out[31]: Index(['Destination_Port', 'Flow_Duration', 'Total_Fwd_Packets',
                'Bwd_Packet_Length_Min', 'Bwd_Packet_Length_Mean',
                'Bwd_Packet_Length_Std', 'Flow_Bytes_s', 'Flow_IAT_Std', 'Flow_IAT_Max',
                'Flow_IAT_Min', 'Fwd_IAT_Total', 'Fwd_IAT_Mean', 'Fwd_IAT_Std',
                'Fwd_IAT_Max', 'Fwd_IAT_Min', 'Bwd_IAT_Total', 'Bwd_IAT_Mean',
                'Bwd_IAT_Std', 'Bwd_IAT_Max', 'Bwd_IAT_Min', 'Fwd_PSH_Flags',
                'Bwd_PSH_Flags', 'Bwd_Packets_s', 'Packet_Length_Mean',
                'Packet_Length_Std', 'Packet_Length_Variance', 'FIN_Flag_Count',
                'SYN_Flag_Count', 'RST_Flag_Count', 'ACK_Flag_Count', 'URG_Flag_Count',
                'CWE_Flag_Count', 'Avg_Fwd_Segment_Size', 'Init_Win_bytes_forward',
                'Init_Win_bytes_backward', 'Active_Min', 'Idle_Mean', 'Idle_Std',
                'Idle_Max', 'Idle_Min'],
               dtype='object')
```

3.2.3 Data Labelling and Splitting for classification

Since classification is done into 3 different types for each algorithm so, there is separate output variable for each type of classification.

1. **Binary Classification**

   On binary label classification is performed by each machine learning algorithm explained in above proposed works. Output attribute in this classification is ‘Attack’ created. Its description is shown below in Table 2.

Table 2 Binary Labels of Attacks

Attack Categories	All Attack labels
Normal	Benign
Attack	DoS Golden eye, DoS Slowloris, DoS Slowhttptest, DoS Hulk, Port scan, DDoS, FTP Patator, SSH Patator, Bot, Web attack: Brute Force, Web attack: XSS

2. Multi-class Classification

On multi-class classification is performed by each machine learning algorithm explained in above proposed works. Output attribute in this classification is 'Label_category' created. Its description is shown below in Table 3.

Table 3 Multi-class label of attacks

Attack Categories	All Attack labels
Benign	Benign
DoS	DoS Golden eye, DoS Slowloris, DoS Slowhttpptest, DoS Hulk
Probe	Port scan
DDoS	DDoS
Brute force	FTP patator, SSH Patator
Botnet	Bot
Web attack	Web attack: Brute Force, Web attack: XSS

3. Classification on all attack label

On All attack label classification is performed by each machine learning algorithm explained in above proposed works. Output attribute in this classification is 'Label'. Its description is shown below in Figure 8.

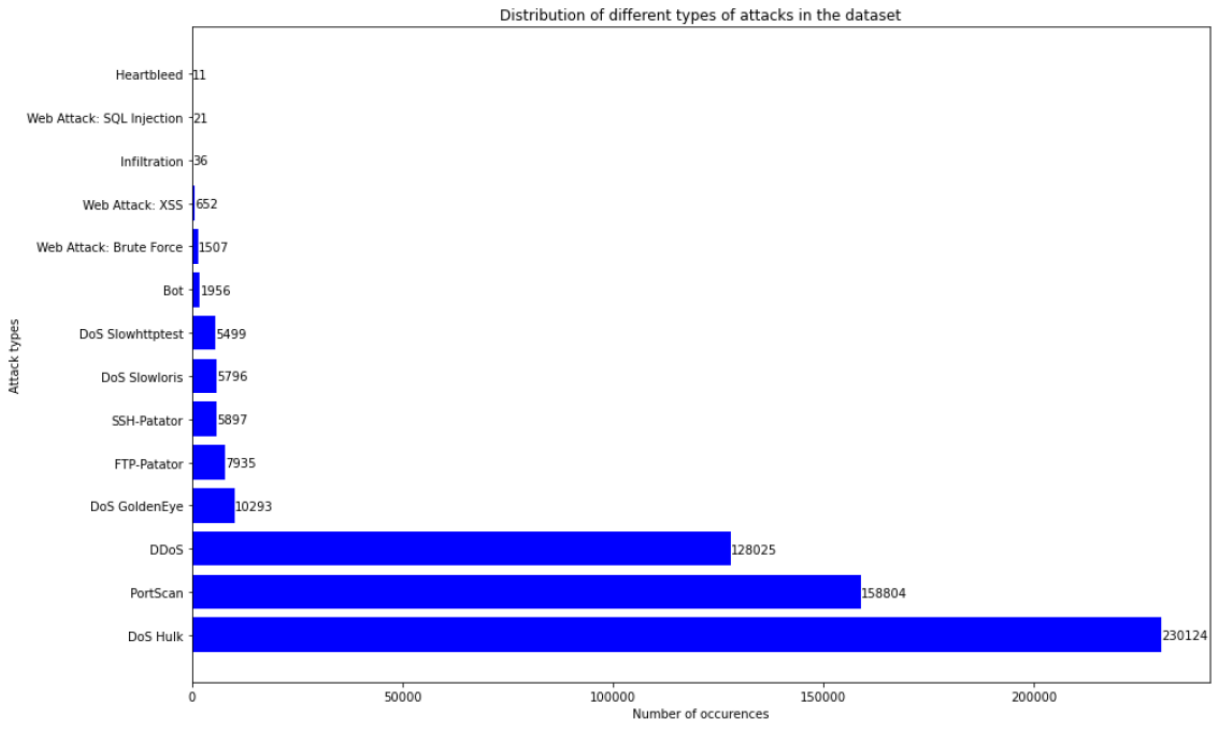

Figure 8 All attack labels plot

In this work, after preprocessing and feature selection the dataset is split into 3 parts first one is training set, testing set, validation set. Splitting of dataset is done by train_test_split method from sci-kit learn library. The train-test split is a method for assessing a machine learning algorithm's performance over a dataset. It may be applied to classification and regression issues, as well as any supervised learning technique. The technique entails splitting a dataset into subgroups. Here CICIDS 2017 dataset is split into ration 6:2:2 in training, validating and testing respectively.

3.2.4 Model training on Machine learning classification algorithms

Now on preprocessed data feature selection is also applied and is reduced to 40 attributes one by one machine learning algorithm is applied and analyzed the results.

For each machine learning algorithms is applied in 3 different categories:

- For all Attack labels: In this category there are total 12 attack labels.
- Binary Classification: In this category there is only two label normal and attack.
- Multi-class classification: Here 7 different attack categories are there.

Following machine learning algorithms is applied and trained model on each type of classification mentioned above are Support vector machine classifier, Decision tree classifier, Naïve Bayes classifier, K-nearest neighbor, K-means clustering, random forest classifier. After model is trained then it is validated on validation dataset and detailed confusion matrix is analyzed performance score accuracy, precision, recall, F1-score.

After analyzing all model's performance, a detailed comparison is done among all models and a best machine learning model is selected for next step in tuning the hyper-parameters to optimize the algorithm.

1. Support vector machine classifier (SVM)

Support Vector machine classifier is applied first with kernel parameter as linear also known as linear support vector machine classifier.

a) On all Labels

On all Label model training time calculated is approx. 408 seconds for SVM classifier while testing time is 0.152 seconds. SVM model showed an accuracy of 94.76%.
Confusion matrix for SVM on all Labels is shown below:

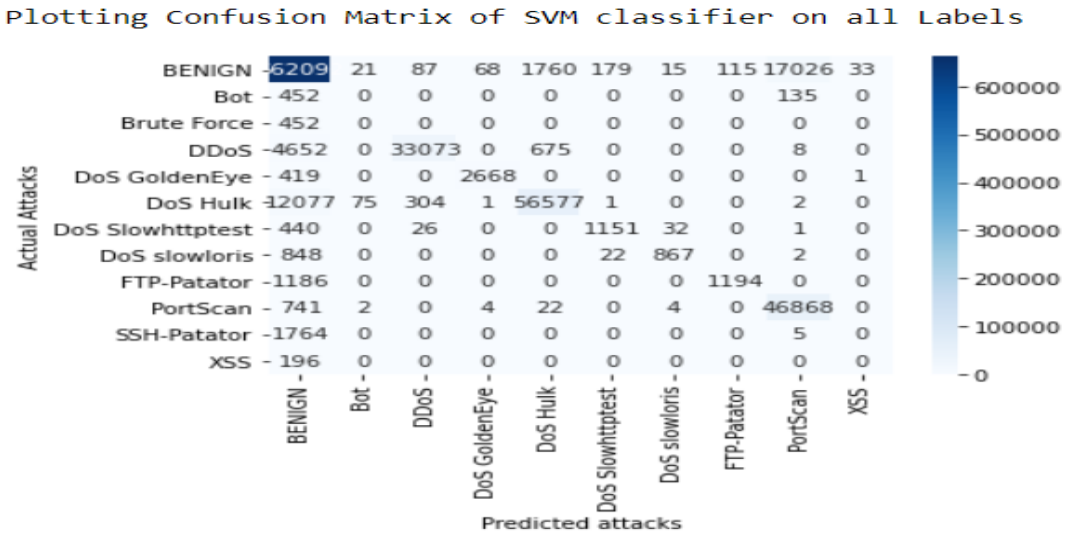

Figure 9 Confusion matrix plot for SVM classifier on all labels

b) Binary Classification

In Binary Classification model training time was 158 seconds and testing time being 0.074 seconds. In this model showed an accuracy of 92.69%.

Confusion matrix for SVM on binary label is shown below:

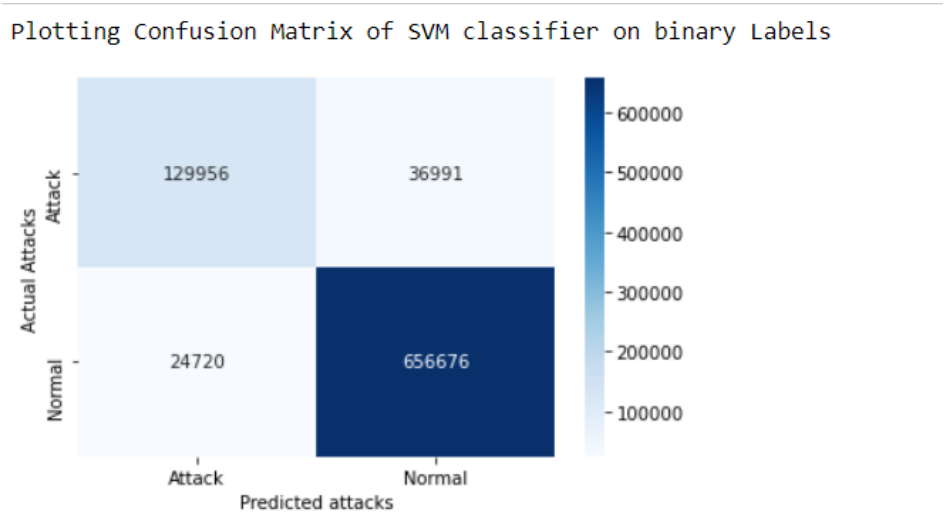

Figure 10 Confusion matrix plot of SVM on Binary classification

c) Multi-class classification

In Multi-class Classification model training time was 352 seconds and testing time being 0.124 seconds. In this model showed an accuracy of 94.34%. Confusion matrix for SVM on multi-class label is shown below:

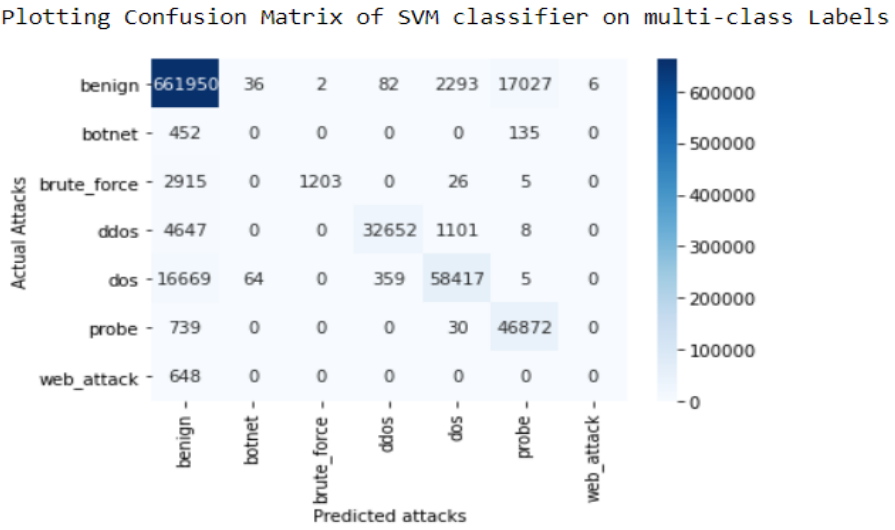

Figure 11 Confusion matrix plot of SVM on multi-class classification

2. Decision Tree

a) On all Labels

On all Label model training time is approx. 112.54 seconds for Decision Tree classifier while testing time is 0.242 seconds. Decision tree model showed an accuracy of 99.96%. Confusion matrix for Decision tree classifier on all Labels is shown below:

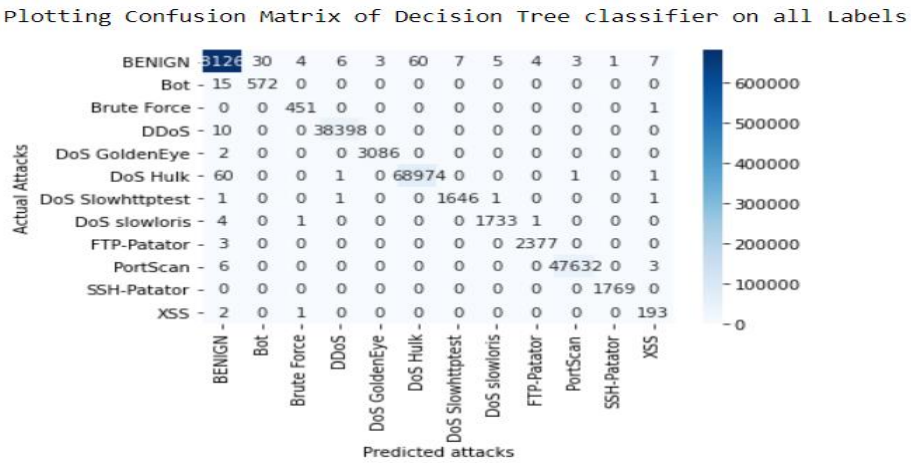

Figure 12 Confusion matrix plot of Decision tree on all labels

b) Binary Classification

In Binary Classification model training time was 103 seconds and testing time being 0.18 seconds. In this model showed an accuracy of 99.96%. Confusion matrix for Decision tree classifier on binary label is shown below:

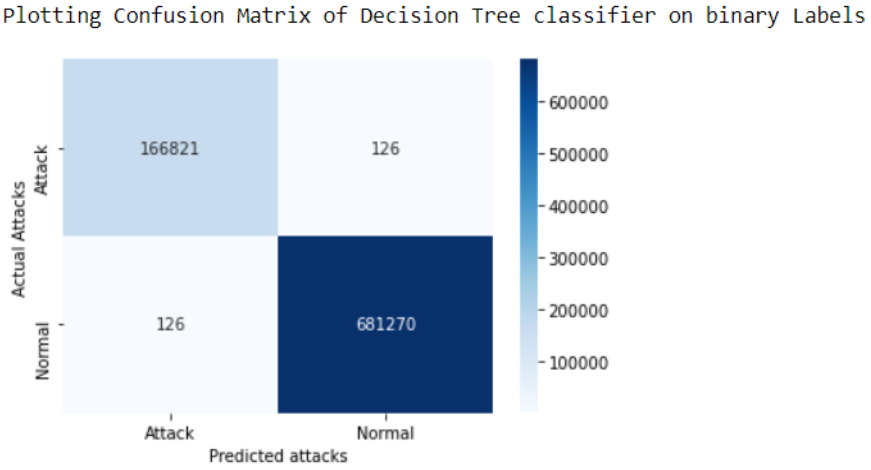

Figure 13 Confusion matrix plot of Decision tree on binary classification

c) Multi-class classification

In Multi-class Classification model training time was 89.94 seconds and testing time being 0.125 seconds. In this model showed an accuracy of 99.96%. Confusion matrix for Decision tree classifier on multi-class label is shown below:

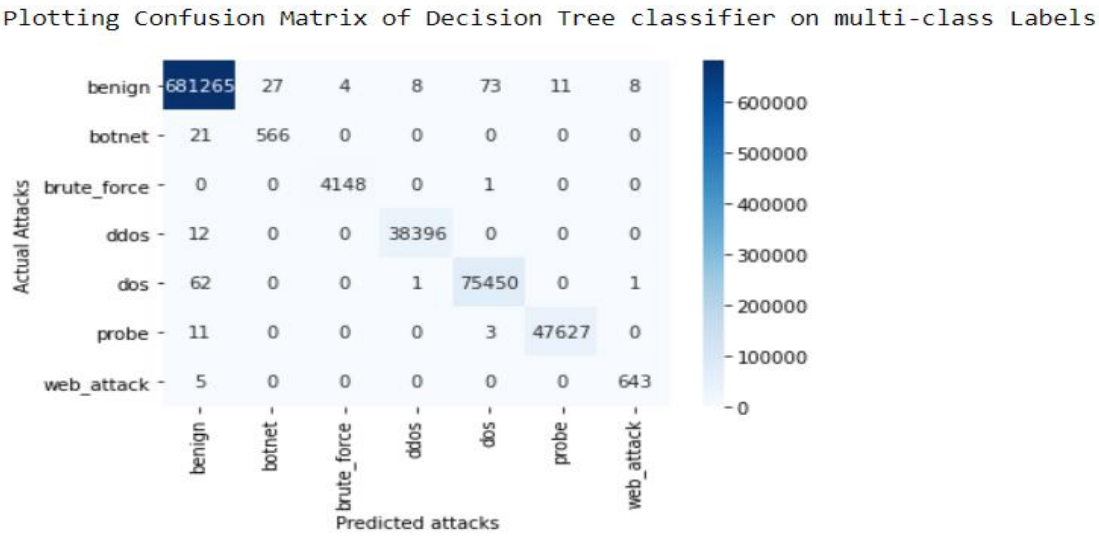

Figure 14 Confusion matrix plot of Decision tree on multi-class classification

3. Naïve Bayes

a) On all Labels

On all Label model training time is approx. 7.247 seconds for Naïve bayes classifier while testing time is 0.115 seconds. Naïve bayes model showed an accuracy of 86.06 %. Confusion matrix for Naïve bayes on all Labels is shown below:

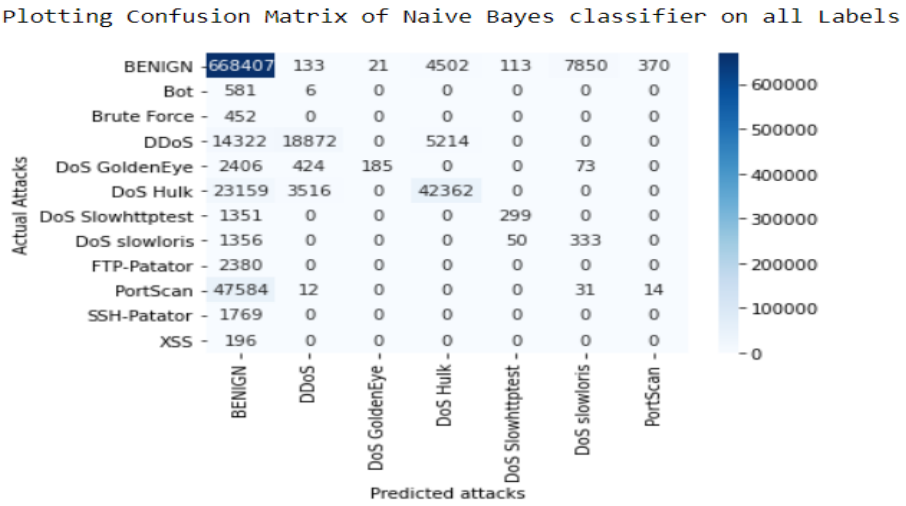

Figure 15 Confusion matrix plot of Naive Bayes on all labels

b) Binary Classification

In Binary Classification model training time was 18.90 seconds and testing time being 0.057 seconds. In this model showed an accuracy of 86.60%. Confusion matrix for Naïve bayes on binary label is shown below:

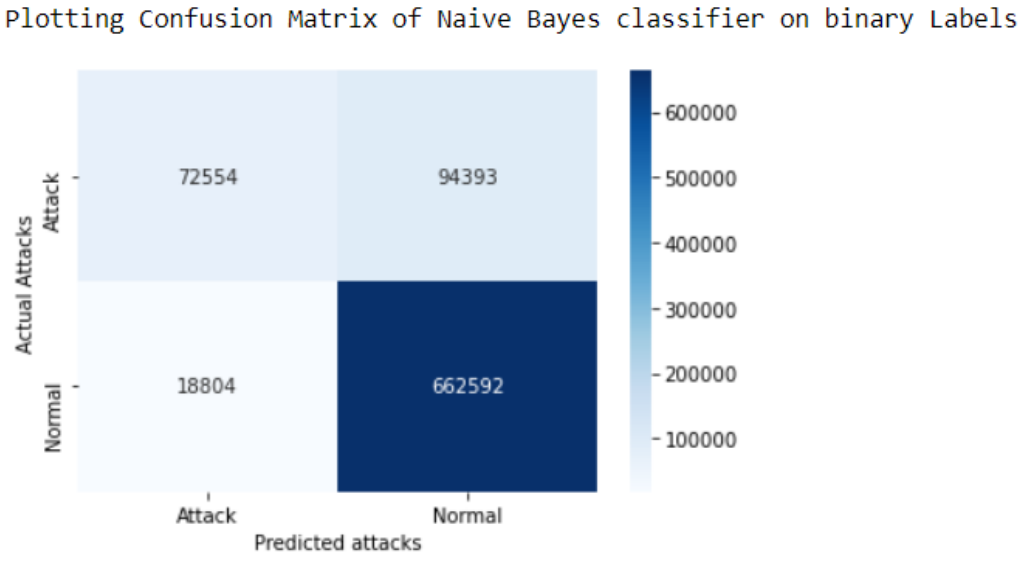

Figure 16 Confusion matrix plot of Naive Bayes on binary classification

c) Multi-class classification

In Multi-class Classification model training time was 4.85 seconds and testing time being 0.073 seconds. In this model showed an accuracy of 85.54%. Confusion matrix for Naïve bayes on multi-class label is shown below:

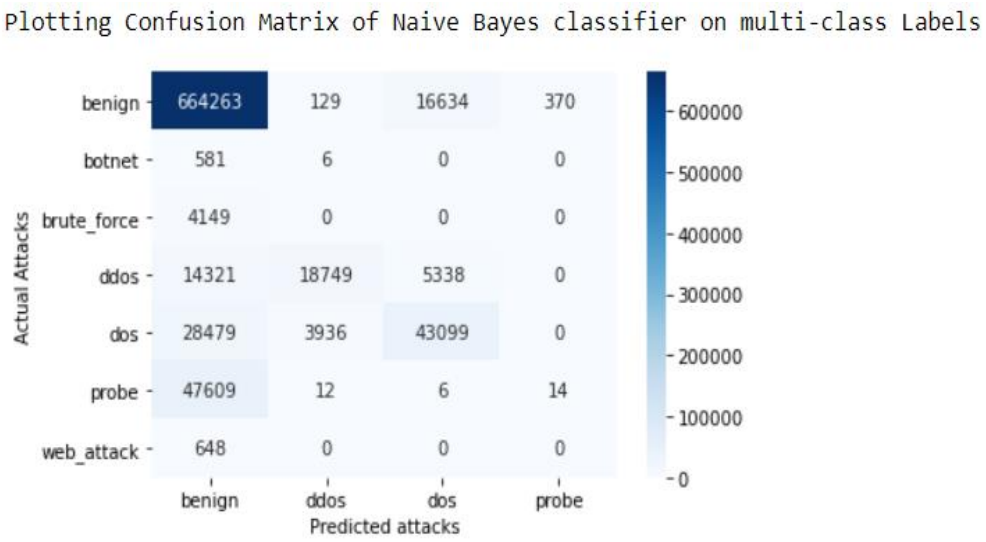

Figure 17 Confusion matrix plot of Naive Bayes on multi-class labels

4. Random Forest Classifier

a) On all Labels

On all Label model training time is approx. 934 seconds for Random Forest classifier while testing time is 9.01 seconds. Random Forest model showed an accuracy of 99.96%. Confusion matrix for Random Forest on all Labels is shown below:

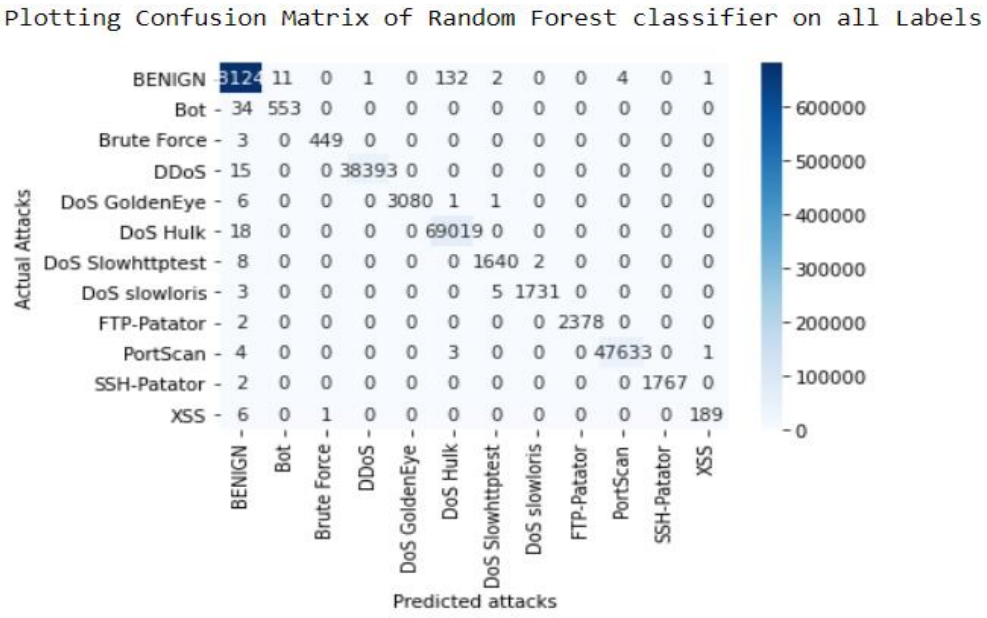

Figure 18 Confusion matrix plot of random Forest on all labels

b) Binary Classification

In Binary Classification model training time was 1027 seconds and testing time being 5.71 seconds. In this model showed an accuracy of 99.97%. Confusion matrix for Random Forest on binary label is shown below:

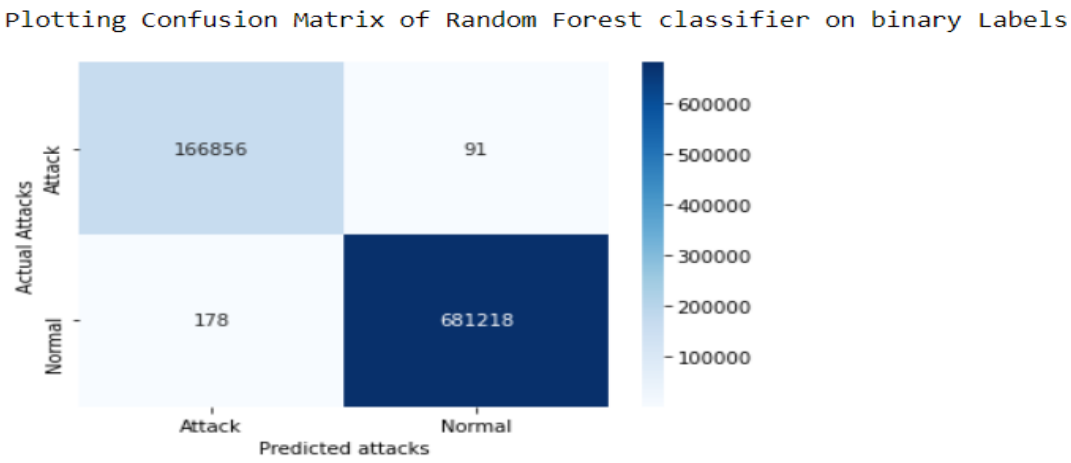

Figure 19 Confusion matrix plot of random Forest on binary labels

c) Multi-class classification

In Multi-class Classification model training time was 767.800 seconds and testing time being 6.95 seconds. In this model showed an accuracy of 99.96%. Confusion matrix for Random Forest on multi-class label is shown below:

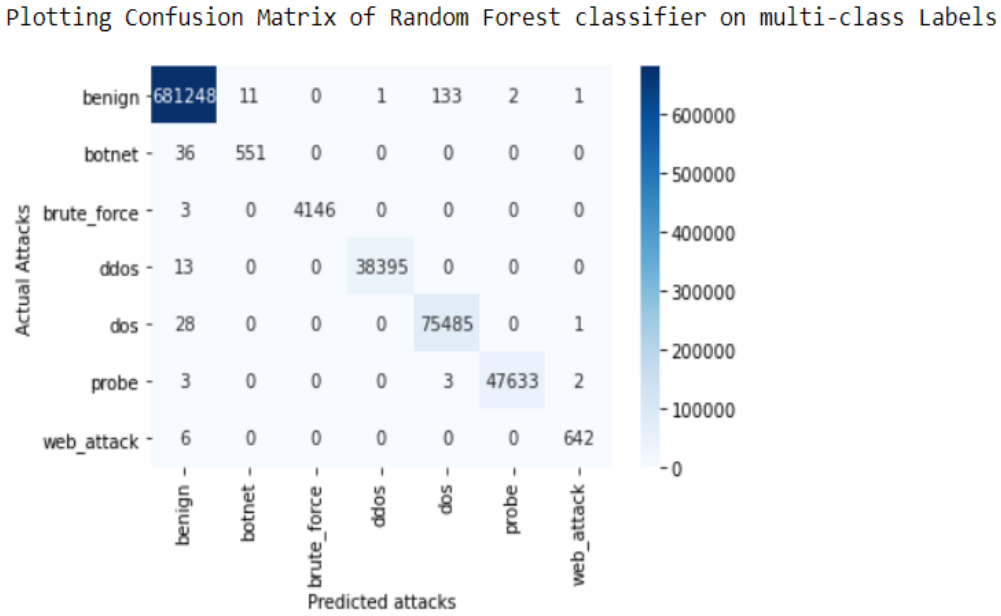

Figure 20 Confusion matrix plot of random Forest on multi-class labels

3.2.5 Comparing Performance scores of all models

Comparison of performance scores of all machine learning models is shown in below Table 4.

Table 4 Performance score comparison of all ML models

	SVM	Decision Tree	Naïve Bayes	Random Forest	
On all labels	94.76% 60.78% 51.26% 54.57%	99.96% 98.70% 99.42% 99.06%	86.06% 34.02% 20.73% 22.39%	99.96% 99.66% 99.02% 99.34%	1. Accuracy 2. Precision 3. Recall 4. F1-score
Binary Classification	92.69% 89.30% 87.04% 88.11%	99.96% 99.94% 99.94% 99.94%	86.60% 83.38% 70.24% 74.03%	99.96% 99.94% 99.96% 99.96%	1. Accuracy 2. Precision 3. Recall 4. F1-score
Multi-class Classification	94.34% 66.00% 55.06% 57.24%	99.96% 98.94% 99.26% 99.10%	85.54% 34.12% 28.99% 30.62%	99.96% 99.74% 99.04% 99.38%	1. Accuracy 2. Precision 3. Recall 4. F1-score

On analyzing all the implemented standard machine learning model it is clearly visible that Random Forest, Decision tree out-performed the classification with very high accuracy and low false alarm rate.

Machine learning models also take a collection of parameters as input, these values are known as hyperparameters and control the behavior of the model (such as the depth of the tree in decision trees and random forests). We can tune these hyperparameters to and the values which yield the optimal results for our problem. For our model, a random forest, listed are the parameters which we will tune, and a description of them according to the scikit-learn library.

To approach towards a more scalable and better classifier method I have optimized the best standard performed model that is random forest classification by tuning the hyper -parameters to get a little improvement in the accuracy.

3.2.6 Optimizing Random Forest Classifier

Random forest algorithm accepts several parameters these are called Hyper-parameters. Hyperparameters is like the settings of an algorithm that can be adjusted to optimize performance, just

as we might turn the knobs of an AM radio to get a clear signal (or your parents might have!). While model parameters are learned during training — such as the slope and intercept in a linear regression — hyperparameters must be set by the data scientist before training [51].

In the case of a random forest, hyperparameters include the number of decision trees in the forest and the number of features considered by each tree when splitting a node. (The parameters of a random forest are the variables and thresholds used to split each node learned during training). Scikit-Learn implements a set of sensible default hyperparameters for all models, but these are not guaranteed to be optimal for a problem.

Random forest algorithm has following hyper-parameters:

- n estimators: The number of trees in the forest.
- max depth: The maximum depth of the tree.
- min samples split: The minimum number of samples required to split a decision node.
- min samples leaf: The minimum number of samples required to be a leaf node.
- max features: The number of features to consider when looking for the best split.
- bootstrap: Whether bootstrap samples are used when building trees. If false, the whole dataset is used to build each tree.

The best hyperparameters are usually impossible to determine ahead of time, and tuning a model is where machine learning turns from a science into trial-and-error based engineering [52].

First, we consider the number of estimators parameter. This controls the number of decision trees that we evaluate in our random forest tress. Figure 22. shows the F1 scores for different values of the n_estimators parameter for the random forest on the validation dataset. The highest peak is when n estimators=800. However, Figure 23. shows why we do not choose this as the value for the number of estimators. Time taken for classification increases linearly with the number of estimators, so setting n estimators=800 would incur a greater time consumption.

Increasing max features increases the model's performance in general since each node now has a larger number of choices to evaluate. However, this is not always true because it reduces the diversity of individual trees, which is the random forest's USP. However, raising the max features will definitely slow down the process. As a result, you must strike the correct balance and select the best max features.

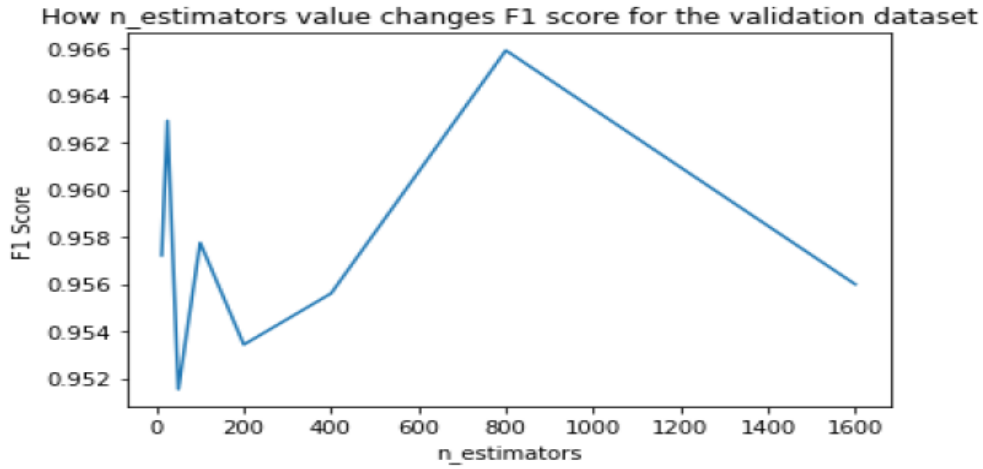

Figure 22 F1-score v/s n_estimators random forest

Although n_estimators=800 is visible was highest, but that many numbers of estimators is not realistic in terms of classification time. We go with the second peak - n_estimators = 25 and reduces the time significantly as shown in Figure 23.

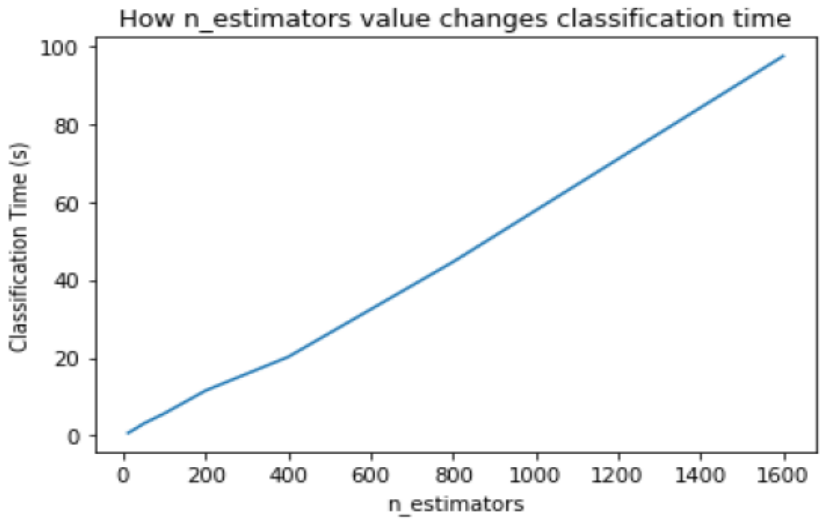

Figure 23 n-estimators v/s classification time for optimizing

To optimize the remaining hyperparameters, we perform a randomized search. This search strategy creates a grid of different values for each hyper-parameter. It then performs training and testing on random combinations of these hyperparameter values, providing each combination with a score. We use Randomized Search CV from the scikit-learn library to implement parameter optimization, and then

call best params to return the parameter setting that gave the best results on the hold out data. This provides us with our final parameter results as shown below:

```
In [54]:
rf_random.best_params_
Out[54]:
{'n_estimators': 25,
 'min_samples_split': 5,
 'min_samples_leaf': 1,
 'max_features': 20,
 'max_depth': 200,
 'bootstrap': True}
```

Now after getting all the Hyper-parameters of Randomized random forest approach the proposed algorithm is applied on all 3 categories of classification and results were compared and studied.

3.2.7 Proposed Optimized Random Forest Classifier

a) On all Labels

On all Label model training time is approx. 462 seconds for optimized random forest classifier while testing time is 0.144 seconds. Proposed model showed an accuracy of 94.83%. Confusion matrix for optimized random forest on all Labels is shown below:

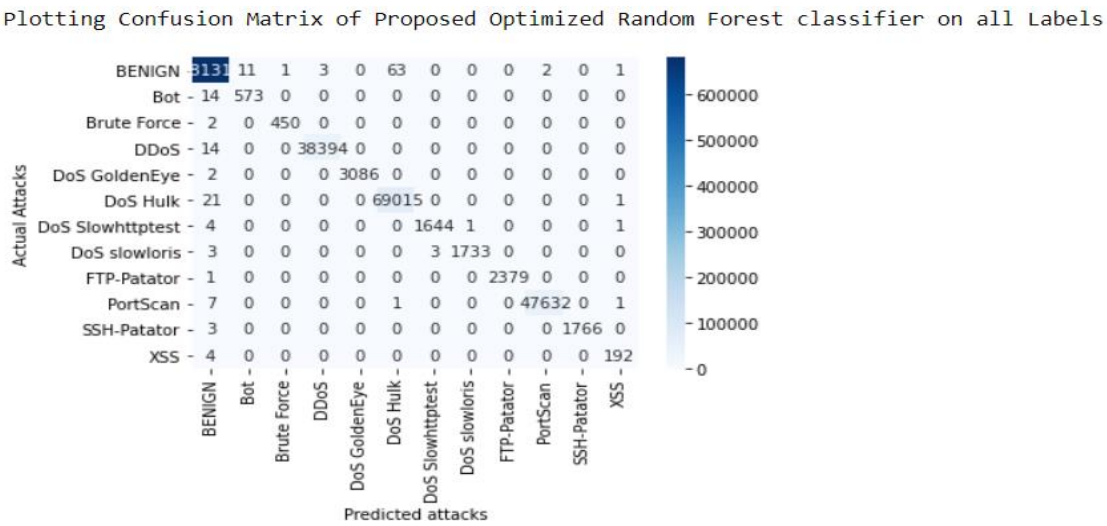

Figure 24 Confusion matrix plot for proposed model on all labels

b) Binary Classification

In Binary Classification model training time was 158 seconds and testing time being 0.72 seconds. In this model showed an accuracy of 92.72%. Confusion matrix for optimized random forest on binary label is shown below:

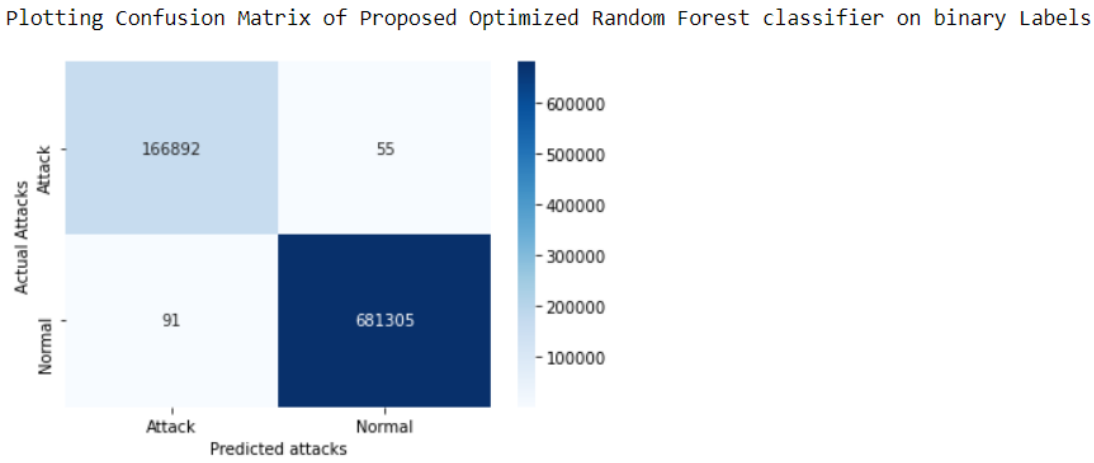

Figure 25 Confusion matrix plot of proposed model on binary labels

c) Multi-class classification

In Multi-class Classification model training time was 340 seconds and testing time being 0.89 seconds. In this model showed an accuracy of 94.43%. Confusion matrix for optimized random forest on multi-class label is shown below:

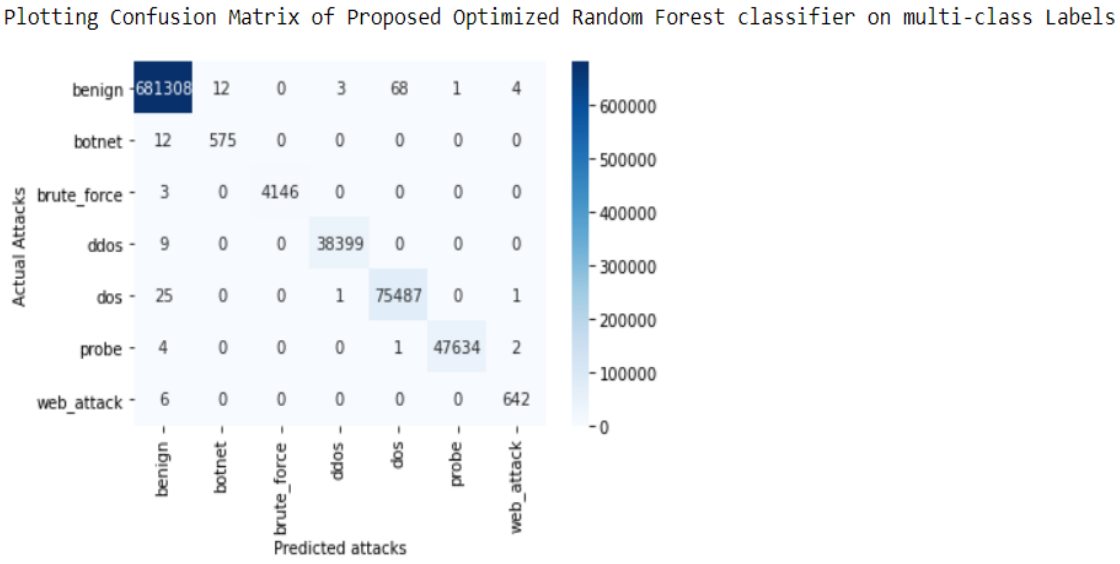

Figure 26 Confusion matrix plot of proposed model on multi-class labels

Analyzing performance score of proposed model:

The proposed randomized random forest algorithm with tuned hyper-parameters has attained performance scores shown in below Table 5.

Table 5 Proposed model Performance scores

Performance scores of Proposed Model (Randomized random Forest)				
	Accuracy	**Precision**	**Recall**	**F1-score**
On all Labels	99.98%	99.62%	99.50%	99.56%
Binary Classifier	99.98%	99.96%	99.97%	99.97%
Multi-class Classifier	99.98%	99.53%	99.55%	99.54%

On analyzing the results of our proposed algorithm, I can successfully state that this model has outperformed all the previous models and achieve a mile stone accuracy in each category of attack for binary as well as multi -class classification.

3.3 Key Contributions

- Understanding of network attacks and its detection, different types of intrusion detection systems, feature optimization techniques, machine learning based classification techniques.

- Implemented machine learning algorithms to detect network attack in CICIDS 2017 dataset and a new randomized random forest feature selection algorithm is proposed. This algorithm is a tuned hyper-parameter of random forest algorithm.

- Comparison of the proposed model with other similar models and is found that the proposed NIDS model has achieved the highest performance for binary as well as multiclass classification.

3.4 Comparison of proposed model and previous studies

A detailed comparison of proposed randomized random forest classifier is done with previous studies in below Table 6.

Table 6 Comparison of proposed model with previous studies

Work	Dataset	Method	Type	Classifier	Accuracy
Hongpo Zhang et al (2020) **[53]**	CICIDS 2017	SGM	Multi-class	RF MLP CNN	93.08% 99.60% 99.85%
Cengiz Colak et al (2017) **[54]**	CICIDS 2017	SMOTE BMCD	Binary Multi-class	GLM Boost Logit Boost RF MLP NB	82.47% 96.95% 99.32% 94.82% 75.35%
S. U. Jan, et al **[55]**	CICIDS 2017	Wrapper based feature selection technique	Multi-class	SVM	98.03
Mohammad Aljanabi et al **[56]**	CICIDS 2017	Improved TLBO, improved JAYA	Multi-class	SVM	98.17
Hassan Musafer et al **[57]**	CICIDS 2017	Auto encoder	Multi-class	Random forest	99.50%
Proposed Model	**CICIDS 2017**	**Chi-square**	**Binary** **Multiclass**	**Randomized random forest**	**99.98%** **99.98%**

From above Table 6 showing comparison among my proposed work and other previous studies showing again that the proposed model in this study achieved a milestone performance.

3.5 Summary

In this chapter, proposed work and its implementation is discuused in detail. Proposed work is impplemented on CICIDS 2017 dataset shown in flow chart above. I evaluated several machine learning methods in this chapter. As we worked toward building our final model, the random forest classifier, which we utilise in our IDS, The learning-based IDS was then tested on its capacity to detect a set of created network threats. In this work dataset is preprocesssed and performed multiple classification algorithms including supervised and unsupervised methods of classification in machine learning. On anlaysing dataset it is found that the dataset has 79 attributes but after some pre-processing left with only 69 attributes then feature selection algorithm is applied to reeduce feature set so that processing time may be reduced and performance may increased. In this work chi-square feature selection method is proposed and here I selected the best 40 features as shown in plot of cumulative feature scores. After feature selection machine leraning algorithm is studied on multiple classification criterions so that performance is anlayze inn efiicient way for all model a best performed nmodel is then selected. Now main focussed wotrk of this study starts here which analyzes the hyper-parameters of selected model which is then tuned to increase the performance of model. And the final results werw analyzed and it is observed that model has increassed its performance and achieve a milestone accuracy. However by analyzing confusion matrix for multiclass classification Botnet category of attack is not upto the mark level of performance as compared to other attacks this may be some future directions of work for this project. However, the proposed model has overall optimal perormance and a new method for detecting network attacks using optized machine learning approaches can be a way towards new models of impleementing ntwork intrusion detection syetms.

Chapter 4: Conclusion and Future Works

4.1 Conclusions

It's critical to develop powerful anomalous intrusion detection systems since they're the first line of defending against new threats. An IDS model is suggested in this work using the tuned random forest algorithm. The models were tested and assessed using the CICIDS2017 dataset, and the findings were encouraging. Proposed randomized random forest algorithm framed by tuning hyper-parameters of random forest classifier achieved an improvement in accuracy of classifying and detecting the network attack in a network system. Successfully implemented the standard machine learning algorithms is 3 categories that is on all labels, binary classification, multi-class classification for support vector machine, decision tree, naïve bayes, random forest. All these models are evaluated on performance and the best preformed model I got was random forest is then tune by varying hyperparameters using randomized search method. In this work a new method is framed to detect network attacks and successfully experimented on CICIDS 2017 data set with achieved accuracy 99.98% which in itself is a milestone. I gained a better grasp of machine learning algorithms and the processes necessary to apply them while working on this project. The research gave me the opportunity to investigate various sorts of network threats and learn how to carry them out. I learned Python and the libraries that go with it (Numpy, Pandas, Scikit-Learn and Scapy). After all was said and done, I learned important abilities in software development, independent research, and time management.

4.2 Future works

This work illustrated the importance of using an optimal subset of features with a suitable classification algorithm for designing an intrusion detection system. On evaluation, the advantages of the proposed method i.e. randomized random forest algorithm for classification with proposed feature extraction method to reducing the features to best contributors improves the model performance by eliminating redundant features. On analyzing results, I could see there might be some improvement still possible as Botnet category of attack is lacking to the optimum performance. Botnet is the weakest of the attacks, with some of its labelling classified as benign. The cause for this misclassification, we believe, is related to the botnet's many operating stages. Botnets connect with a command on server, which is capable of executing a wide range of instructions. As a result, it's probable that our algorithm has learned to recognize botnet when malicious network traffic is generated as a result of botnet activity (such as being used in a DDoS attack). However, bots can remain idle, or the adversary might reduce the intensity of the assaults to pass as "regular" traffic, which is most likely why our algorithm classifies botnet traffic as benign.

Some possible ways for future directions of work to achieve more success could be:

- **Deep Learning:** Neural networks are used in deep learning, which is a form of machine learning. The computer learns to execute a task, in this case, identifying threats, using multiple neural network-based techniques that yielded promising results.

- **Feature Selection:** I believe that the reason that detecting XSS attacks did not attain the same level of performance as other attacks was due to the features used. As a result, future study might combine application-level data to produce more relevant characteristics in the hopes of improving botnet and online attack performance.

References

[1] G. C. S. J. M. D. Yuyang Zhou, "Building an Efficient Intrusion Detection System Based on Feature Selection and Ensemble Classifier," *Computer Networks,* vol. 174, no. 1389-1286, p. 107247, 2020.

[2] H.-J. a. L. C.-H. a. L. Y.-C. a. T. K.-Y. Liao, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications,* vol. 36, pp. 16-24, 2013.

[3] Ahmad, Z, Shahid Khan, A, Wai Shiang, C, Abdullah, J, Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerging Tel Tech. 2021; 32:e4150. https://doi.org/10.1002/ett.4150

[4] Kumar I., Mohd N., Bhatt C., Sharma S.K. (2020) Development of IDS Using Supervised Machine Learning. In: Pant M., Kumar Sharma T., Arya R., Sahana B., Zolfagharinia H. (eds) Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, vol 1154. Springer, Singapore. https://doi.org/10.1007/978-981-15-4032-5_52

[5] A. H. L. a. A. A. G. Iman Sharafaldin, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, Jan 2018.

[6] "CICIDS 2017 Dataset," [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html.

[7] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2, 20 (2019). https://doi.org/10.1186/s42400-019-0038

[8] Othman, Suad & Nabeel, Taha & Ba-Alwi, Fadl & Zahary, Ammar. (2018). Survey on Intrusion Detection System Types. 7. 444-462.

[9] Kai Peng, Victor C. M. Leung, Lixin Zheng, Shangguang Wang, Chao Huang, Tao Lin, "Intrusion Detection System Based on Decision Tree over Big Data in Fog Environment", Wireless Communications and Mobile Computing, vol. 2018, Article ID 4680867, 10 pages, 2018. https://doi.org/10.1155/2018/4680867

[10] Poria Pirozmand, Mohsen Angoraj Ghafary, Safieh Siadat, Jiankang Ren, "Intrusion Detection into Cloud-Fog-Based IoT Networks Using Game Theory", Wireless Communications and Mobile Computing, vol. 2020, Article ID 8819545, 9 pages, 2020. https://doi.org/10.1155/2020/8819545

[11] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set,” Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[12] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[13] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Idris, A. M. Bamhdi and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis With Information Gain for Anomaly Detection," in IEEE Access, vol. 8, pp. 132911-132921, 2020, doi: 10.1109/ACCESS.2020.3009843.

[14] A. Borkar, A. Donode and A. Kumari, A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS)," 2017 Inter-national Conference on Inventive Computing and Informatics (ICICI), Coimbatore,2017, pp. 949-953.

[15] Das, Sumit, Aritra Dey, Akash Pal and Nabamita Roy. \Applications of Articial Intelligence in Machine Learning: Review and Prospect." 2015 International Journal of Computer Applications 115, 2015, pp. 31-41.

[16] Mohamed A.B., Idris N.B., Shanmugum B. (2012) A Brief Introduction to Intrusion Detection System. In: Ponnambalam S.G., Parkkinen J., Ramanathan K.C. (eds) Trends in Intelligent Robotics, Automation, and Manufacturing. IRAM 2012. Communications in Computer and Information Science, vol 330. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35197-6_29

[17] Tiwari, Mohit & Kumar, Raj & Bharti, Akash & Kishan, Jai. (2017). INTRUSION DETECTION SYSTEM. International Journal of Technical Research and Applications. 5. 2320-8163.

[18] Othman, Suad & Alsohybe, Nabeel & Ba-Alwi, Fadl & Zahary, Ammar. (2018). Survey on Intrusion Detection System Types. 7. 444-462.

[19] Yu, Kun-Ming & Wu, Ming-Feng & Wong, Wai-Tak. (2008). Protocol-based classification for intrusion detection. 3.

[20] K. V. Rajkumar, V. Vaidehi, S. Pradeep, N. Srinivasan, and M. Vanishree, "Application Level IDS using Protocol Analysis," 2007 International Conference on Signal Processing, Communications and Networking, 2007, pp. 355-359, DOI: 10.1109/ICSCN.2007.350762.

[21] Garg, Akash & Maheshwari, Prachi. (2016). A hybrid intrusion detection system: A review. 1-5. 10.1109/ISCO.2016.7726909.

[22] A. H. Almutairi and N. T. Abdelmajeed, "Innovative signature based intrusion detection system: Parallel processing and minimized database," 2017 International Conference on the Frontiers and Advances in Data Science (FADS), 2017, pp. 114-119, doi: 10.1109/FADS.2017.8253208.

[23] Khraisat A, Gondal I, Vamplew P (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp 149–155

[24] Watt, Jeremy, Reza Borhani, and Aggelos K. Katsaggelos.Introduction to Machine Learning." Chapter. In Machine Learning Rened: Foundations, Algorithms, and Applications, 2nd ed., pp 1{18. Cambridge: Cambridge University Press, 2020.doi:10.1017/9781108690935.003.

[25] Girish Chandrashekar, Ferat Sahin, A Survey on Feature Selection Methods," Computers Electrical Engineering, Volume 40, Issue 1, 2014, pp 16-28, ISSN 0045-7906.

[26] Rafael G. Mantovani, Andre L. D. Rossi, Joaquin Vanschoren, Bernd Bischl and Andre C. P. L. F., 2015. Effectiveness of Random Search in SVM hyper-parameter Tuning. IEEE Proceedings of the 2015 International Joint Conference on Neural Networks, July 2015.

[27] Ahlam Alrehili and Kholood Albalawi, 2019. Sentiment Analysis of Customer Reviews using Ensemble Method, International Conference on Computer and Information Sciences (ICCIS), IEEE.

[28] Arif Yulianto, Parman Sukarno and Novian Anggis Suwastika, (2019), Improving AdaBoost-based Intrusion Detection System (IDS) Performance on CIC IDS 2017 Dataset. In Journal of Physics: Conference Series (Vol. 1192, No. 1, p. 012018). IOP Publishing

[29] Razan Abdulhammed, Hassan Musafer, Ali Alessa, Miad Faezipour and Abdelshakour Abuzneid., (2019, Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. Electronics 8(3):322; DOI: 10.3390/electronics8030322.

[30] Khraisat, A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets, and challenges. Cybersecurity 2, 20 (2019). https://doi.org/10.1186/s42400-019-00387

[31] Welikala, R.A, Fraz, MM, Dehmeshki, J, Hoppe, A, Tah, V, Mann, S, Williamson, TH & Barman, SA (2015) Genetic Algorithm Based Feature Selection combined with dual classification for the Automated Detection of proliferative Diabetic Retinopathy", Computerized Medical Imaging and Graphics, vol. 43, pp.64-77

[32] Wenjuan Lian, Guoqing Nie, Bin Jia, Dandan Shi, Qi Fan, and Yongquan Liang "An Intrusion Detection Method Based on Decision Tree-Recursive Feature Elimination in Ensemble Learning" https://doi.org/10.1155/2020/2835023

[33] A. Alazab, M. Hobbs, J. Abawajy and M. Alazab, "Using feature selection for intrusion detection system," 2012 International Symposium on Communications and Information Technologies (ISCIT), 2012, pp. 296-301, DOI: 10.1109/ISCIT.2012.6380910.

[34] B. Abolhasanzadeh, "Nonlinear dimensionality reduction for intrusion detection using auto-encoder bottleneck features," 2015 7th Conference on Information and Knowledge Technology (IKT), 2015, pp. 1-5, DOI: 10.1109/IKT.2015.7288799.

[35] S. Omar, H. H. Jebur, and S. Benqdara, "An adaptive intrusion detection model based on machine learning techniques," International Journal of Computer Applications, vol. 70, no. 7, pp. 1–5, 2017.

[36] Cao, Xi Hang et al. "A robust data scaling algorithm to improve classification accuracies in biomedical data." BMC bioinformatics vol. 17,1 359. 9 Sep. 2016, doi:10.1186/s12859-016-1236-x

[37] Wang, Z., Lin, Z. Optimal Feature Selection for Learning-Based Algorithms for Sentiment Classification. Cogn Comput 12, 238–248 (2020). https://doi.org/10.1007/s12559-019-09669-5

[38] (2008) Pearson's Correlation Coefficient. In: Kirch W. (eds) Encyclopedia of Public Health. Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-5614-7_2569

[39] Ostertagova, Eva & Ostertag, Oskar. (2013). Methodology and Application of One-way ANOVA. American Journal of Mechanical Engineering. 1. 256-261. 10.12691/ajme-1-7-21.

[40] Tian, Yingjie & Shi, Yong & Liu, Xiaohui. (2012). Recent advances on support vector machines research. Technological and Economic Development of Economy. 18. 10.3846/20294913.2012.661205.

[41] Patel, Harsh & Prajapati, Purvi. (2018). Study and Analysis of Decision Tree Based Classification Algorithms. International Journal of Computer Sciences and Engineering. 6. 74-78. 10.26438/ijcse/v6i10.7478.

[42] Kaviani, Pouria & Dhotre, Sunita. (2017). Short Survey on Naive Bayes Algorithm. International Journal of Advance Research in Computer Science and Management. 04.

[43] K. Taunk, S. De, S. Verma and A. Swetapadma, "A Brief Review of Nearest Neighbor Algorithm for Learning and Classification," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 1255-1260, doi: 10.1109/ICCS45141.2019.9065747.

[44] K. P. Sinaga and M. Yang, "Unsupervised K-Means Clustering Algorithm," in IEEE Access, vol. 8, pp. 80716-80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[45] A. Tesfahun and D. L. Bhaskari, "Intrusion Detection Using Random Forests Classifier with SMOTE and Feature Reduction," 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, 2013, pp. 127-132, doi: 10.1109/CUBE.2013.31.

[46] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. 2, 160 (2021). https://doi.org/10.1007/s42979-021-00592-x

[47] Kurniabudi, Kurniabudi & Stiawan, Deris & Dr, Darmawijoyo & Idris, Mohd & Bamhdi, Alwi & Budiarto, Rahmat. (2020). CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.3009843.

[48] CICIDS dataset description. https://www.unb.ca/cic/datasets/ids-2017.html

[49] Y. Zhai, W. Song, X. Liu, L. Liu and X. Zhao, "A Chi-Square Statistics Based Feature Selection Method in Text Classification," 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), 2018, pp. 160-163, doi: 10.1109/ICSESS.2018.8663882.

[50] Thaseen, Sumaiya & Cherukuri, Aswani Kumar. (2016). Intrusion Detection Model Using Chi Square Feature Selection and Modified Naïve Bayes Classifier. 10.1007/978-3-319-30348-2_7.

[51] Probst, P, Wright, MN, Boulesteix, A-L. Hyperparameters and tuning strategies for random forest. WIREs Data Mining Knowl Discov. 2019; 9:e1301. https://doi.org/10.1002/widm.1301

[52] Probst, P., Wright, M.N., & Boulesteix, A. (2019). Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9.

[53] H. Zhang, C. Q. Wu, S. Gao, Z. Wang, Y. Xu and Y. Liu, "An Effective Deep Learning Based Scheme for Network Intrusion Detection," 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 682-687, doi: 10.1109/ICPR.2018.8546162.

[54] Colak, M. & Karaaslan, Erol & Colak, C. & ARSLAN, Ahmet & Erdil, Nevzat. (2017). Handling imbalanced class problem for the prediction of atrial fibrillation in obese patient. Biomedical Research (India). 28. 3293-3299.

[55] Jan, Sana & Ahmed, Saeed & Shakhov, Vladimir & Koo, Insoo. (2019). Toward a Lightweight Intrusion Detection System for the Internet of Things. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2907965.

[56] Aljanabi, Mohammad & Ismail, Mohd Arfian. (2021). Improved Intrusion Detection Algorithm based on TLBO and GA Algorithms. International Arab Journal of Information Technology. 18. 170-179. 10.34028/iajit/18/2/5.

[57] Musafer, Hassan & Abuzneid, Abdelshakour & Faezipour, Miad & Mahmood, Ausif. (2020). An Enhanced Design of Sparse Autoencoder for Latent Features Extraction Based on Trigonometric Simplexes for Network Intrusion Detection Systems. Electronics. 9. 259. 10.3390/electronics9020259.