A Novel Two Layer Intrusion Detection System: A Machine Learning Approach Based on Random Forest and Support Vector Machine

Sabrina Afroz
ID: 2015–1–60–160

Samin Nawer Rafa
ID: 2016–2–60–019

S.M. Ariful Islam
ID: 2016–2–60–059

Supervisor
Dr. Maheen Islam
Associate Professor
Department of Computer Science and Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

Department of Computer Science and Engineering
East West University
Dhaka-1212, Bangladesh

October, 2020

ABSTRACT

Unauthorized access or intrusion is a massive threatening issue in the modern era. This study focuses on designing a model for an ideal intrusion detection system capable of defending a network by alerting the admins upon detecting any sorts of malicious activities. The study proposes a two layered anomaly-based detection model that uses filter co-relation method for dimensionality reduction along with Random forest and Support Vector Machine as its classifiers. It achieved a very good detection rate against all sorts of attacks including a low rate of false alarms as well. The contribution of this study is that it could be of a major help to the computer scientists designing good intrusion detection systems to keep an industry or organization safe from the cyber threats as it has achieved the desired qualities of a functional IDS model.

DECLARATION

We, hereby, declare that the work presented in this thesis is the outcome of the investigation performed by us under the supervision of **Dr. Maheen Islam**, Associate Professor, Department of Computer Science and Engineering, East West University. We also declare that no part of this thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

....

(**Dr. Maheen Islam**)
Supervisor

Signature

....

(Sabrina Afroz)
ID: 2015–1–60–160

Signature

....

(Samin Nawer Rafa)
ID: 2016–2–60–019

Signature

....

(S.M. Ariful Islam)
ID: 2016–2–60–059

LETTER OF ACCEPTANCE

We, hereby, declare the thesis is from the student's own work and best effort of us and all other sources of information used in this paper have been acknowledged. This thesis has been submitted with our approval.

Supervisor
Signature

....
(Dr. Maheen Islam)
Associate Professor
Department of Computer Science and Engineering
East West University, Dhaka

Chairperson
Signature

....
(Dr. Taskeed Jabid)
Associate Professor
Department of Computer Science and Engineering
East West University, Dhaka

ACKNOWLEDGMENTS

As it is true for everyone, we have also arrived at this point of achieving a goal in our life through various interactions with and help from other people. However, written words are often elusive and harbor diverse interpretations even in one's mother language. Therefore, we would not like to make efforts to find best words to express our thankfulness other than simply listing those people who have contributed to this thesis itself in an essential way. This work was carried out in the Department of Computer Science and Engineering at East West University, Bangladesh.

First of all, we would like to express our deepest gratitude to the almighty Allah for the blessings on us. Next, our special thanks go to our supervisor, **Dr. Maheen Islam**, who gave us this opportunity, initiated us into the field of "**A Novel Two Step Intrusion Detection System Based on Random Forest and Support Vector Machine.**", and without whom this work would not have been possible. His encouragements, visionaries and thoughtful comments and suggestions, unforgettable support at every stage of our B.Sc. study were simply appreciating and essential. His ability to help us enough to finally answer our own question correctly is something valuable what we have learned and we would try to emulate, if ever we get the opportunity.

There are numerous other people too who have shown me their constant support and friendship in various ways, directly or indirectly related to our academic life. We will remember them in our heart and hope to find a more appropriate place to acknowledge them in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

The Internet is one of the most influential innovations in recent history. Despite the most people using the Internet for productive activities, some people use it as an opportunity for malicious intent. As the Internet links more users together and computers are more prevalent in our daily lives, the Internet and the computers connected to it increasingly become more enticing targets of attacks [6].

Intrusion refers to the unwanted interruption or unauthorized access in a network or a system. It is the process of entering a computer system by manipulating the security systems. Intrusion can lead to networking flooding, information theft or even hacking. That is where the significance of an intrusion detection is considered.

An intrusion detection system (IDS) is basically a device or a software application that inspects a network for malicious activity or violations of policy. Any malicious activity or violation is reported to the administrator of the system using a security information and event management system. Some IDS's are capable of responding immediately to detected intrusion after discovery. To prevent a system from being harmed, IDS has become an effective research area since it was first introduced in 1980s [1]. The detector eradicates redundant information from the audit data and then makes a decision to assess the probability that these activities can be considered as a sign of an intrusion [2]. Furthermore, it is hard to keep intrusion detection signature sets up to date as the number of continuously discovered vulnerabilities keep increasing [3]. That is why it is really important to establish an IDS that is capable of detecting almost all types of anomalies.

Network based intrusion detection systems are generally rule-based or anomaly based. But the anomaly-based approach captures network traffic and then detects patterns that are different from normal behavior which could be defined as anomaly activities and alerts the user about these abnormal activities. This approach is superior because it is able to work against novel and unseen malicious activities [4].

1.1.1 General Approaches of IDS

There are two approaches for intrusion detection system:

- Signature Detection and
- Anomaly Detection.

Signature detection (also known as misuse detection) searches for patterns signaling well-known attacks are searches. But anomaly detection searches for patterns thar are different from normal behavior. Signature detection works efficiently on known attacks. Its disadvantage is that it is unable to detect newer attacks. Though anomaly detection is capable of detecting novel attacks, it signals some unusual events which are not actually threatening. As a result, sometimes it could generate false alarms [4]. A desirable system should follow both of these approaches.

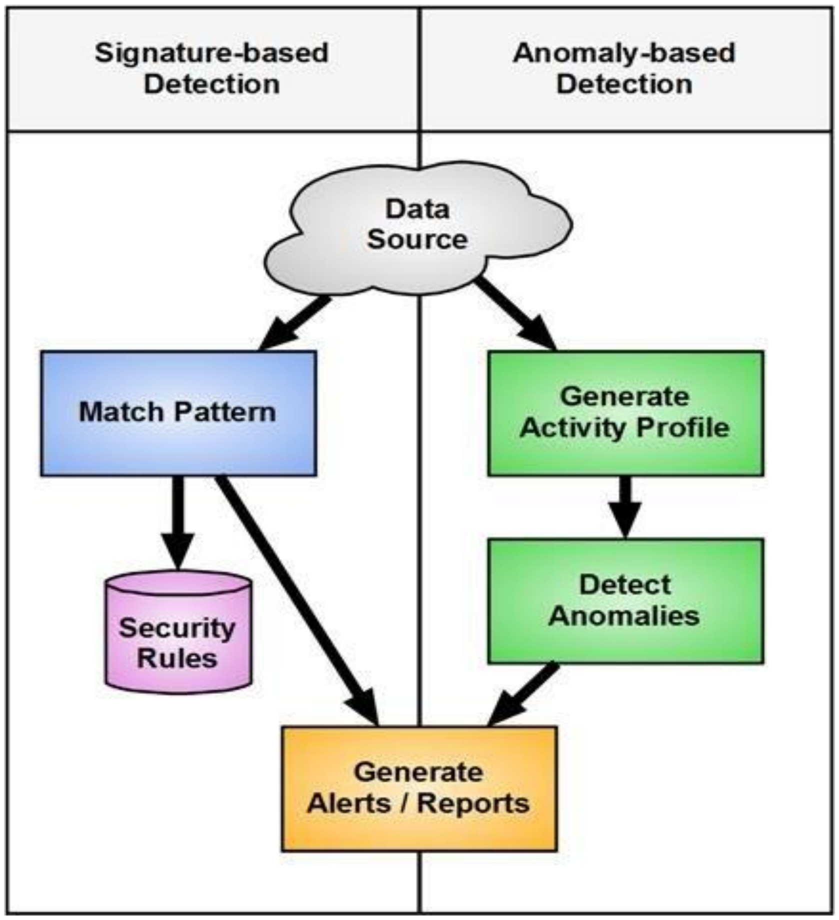


Figure 1. 1 Signature and Anomaly Based IDS

1.1.2 Anomaly

Anomaly-based approach can build a model based on the behavior of normal systems after capturing network traffic and works to detect patterns that differentiate from normal behavior which is called anomaly activities. It alerts the user from these unusual activities. Main benefit of this approach is its functionality against novel and unseen malicious activities. There are two different categories for anomaly detection [15]:

- Supervised Anomaly Detection: In the supervised one, the normal behavior model of networks is established by training with a labeled dataset. The models classify new network connection and distinguish anomaly behaviors from normal ones.
- Unsupervised Anomaly Detection: In the unsupervised anomaly detection, it approaches work without any labeled training data and most of them detect anomalous activities by clustering or outliers-detections techniques.

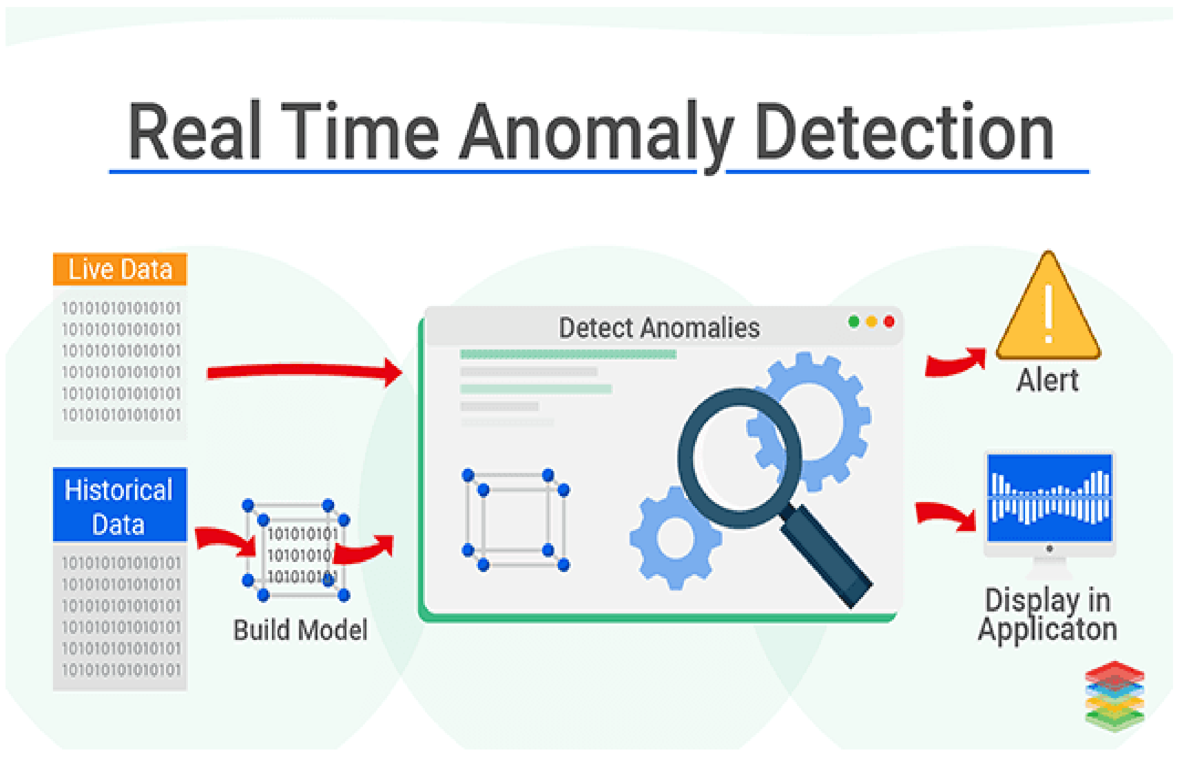


Figure 1. 2 Synchronization between beacon node and sensor node

1.1.3 Network Based System Categories

Network based systems are divided into four categories:

DoS: Denial of Service (DoS) is an attack which can shut down a machine or network, making it impossible to access to its intended users. DoS attacks accomplish this by flooding the target's network with traffic, or sending such information that triggers it to crash. In both case scenarios, the DoS attack deprives authorized users of the service or resource they expected [4].

DoS attackers often target web servers of high-profile organizations such as banking, commerce, media companies, government and trade organizations. DoS does not necessarily result in data theft or loss of important information but it can harm the victims by a great extend as it can cost a great deal of time and expense to the victims.

Probe: A probe is an attack which is considerably crafted so that its target detects and reports it with a recognizable "fingerprint" in the report. The attacker then uses the collaborative infrastructure to learn the detector's location and defensive capabilities from this report [16]. In another word, probe attacker tries to obtain information about its target host which is similar to ports scanning [4],

U2R: Full form of U2R is User to Root attack. In a U2R attack, the attacker first succeeds in gaining control on the remote system by disguising as a user session, ideally in the form of an interactive shell or TELNET window. Combining numerous traditional techniques, the attacker attempts to incrementally gain his privileges until he achieves super-user permissions [8]. An example of a privilege gaining technique is to use stack-smashing, which feeds a packet to a set-UID-to-root program corrupting its address space so that a return-from-subroutine instruction results in the beginning of a set-UID-to-root command shell [1]. A huge amount of archaic mechanisms prevails in both UNIX and Windows. This is what enables attackers rapidly to gain their privileges. For example, in an obsolete implementation of local e-mail, an attacker is able to link passwords to root's personal mailbox. He then can use the command mail root to construct and install a fake login with super-user permissions [19]. This attacker takes advantage of the fact that e-mail was not built according to the least privilege principle [16].

R2L: R2L refers to Remote to Local. R2L is an attack type where attackers try to get access remotely to victim machine following brute force password guessing method [4]. A Remote to Local attack also known as a Remote to User attack) is conceptually similar to the user-to-root attack. The difference is that it is more modest in its ultimate ambition. Such an attack is conducted when an attacker sends packets to the target host intending to disclose vulnerabilities. This enables the attacker to make use of a local user's privileges. Such vulnerabilities would ideally be found in such utilities as xlock, guest, xnsnoop, phf, and Sendmail [7]. Acquiring a foothold as an authorized user can help as an important overture to conduct a subsequent user-to-root attack.

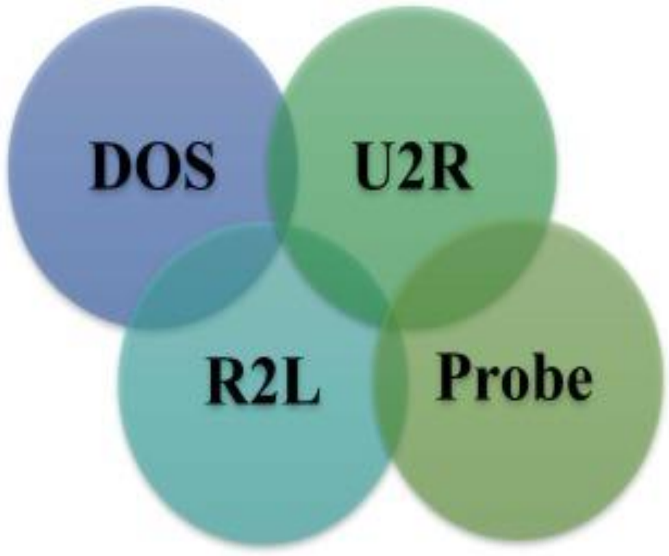


Figure 1. 3 Names of the different attack types

1.1.4 Supervised Learning

It is called supervised learning because the process of supervised learning (from the training dataset) can be considered as a teacher supervising the entire learning process. Thus, the "learning algorithm" iteratively makes predictions on the training data and is corrected by the "teacher". The learning comes to an end when the algorithm achieves an acceptable level of performance or the desired accuracy.

The basic aim of supervised learning is to calculate the mapping function (mentioned above) greatly so that when there is a new input data, the corresponding output variable could be predicted.

The previous work for supervised learning is called as training data in Data Mining terminology. It gets to learn things from the training data. It has a response variable which decided what an object is based on its features. From the training data, it learns about the features of an object. Then matches the features to the object and gives a conclusion whether the thing that has been shown is that specific object or not

This type of information is decoded from the data used to train the model. This type of learning is what refers to as Supervised Learning. Such problems are listed under classical Classification Tasks.

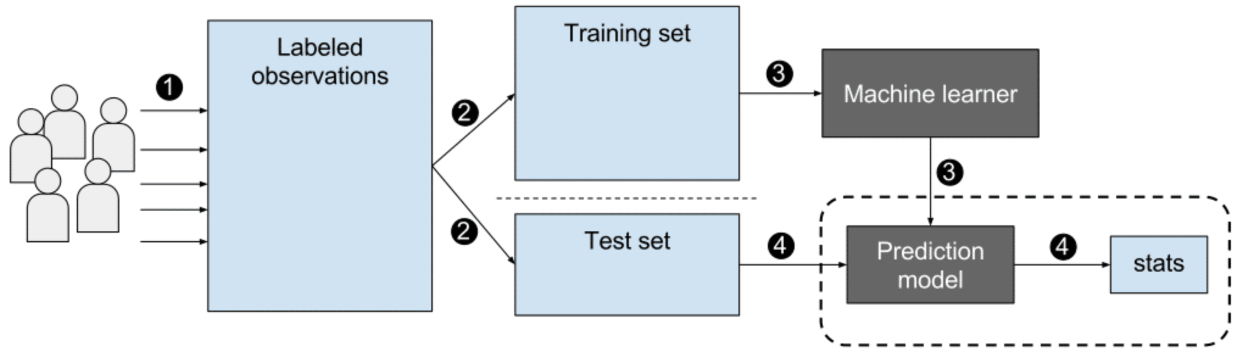


Figure 1. 4 Supervised Learning

1.1.5 Unsupervised Learning

It is called so, because there is no correct answer in an unsupervised learning. There is no teacher here like that is in supervised learning. Algorithms are left on their own devises to discover and present the interesting structure in the data.

The basic aim of Unsupervised learning is to model the distribution in the data so that more about the data could be learned.

If there is a basket filled with some fresh fruits, the task of unsupervised learning is to arrange the same type of fruits at one place.

There is no information about the fruits beforehand as there is no training data here. So, the task will be to figure out the object without any prior knowledge. There is no response value either.

First, the physical characteristics of a specific object is selected. For example, the selected attribute is its color. Then the objects would be arranged on the basis of the color.

Here, there is no significance in knowing or learning anything beforehand for unsupervised learning.

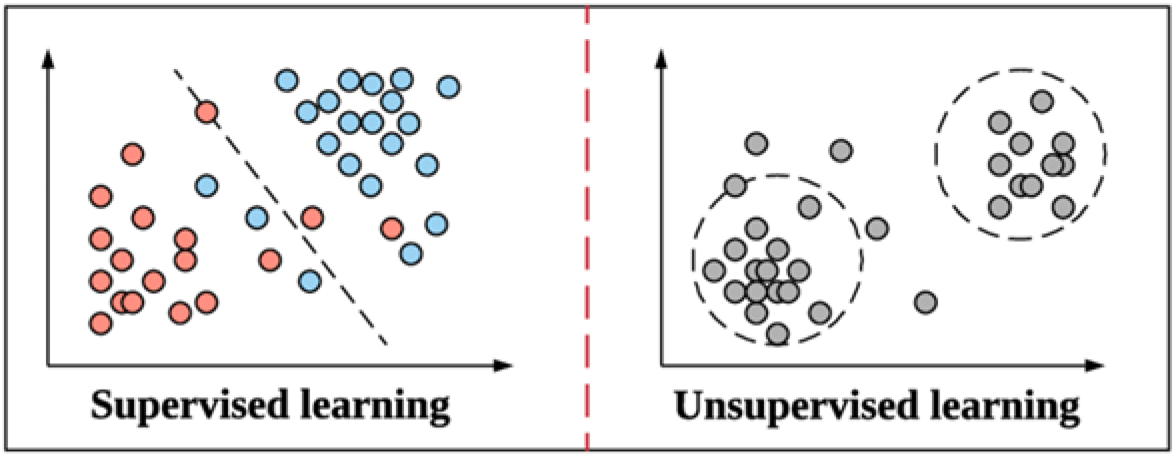


Fig 1. 5 Difference between supervised and unsupervised learning

1.1.6 Classification of IDS Based on Location

IDS can be classified into two categories based on location:

- Host Based IDS
- Network Based IDS

Getting access to the internet, HIDS runs on all the devices in the network with and other parts of the enterprise network. HIDS possess some advantages over NIDS, because of their ability to visualize more closely at internal traffic. It also functions well as a second line of defense against malicious activities that a NIDS fails to detect.

It checks the file set of the entire system and compares it to its priorly saved image of the file set. After that it checks whether there are any significant differences outside normal business use and then alerts the administrator if there is any missing or significantly altered files or settings.

A Network Intrusion Detection System (NIDS) is generally placed at all the strategic points throughout the network so that it can cover the places where traffic is most likely to be vulnerable to attack. Usually, it is applied to all the subnets so that it can attempt to match any traffic passing by to a library of known attacks. It passively keeps checking at the network traffic coming through the subnet points on the network on which it is placed. They are most like to be relatively easy to secure and are difficult to detect for the intruders. So, an intruder may not realize that their potential attack is being detected by the NIDS.

NIDS software has to analyze a large amount of network traffic. So, sometimes the specificity turns low. As a result, sometimes an attack might be missed or might not be able to detect something happening in encrypted traffic. In specific cases, more manual involvement might be needed from an administrator to ensure their configuration correctly.

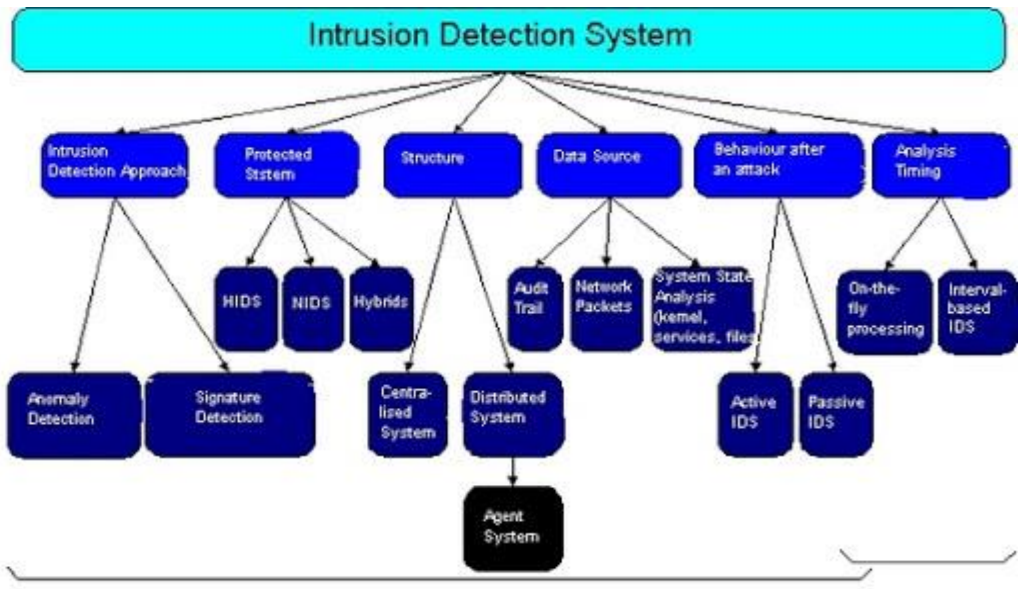


Figure 1. 5 Classification of IDS

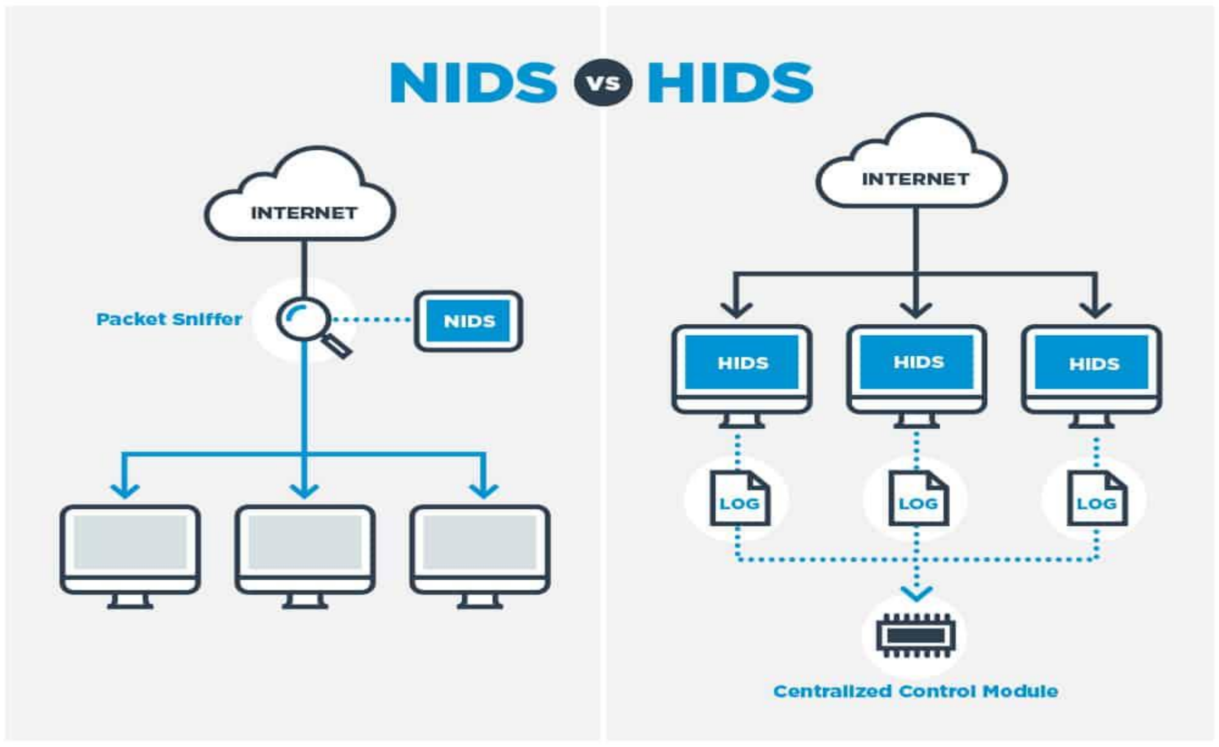


Figure 1. 6 Difference Between NIDS and HIDS

1.1.7 Intrusion Prevention System (IPS)

An Intrusion Prevention System (IPS) is essentially a danger anticipation innovation which inspects network traffic streams to detect and intercept vulnerability exploits. vulnerability exploits refer to malicious inputs to a target application or software that attackers use to break in and take control of. After a successful exploitation, the attacker can debilitate the target or can possibly gain access to all the available rights and permissions to the undermined application.

The IPS frequently sits legitimately behind the firewall and gives an integral layer of investigation that adversely chooses for risky substance. Unlike its archetype the Intrusion Detection System (IDS) which is a passive system that inspects traffic and reports back on possible threats, the IPS is put inline in the immediate correspondence path among source and destination, effectively breaking down and taking computerized actions on all traffic streams that enter the network system. The possible actions include:

- Sending alarms to the administrator just like an IDS
- Erasing the malicious packets
- Blocking traffic from the source address
- Resetting the connection

As an inline security segment, the IPS must work effectively to abstain from mortifying network performance. It should likewise work quick since endeavors can occur in near real-time. The IPS should also be able to detect and react precisely, in order to take out threats and false positives.

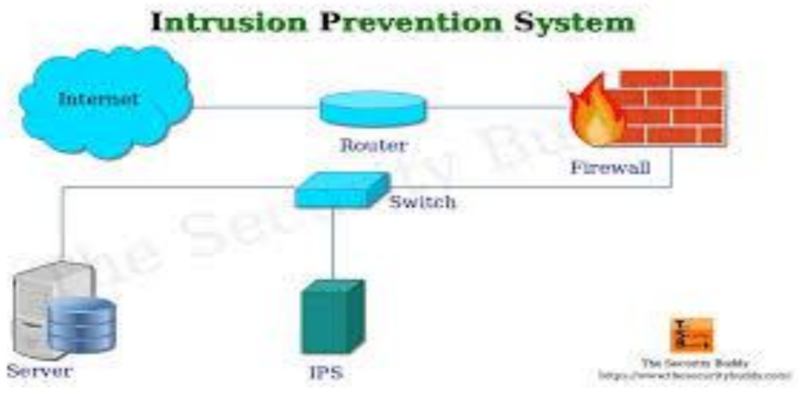


Figure 1. 7 Intrusion Prevention System

1.1.8 Difference Between IDS and IPS

An IDS is an intrusion detection system, not necessarily designed to respond to an attack. An IDS is capable of being a part of a larger security tool with responses and remedies, but is not able to prevent an attack from happening. It is merely a monitoring system.

Another kind of security system is the Intrusion Prevention System or IPS. An IPS is basically an IDS incorporated with a response or control system. IDS fail to alter or modify network packets as they come through. On the other hand, an IPS is able to prevent the packet from being delivered if it senses any malicious activities in the packet. So basically, it prevents packet delivery based on the contents of the packet.

A signature-based IDS and IPS refer to a database of known threats and is able to generate flags or respond to the threat based on this database. IDS require an administrator to check on the results of the flagged items, whereas an IPS would automatically take necessary actions to block the threat.

Overall, both IDS and IPS are largely hands-off. An IDS would simply send alerts to the administrators so that they can respond to the attacks. It is not mandatory for the admin to review every single IDS process themselves. As a result, these turn out to be beneficial and low-maintenance tools in a security stack.

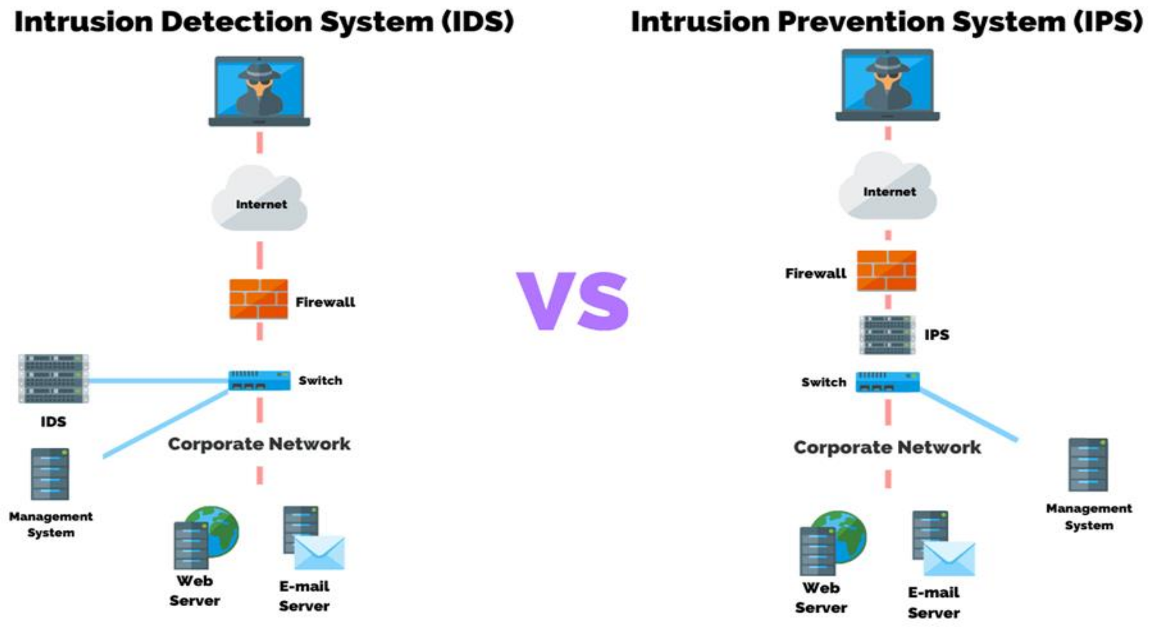


Figure 1. 8 Difference between IDS and IPS

1.1.9 Two Layered System

Two Layered system refers to the algorithms being used twice for classification purposes. To design an efficient model for an IDS requires some methods and techniques. The quality of a model depends on the techniques that have been used. In a single layer application, the used algorithm classifies the data ones. But if multiple layers are used, the whole data gets classified multiple times which leads to better classification. Better classification leads to getting better result.

In a two layered system, the data would get classified twice with two types of classifying algorithm. So, the data would be evaluated twice getting a better response than the previous one.

The advantage of designing a two layered model is that the data could be collected through communicating directly to the database which might not be possible in a single layered one.

1.2 Motivation

First, we had interest about internet security. While researching on unauthorized access and malicious activities, the concept of designing a model for an ideal intrusion detection system came into our minds. For any institution, it is very important to protect its server from all kinds of infiltration. Bangladesh Bank heist is an example of how dangerous keeping a server unprotected could be. This is where the importance of IDS can be understood.

IDS can detect if any other network is trying to infiltrate or what kind of attack is being sent to the server and thus keep the server safe. For that reason, we have chosen to design an IDS that is capable of detecting most of the attacks while providing less false alarms at a low computational cost.

1.3 Objectives

Aim: proposing a model which can detect anomalies at a low computational cost and false alarm rate.

Objective:

- Detecting multi attack types
- Reducing complexity
- Reducing false alarm rate
- Getting higher detection rate against all types of attacks (DOS, Probe, U2R, R2L)
- Getting better accuracy

1.5 Contribution

The contribution of our research is as follows:

Our research is all about designing a model for an ideal intrusion detection system in which the target is to detect all possible attacks and give out warnings. The purpose is to avoid any type of danger regarding networks or systems.

In the corporate world, preventing unwanted intrusion or unauthorized access is very much important. Information theft can lead an industry to go bankrupt. Flooding the network system of a significant organization can lead to huge difficulties not only for the employees but also for the general people who are served by that organization. Hacking can make it face huge amount of losses or even go bankruptcy.

This research is for avoiding a critical situation in which a network or system gets harmed due to intrusion. It may not provide functions for blocking any malicious activities but can provide warnings so that any kind of unwanted danger could be avoided.

1.6 Outline of the Report

The outline of the rest of this report has been structured as follow:

Chapter 2: Literature Review

Chapter 3: Proposed Method

Chapter 4: Implementation Details

Chapter 5: Result Analysis and Discussion

Chapter 6: Conclusion and Future work

CHAPTER 2

LITERATURE REVIEW

2.1 Background Study

In the previous chapter, we discussed about the basics of intrusion detections system. We realized the significance of a good intrusion detection system and how a good model can protect a network from harmful attacks.

Machine Learning is a sub-area of artificial intelligence, whereby the term refers to the ability of IT systems to independently find solutions to problems by recognizing patterns in databases. In other words: Machine Learning enables IT systems to recognize patterns on the basis of existing algorithms and data sets and to develop adequate solution concepts. Therefore, in Machine Learning, artificial knowledge is generated on the basis of experience.

Machine learning algorithms are the engines of machine learning, meaning it is the algorithms that turn a data set into a model. Which kind of algorithm works best (supervised, unsupervised, classification, regression, etc.) depends on the kind of problem you're solving, the computing resources available, and the nature of the data.

There are two kinds of machine learning algorithms: supervised and unsupervised. In supervised learning, a training data set is provided including answers, such as a set of pictures of flowers along with the names of the flowers. The goal of that training would be a model that could correctly identify a picture that had not been previously. In unsupervised learning, the algorithm goes through the data by itself and tries to emerge with meaningful results. The result might be, a set of clusters of data points that could be related within each cluster. That works better when the clusters do not tend to overlap.

2.2 Related Works

The study detects anomaly with the help of a two-step classification model. Many models have been proposed for anomaly detection based on, machine learning approaches. Some of them are statistical approaches and some are not.

In paper [1], They have proposed an SVM-based intrusion detection system, which combines a hierarchical clustering algorithm, a simple feature selection procedure and the SVM technique. They were able to detect common types of attacks easily but failed to detect rare attacks efficiently. Most of the proposed models for detecting anomalous activities use statistical approaches like dimension reduction (e.g. Principal Component Analysis (PCA) and Filter Method). In paper [5], they have proposed an unsupervised anomaly detection model that uses grid-base clustering based on subspace algorithm for finding anomalous activities though they failed to mention how specific attack types should be dealt with. In paper [6], they have proposed a model based on distance and density of clusters to figure out the attacks. But their complexity was high which increased the computational cost. In paper [7], they proposed a model combining misuse detection and anomaly detection components using the random forests algorithm but their detection rate was not sufficient. In paper [8], they have combined a neuro-fuzzy network, the fuzzy inference approach and genetic algorithms to design an intrusion detection system. The model was able to get high detection rate on major attacks but still suffers from low detection rate on rare and difficult attacks. In paper [9], they have proposed a three-level supervised classification model using decision tree and Na¨ıve Bayes and also Bayesian clustering to detect anomalous activities and their model was able gain good results on different types of attacks. But there was a high rate of false alarms. In paper [10], they applied PCA for dimension reduction and also Na¨ıve Bayes algorithm for classification of anomalous behaviors. They had applied several combination feature selection methods to get to their desired result but failed to detect unseen attack types. In paper [12], they have proposed a model based on clustering algorithms to extract prominent features from dataset obtaining high detection rate on normal and DoS classes but not on unseen rare type of attacks. They were able to do so as their frequent pattern being in both training and test set.

In paper [13], they have proposed a model which uses fuzzy classification and Evolutionary Algorithms for evolving fuzzy classifiers to detect anomalies. Unfortunately, the computational cost was higher than many other models. In paper [14], they have proposed a logistic regression-based anomaly detection system exploiting hierarchical feature reduction to differentiate between anomalous behaviors and normal ones. Their detection rate of rare attack types like U2R and R2L was praiseworthy but increase in false alarm rate added as a disadvantage. In paper [15], the writers explained the uses and benefits of machine learning in cyber security and beyond. But no specific techniques were introduced to get a good IDS model. In paper [20], they have proposed a two layered model that uses Random Forest and SVM as their classifiers. The study achieved very good detection rates. In paper [26], the researchers used filter method as its dimensionality reduction module and used Random Forest algorithm to classify the dataset. They achieved good detection rates but not enough against all sorts of attack types. In paper [22], the writers explained the functionalities of the feature selection process and stated the significant benefits it is able to provide a dataset ready to be classified.

CHAPTER 3

PROPOSED METHOD

3.1 Problem domain

First tier consists data preprocessing and dimension reduction which has better result for decision making and first stage of classification using Support Vector Machine (SVM). At the second tier of the proposed model for better separation between normal and anomalous activities, specific classification using Random Forest algorithm will be performed.

For better disjunction between normal and anomalous classes, feature selection would be performed prior to performing classification. In this study, new feature space contains of five dimensions, in this phase filter co-relation feature selection method would be used which would select five effective features (fm1, fm2, fm3, fm4, fm5) out of 42 and then the classification would be performed. After that SVM algorithm would be applied to the dataset as a classifier which would distinguish between the normal and the anomalous data.

To improve the performance of the used classifier, a technique using RF-d tree would be applied to store the diminished data training set. At this stage an elaborated analysis of the proposed model could be explained. The proposed model is capable of converting a high dimensional data set into a lower one. And the model can perform its classification by using two machine learning classifiers making this a two layered model.

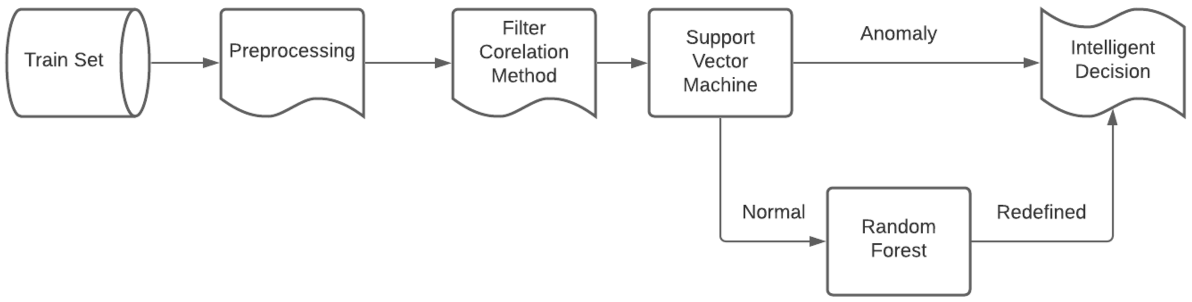


Fig 3. 1 Research framework for our proposed method

3.2 Dimension Reduction

The number of input variables or features for a dataset can be referred to as its dimensionality. Dimensionality reduction refers to the techniques that reduce the number of input variables in a dataset. More input features often make a predictive modeling task more challenging to model, more generally referred to as the curse of dimensionality.
High-dimensionality statistics and dimensionality reduction techniques are often used for data visualization. Nevertheless, these techniques can be used in applied machine learning to simplify a classification or regression dataset in order to better fit a predictive model.

3.2.1 Feature Selection

Feature selection (FSS) is a commonly used pre-processing step in machine learning or data mining. It is pretty much efficient in dimensionality reduction and is able to remove redundant features which results in increasing accuracy of the model. It aids to the problem of identifying features which are convenient in predicting classes. Feature selection methods can be classified into three categories:

- filter method,
- wrapper method and
- embedded method [20].

In this study, Filter method has been used for dimensionality reduction and feature selection. The aim of FS method is to choose a subset of the pertinent features of the original dataset of features based on some criteria such as correlation, redundancy, and inconsistency.
Feature selection is a huge step in providing a high accuracy rate in supervised classification. It also contributes in improving the quality of classification and reducing the computational complexity of any classification algorithm. The core aim is to decrease the dimensionality by selecting the smallest subset of features from the original feature set which is able to achieve maximum classification accuracy. This task helps classification performance by eradicating the irrelevant and redundant features [21].

3.2.1.1 Filter Method:

Feature selection is a frequently used technique to reduce the number of features in many applications which involves data of high dimensionality. Both theoretically and practically, feature selection is proven to be efficient in reducing or decreasing the dimensionality, eradicating irrelevant and redundant features among the relevant ones. It has been also proven that feature selection technique is capable of enhancing learning efficiency, improving predictive accuracy and decreasing complexity of learned results. A lot of the feature selection methods primarily focus on estimating the correlation (or similarity) between two features. However, most correlation measures are restricted to handling only certain data types. Feature space containing continuous or discrete feature or their combination may present a severe challenge to feature selection in terms of efficiency and effectiveness [22]

Considering the correlation between two variables as a goodness measure, the definition turns out that a feature is pleasant if it is highly correlated to the class. But not being highly correlated to any of the other features indicates the opposite. In other words, if the correlation between a feature and the class is high enough to make it seem relevant to the class and the correlation between it and any other relevant features is unable to reach a level so that it can be assumed by any of the other relevant features, it will be regarded as a good feature for the classification task.

The most well-known measure is linear correlation co-efficient. For a pair of variables (X, Y), the linear correlation coefficient r is given by the formula

$$\frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2}\ \sqrt{\sum_i (y_i - \bar{y}_i)^2}}$$

where $\bar{x}_i$ is the mean of X, and $\bar{y}_i$ is the mean of Y, the value of r lies between -1 and 1. If X and Y are completely correlated, r takes the value of 1 or -1; if X and Y are totally independent, in that case r is zero [22].

At first, it helps to remove features with close to zero linear correlation to the class. Secondly, it helps to decrease redundancy among selected features.

Linear correlation measures are able to capture linear correlations but may be unable to capture correlations which are not linear in nature

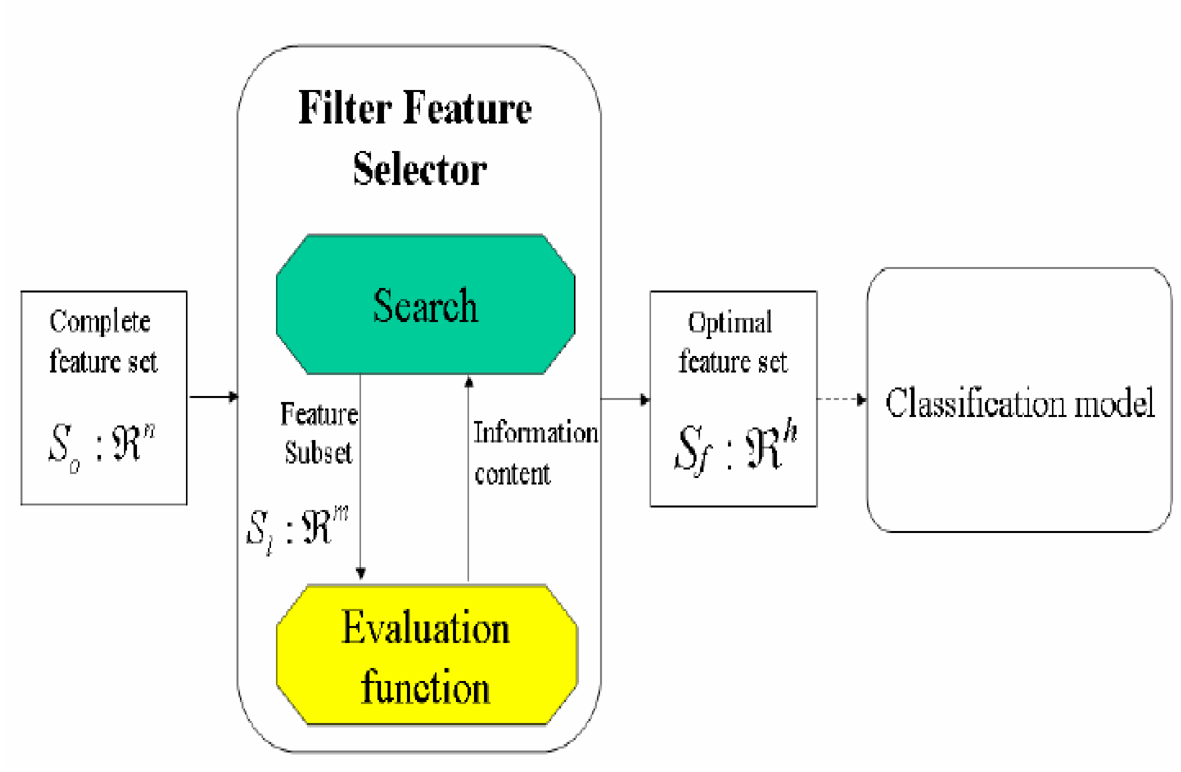


Fig 3. 2 Feature Selection Model Based on Filter Method

Fig shows the working procedure of Filter Co-relation method. A complete feature set is converted in to different subsets and then an optimal feature set thus evaluating the function. The process ends in classifying the model and reducing its dimensions.

3.3 Algorithm

3.3.1 Random Forest

The random forest algorithm is an ensemble of unpruned classification or regression trees [15]. On the other hands Random fore (RF) is an ensemble classifier used to improve the accuracy. Random forest consists of many decision

trees. Random forest is able to generate many classification trees and each tree is manufactured by a different bootstrap sample from the original data which uses a tree classification algorithm. After the formation of the forest, a new object needed to be classified is put down to each of the trees in the forest for classification. Each tree gives a vote that specifies the tree's decision about the class of the object. Random forest has low classification error in comparison with the other traditional classification algorithms. Number of trees, minimum node size and number of features are used here for splitting each node [20] [25].

Advantages of Random Forest Classifier are listed below:

- Generated forests could be saved for future reference.
- Random forest is capable of overcoming the over fitting problem.
- In Random Forest classifiers, accuracy and variable importance are usually generated automatically.

While constructing the individual trees in a random forest, the best node is selected to split on by applying randomization [20].

In this section. the Random Forest has been presented along with its training

procedure and its additional features.

Random Forest is an ensemble of B trees. $\{T_1(X), ..., T_B(X)\}$, where $X = \{x_1, ..., x_p\}$ is a

p-dimensional vector of molecular descriptors or properties associated with a molecule. The ensemble produces B outputs. $\{\hat{Y}_1 = T_1(X), ..., \hat{Y}_B = T_B\ (X)\}$ where $\hat{Y}_B$, b=1, ..., B, is the prediction for a molecule by the b^{th} tree. Outputs of all trees are aggregated to produce one final prediction, $\hat{Y}_B$. For classification problems, V is the class predicted by the majority of trees. In regression it is the average of the individual tree predictions [12] [24] [21].

Random Forest can be implemented with scikit-learn software. The final feature importance, at the Random Forest level, is its average over all the trees. The sum of the feature's importance value on each tree is calculated and divided by the total number of trees:

$$RFfi_i = \frac{\sum_{j \in all\ trees} normfi_{ij}}{T}$$

Where,

- RFfi sub(i)= the importance of feature i calculated from all trees in the Random Forest model
- normfi sub(ij)= the normalized feature importance for i in tree j
- T = total number of trees

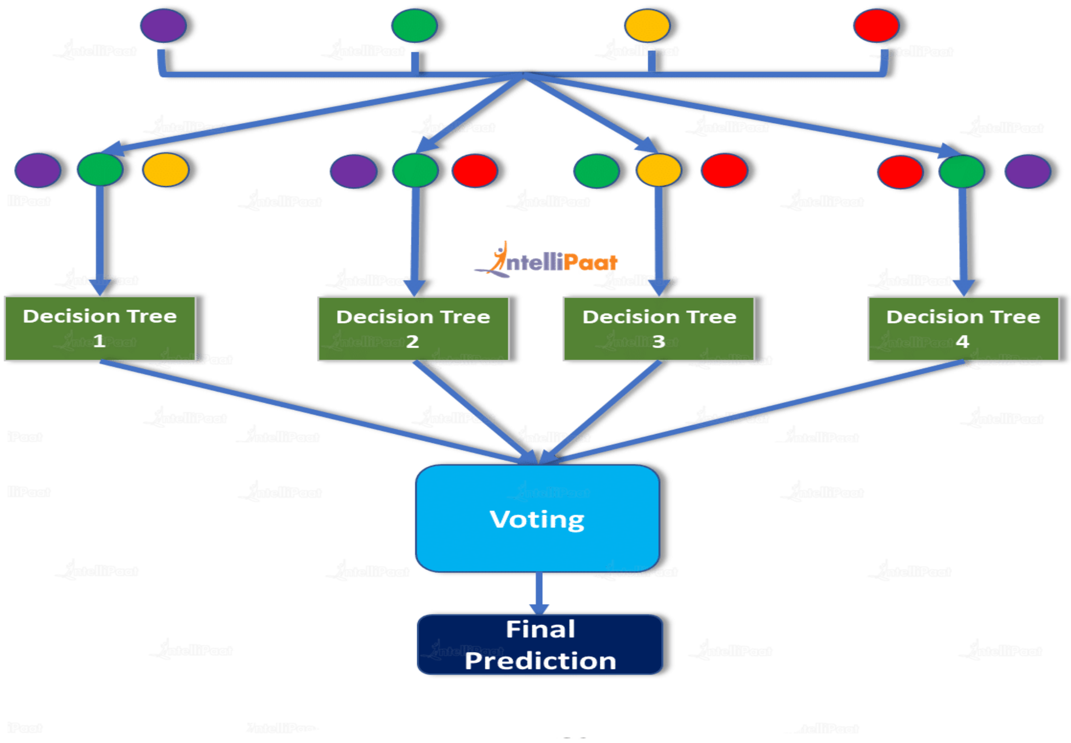


Fig 3. 3 Example of a Random Forest tree

3.3.2 Support Vector Machine

Support Vector Machine is a supervised learning method [23]. It is able to perform classification by constructing an N-dimensional hyperplane that optimally separates the data into different categories. In the basic classification, SVM classifies the data into two categories.

Given a training set of instances, labeled pairs {(x, y)}, where y is the label of instance x, SVM works by maximizing the margin to obtain the best performance in classification [11].

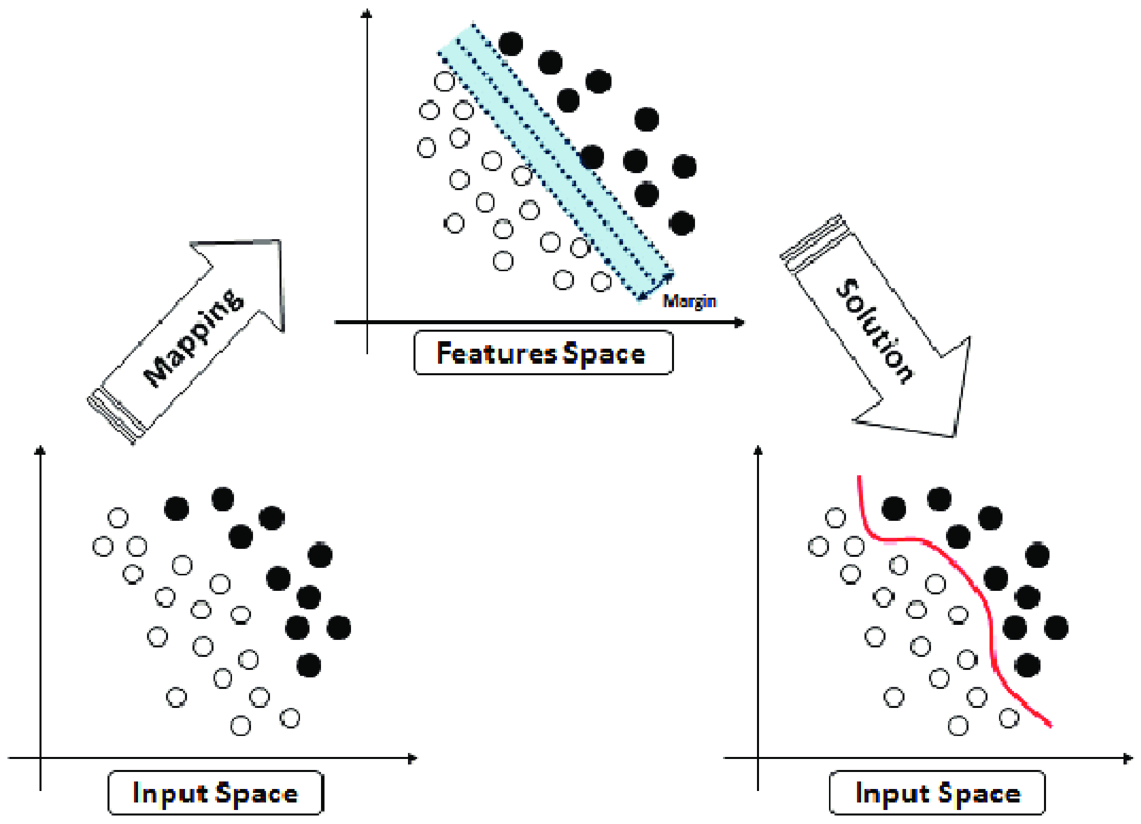


Fig 3. 4 SVM-Classifier

In SVM, the problem of computing a margin-maximizing boundary function is specified by the following quadratic programming (QP) problem:

minimize:

$$W(\alpha) = -\sum_{i=1}^{\ell} \alpha_i + \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} y_i y_j \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j$$

subject to:

$$\sum_{\substack{i=1 \\ 0 \leq \alpha_i \leq C}}^{\ell} y_i \alpha_i = 0$$

where l is the number of training data, a is a vector of l variables, in which each component ai corresponds to the training data xi, and C is the soft margin parameter, which controls the influence of the outliers (or noise) in the training data. The boundary function of SVM is described by support vectors, which are the data points located closest to the class boundary. The above quadratic programming problem computes vector a, where each element specifies a weight of each data point. Those data points, whose ai's value are greater than 0, are the support vectors; and data points with ai value equal to zero are the non-support vectors. For the boundary function, only support vectors are useful and considered. In the SVM, all training data points are mapped into a higher dimensional space. Then, SVM can find a separating hyperplane with a maximal margin in this higher dimensional space [11] [25].

$K(x_i, x_j) = (x_i \cdot x_j + 1)^p;$ polynomial kernel.

$K(x_i, x_j) = e^{\frac{-1}{2\sigma^2}(x_i - x_j)^2};$ Gaussian kernel; Special case of Radial Basis Function.

$K(x_i, x_j) = e^{-\gamma(x_i - x_j)^2};$ RBF Kernel

$K(x_i, x_j) = \tanh(\eta\, x_i \cdot x_j + \nu);$ Sigmoid Kernel; Activation function for NN.

3.4 Dataset

A detailed analysis of the applied dataset will be discussed first at this section, then performance of IDS will be stated and at the end evaluation of the proposed model will be explained.

3.4.1 NSL-KDD Dataset

NSL-KDD benchmark dataset has been used to evaluate the proposed model. The newer version of NSL-KDD dataset is KDD99 dataset [18]. This data set has been introduced for network intrusion detection systems competition. Each NSL-KDD record contains a host-to-host connection which includes 41 distinguished features (e.g. Num_failed, logins, Root_shell, Count) which are called as normal, anomaly or one of the specific attack names. All attacks can be categorized into four major groups: DoS, probe, U2R and R2L. The feature vector contains three categorical or discrete values, five symbolic values and the rest are continuous values. NSL-KDD dataset has been applied to evaluate the proposed model since the original KDD dataset had some flaws [10] [18].

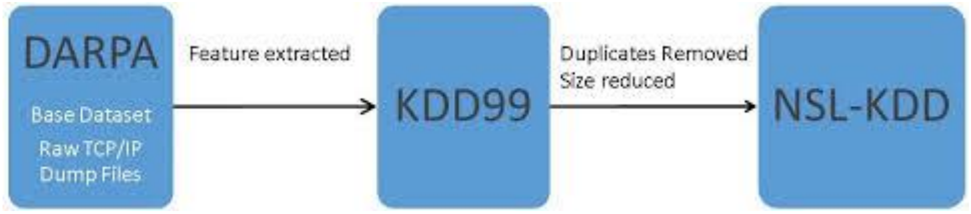


Fig 3. 5 Evaluation of NSL-KDD Dataset

The dataset consists of two training sets and one test set which contains DoS, Probe, U2R, R2L attack classes beside normal one.

3.4.2 Data Preprocessing

the original dataset will be converted to a normal form in the data preprocessing phase in order to get better decision making and mitigating the computational overhead [4], it will be as follows:

- Each categorical feature values will be assigned to a unique integer number like (TCP = 1, UDP = 1, ICMP = 1).
- Continuous-valued features will be discretized using logarithm to the base 2 and then casting the result value to integer for avoiding any bias. This step uses this equation for each Continuous-valued z.

$$\text{if } (z \geq 2)\ z = \log (z + 1)$$

After the normalization for better classification, attack labels will be grouped into four major classes and a normal class [4]

$$Detection\ Rate: DR = \frac{TP}{FN + TP}$$
$$False\ Alarm\ Rate: FAR = \frac{FP}{FP + TN}$$
$$False\ Alarm\ Rate: FAR = \frac{FP}{FP + TN}$$

3.4.3 Complexity analysis

As it has been said in the contributions section, this has model provided "lower computational complexity on account of optimal dimension reduction". With respect to this reduction, in the first step SVM classifier has been applied and its computational complexity reflects as ***O(k*d)***, where ***k*** is number of support vectors in the respective dataset and ***d*** indicates dimensionality of data [20] [25]. In this stage, due to filter correlation method transformation, the classifier will be fed with only five attributes instance of 41, the computation complexity reduced by fifteen folds. On other hand, at the step where Random Forest algorithm has been applied, the model just needed to remember only two dimensions of training set [24].

As a result, now it would consume less space than the original dataset. In addition to the ensemble classification, (RF-d tree) has been used in this study for searching decision tree and due to the preprocessing phase and summarization, all features have an integer value now. So, finding decision tree will be more convenient and in such (two dimensions) RF-d tree points takes ***O*(*max depth of tree** *k*)** time on average [26].

3.4.4 Testing environment and Constraints

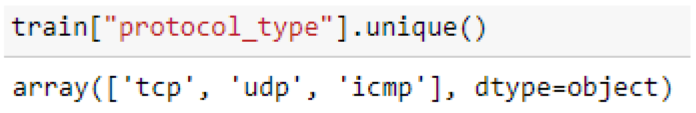
The experiment was processed within a JUPYTER NOTEBOOK environment, which was running on a laptop powered by INTEL Core I3 2.10 GHz CPU and 12 GB RAM.

CHAPTER 4

IMPLEMENTATION DETAILS

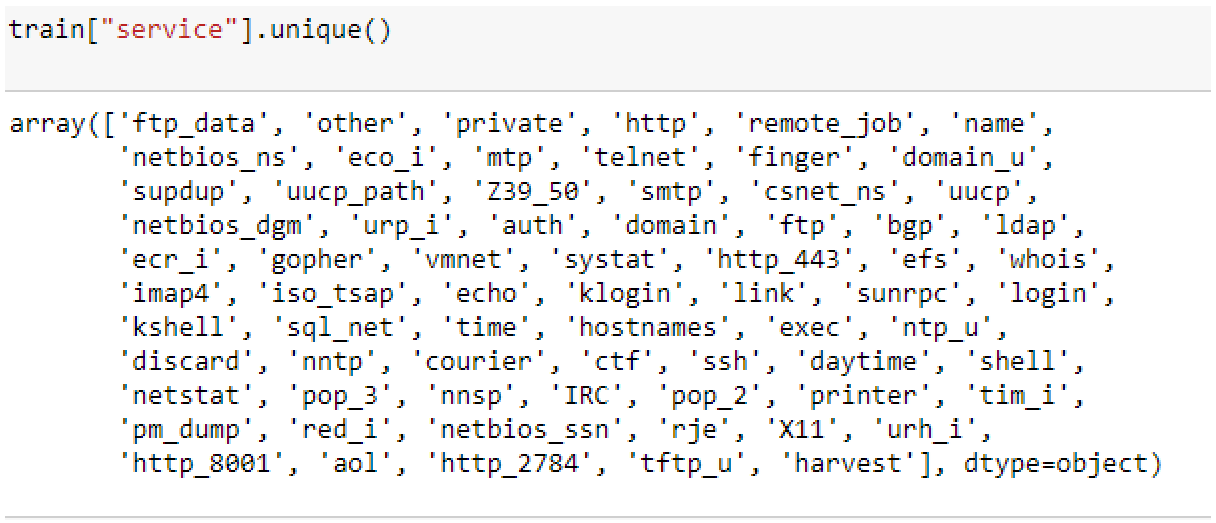
4.1 Data Preprocessing

In the proposed model, the data trained data has been preprocessed. The preprocessing part has been implemented by converting the strings to numerical values. After that it has been added to the main dataset.



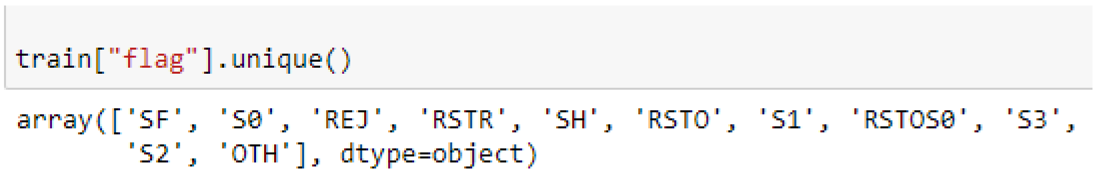
```
train["protocol_type"].unique()
```

```
array(['tcp', 'udp', 'icmp'], dtype=object)
```



```
train["service"].unique()
```

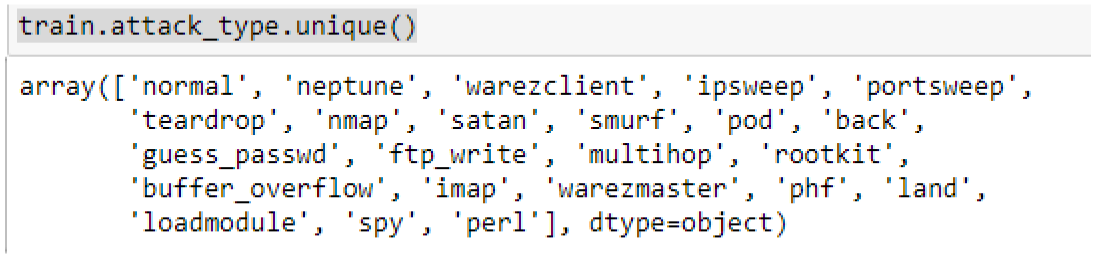
```
array(['ftp_data', 'other', 'private', 'http', 'remote_job', 'name',
       'netbios_ns', 'eco_i', 'mtp', 'telnet', 'finger', 'domain_u',
       'supdup', 'uucp_path', 'Z39_50', 'smtp', 'csnet_ns', 'uucp',
       'netbios_dgm', 'urp_i', 'auth', 'domain', 'ftp', 'bgp', 'ldap',
       'ecr_i', 'gopher', 'vmnet', 'systat', 'http_443', 'efs', 'whois',
       'imap4', 'iso_tsap', 'echo', 'klogin', 'link', 'sunrpc', 'login',
       'kshell', 'sql_net', 'time', 'hostnames', 'exec', 'ntp_u',
       'discard', 'nntp', 'courier', 'ctf', 'ssh', 'daytime', 'shell',
       'netstat', 'pop_3', 'nnsp', 'IRC', 'pop_2', 'printer', 'tim_i',
       'pm_dump', 'red_i', 'netbios_ssn', 'rje', 'X11', 'urh_i',
       'http_8001', 'aol', 'http_2784', 'tftp_u', 'harvest'], dtype=object)
```



```
train["flag"].unique()
```

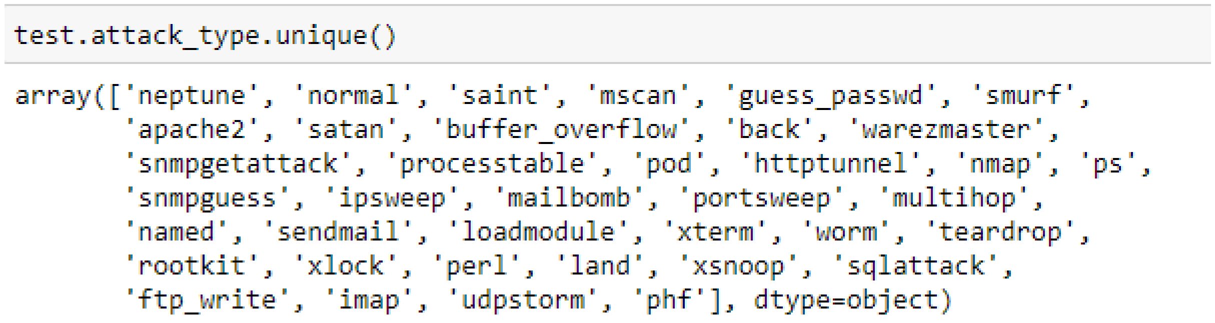
```
array(['SF', 'S0', 'REJ', 'RSTR', 'SH', 'RSTO', 'S1', 'RSTOS0', 'S3',
       'S2', 'OTH'], dtype=object)
```

4.2 Attack types for Train, Test and Train_20% Dataset



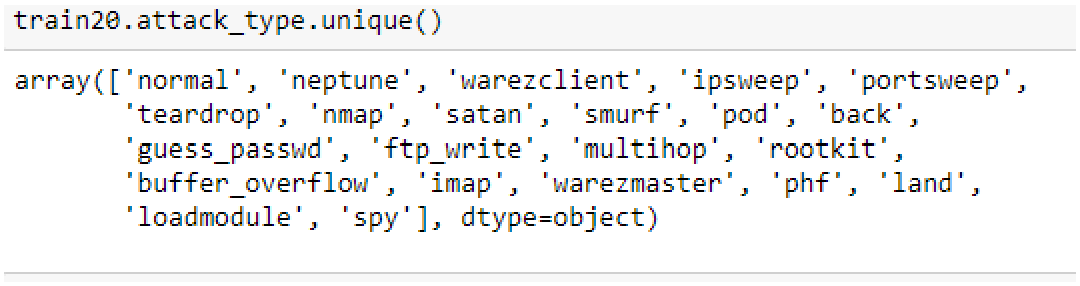
```
train.attack_type.unique()
```

```
array(['normal', 'neptune', 'warezclient', 'ipsweep', 'portsweep',
       'teardrop', 'nmap', 'satan', 'smurf', 'pod', 'back',
       'guess_passwd', 'ftp_write', 'multihop', 'rootkit',
       'buffer_overflow', 'imap', 'warezmaster', 'phf', 'land',
       'loadmodule', 'spy', 'perl'], dtype=object)
```



```
test.attack_type.unique()
```

```
array(['neptune', 'normal', 'saint', 'mscan', 'guess_passwd', 'smurf',
       'apache2', 'satan', 'buffer_overflow', 'back', 'warezmaster',
       'snmpgetattack', 'processtable', 'pod', 'httptunnel', 'nmap', 'ps',
       'snmpguess', 'ipsweep', 'mailbomb', 'portsweep', 'multihop',
       'named', 'sendmail', 'loadmodule', 'xterm', 'worm', 'teardrop',
       'rootkit', 'xlock', 'perl', 'land', 'xsnoop', 'sqlattack',
       'ftp_write', 'imap', 'udpstorm', 'phf'], dtype=object)
```



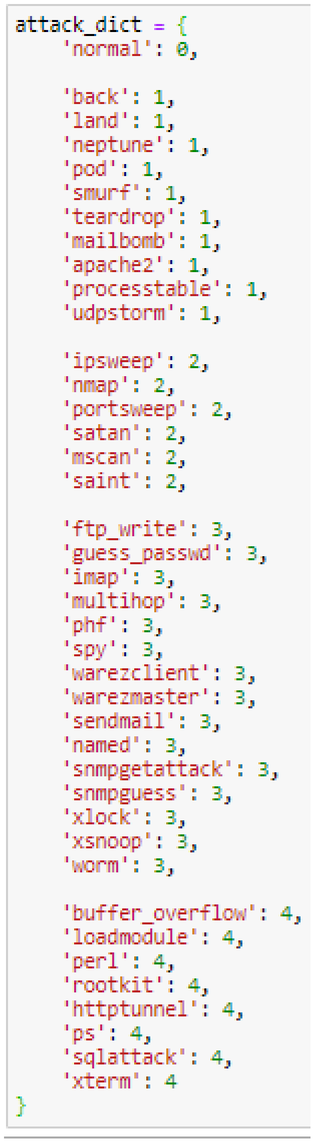
```
train20.attack_type.unique()
```

```
array(['normal', 'neptune', 'warezclient', 'ipsweep', 'portsweep',
       'teardrop', 'nmap', 'satan', 'smurf', 'pod', 'back',
       'guess_passwd', 'ftp_write', 'multihop', 'rootkit',
       'buffer_overflow', 'imap', 'warezmaster', 'phf', 'land',
       'loadmodule', 'spy'], dtype=object)
```

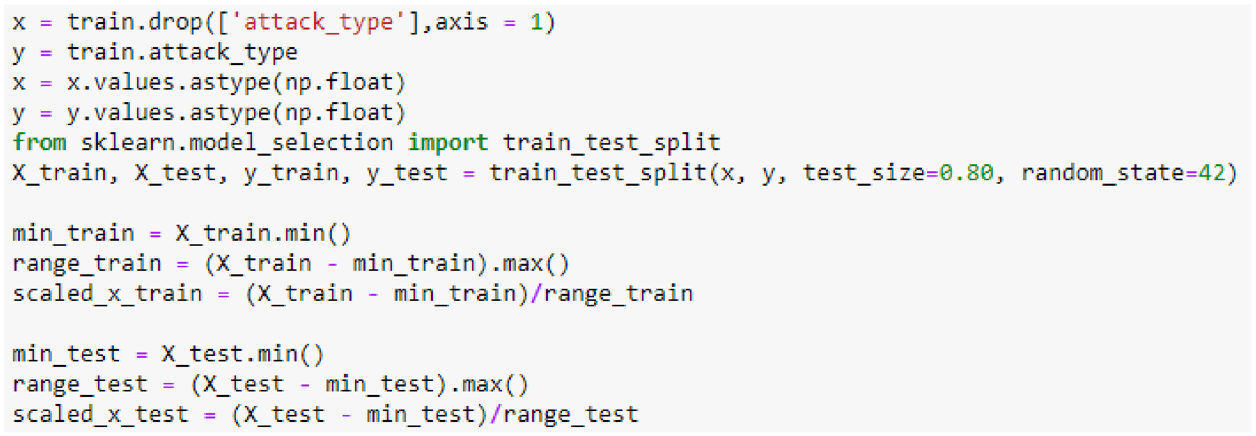
In this section the sub attack types have been derived. The train set consists of 23 sub attack types, test set consists of 38 sub attack types and train_20 consists of 22 sub attack types. The target values of the dataset are the attack types. The next step is to extract the attack types for the sub attack ones.

4.3 Converting to Numerical Values

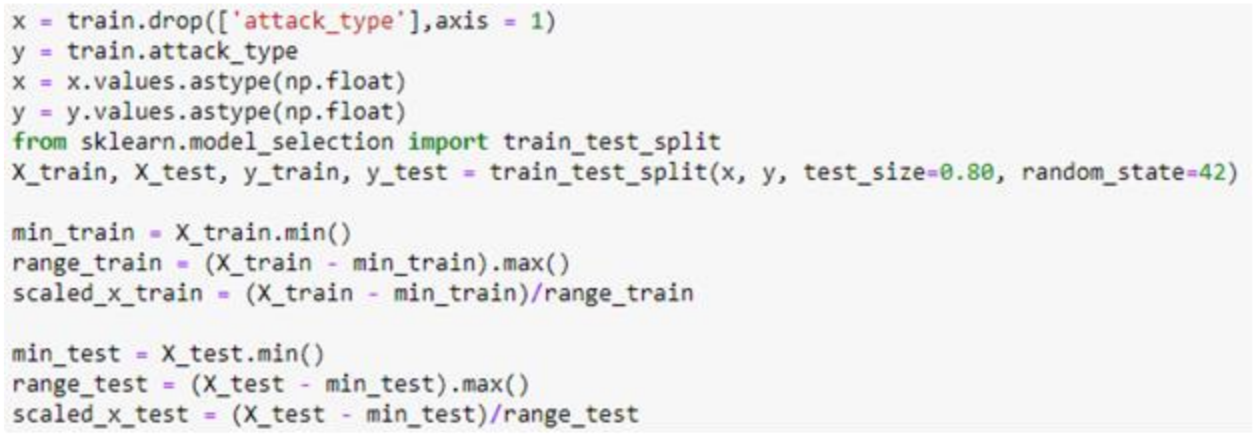
To calculate detection rate and false alarm rate, the attack types has been considered as numerical values (0,1,2,3,4).



```
attack_dict = {
    'normal': 0,

    'back': 1,
    'land': 1,
    'neptune': 1,
    'pod': 1,
    'smurf': 1,
    'teardrop': 1,
    'mailbomb': 1,
    'apache2': 1,
    'processtable': 1,
    'udpstorm': 1,

    'ipsweep': 2,
    'nmap': 2,
    'portsweep': 2,
    'satan': 2,
    'mscan': 2,
    'saint': 2,

    'ftp_write': 3,
    'guess_passwd': 3,
    'imap': 3,
    'multihop': 3,
    'phf': 3,
    'spy': 3,
    'warezclient': 3,
    'warezmaster': 3,
    'sendmail': 3,
    'named': 3,
    'snmpgetattack': 3,
    'snmpguess': 3,
    'xlock': 3,
    'xsnoop': 3,
    'worm': 3,

    'buffer_overflow': 4,
    'loadmodule': 4,
    'perl': 4,
    'rootkit': 4,
    'httptunnel': 4,
    'ps': 4,
    'sqlattack': 4,
    'xterm': 4
}
```



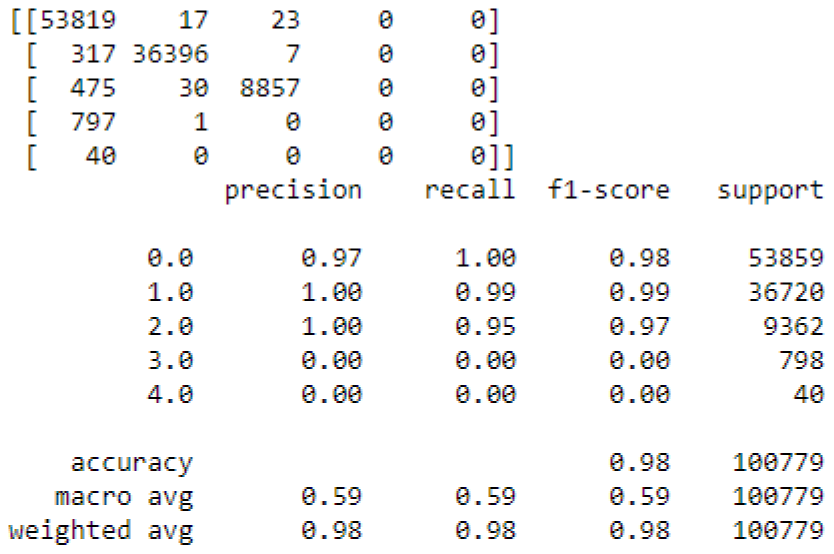
```python
x = train.drop(['attack_type'],axis = 1)
y = train.attack_type
x = x.values.astype(np.float)
y = y.values.astype(np.float)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.80, random_state=42)

min_train = X_train.min()
range_train = (X_train - min_train).max()
scaled_x_train = (X_train - min_train)/range_train

min_test = X_test.min()
range_test = (X_test - min_test).max()
scaled_x_test = (X_test - min_test)/range_test
```



```python
x = train.drop(['attack_type'],axis = 1)
y = train.attack_type
x = x.values.astype(np.float)
y = y.values.astype(np.float)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.80, random_state=42)

min_train = X_train.min()
range_train = (X_train - min_train).max()
scaled_x_train = (X_train - min_train)/range_train

min_test = X_test.min()
range_test = (X_test - min_test).max()
scaled_x_test = (X_test - min_test)/range_test
```

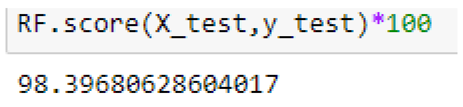
4.5 Calculating precision, Recall and Detection Rate

In this section, precision, recall, f_score, support has been derived by applying confusion matrix on the dataset. After normalization, two datasets have been derived based on y_predict and y_test.



```
[[53819    17    23     0     0]
 [  317 36396     7     0     0]
 [  475    30  8857     0     0]
 [  797     1     0     0     0]
 [   40     0     0     0     0]]
              precision    recall  f1-score   support

         0.0       0.97      1.00      0.98     53859
         1.0       1.00      0.99      0.99     36720
         2.0       1.00      0.95      0.97      9362
         3.0       0.00      0.00      0.00       798
         4.0       0.00      0.00      0.00        40

    accuracy                           0.98    100779
   macro avg       0.59      0.59      0.59    100779
weighted avg       0.98      0.98      0.98    100779
```

Based on precision, recall, f1_score and support, detection rate has been obtained. For the calculation part, the formulas for calculating detection and false positive rate has been used here in this section.



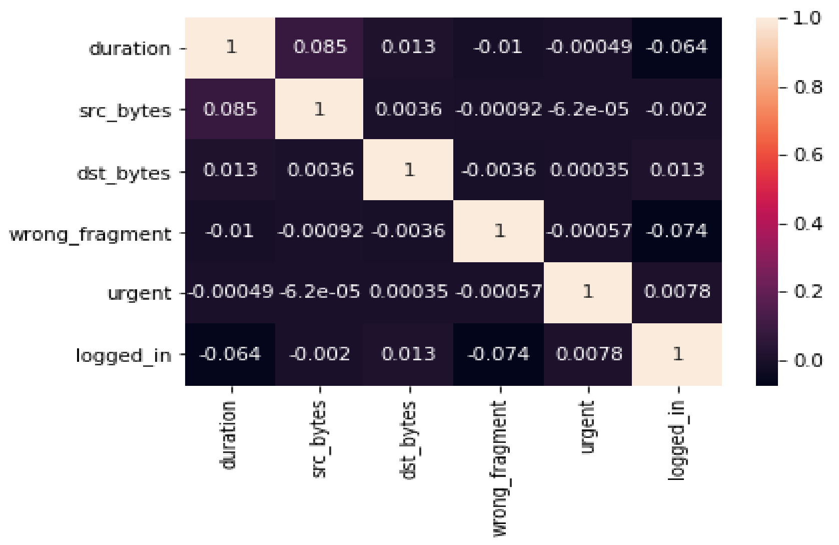
```
RF.score(X_test,y_test)*100

98.39680628604017
```

CHAPTER 5

RESULT ANALYSIS AND DISCUSSION

The proposed model was trained by both training set (Train 20%, $Train^{+}$) and then evaluated by given test set ($Test^{+}$) provided by NSL-KDD which contains 22544 instances. So, all of the given results in the research are evaluated by the test set. After normalizing the test set, the projection matrix (W_r) which was obtained from the training set, has been applied to the test set. But the proposed model is capable of almost solving this issue by using Random Forest as its second classifier. At this step around 45 iterations have been experimented. According to the detection rates of this experiment, five k values have been nominated to be applied in the proposed model, k = 5 has been chosen in order to obtain better detection rate on the rare classes of attacks in comparison with other nominated values.

```
<matplotlib.axes._subplots.AxesSubplot at 0x2151beab508>
```



	duration	src_bytes	dst_bytes	wrong_fragment	urgent	logged_in
duration	1	0.085	0.013	-0.01	-0.00049	-0.064
src_bytes	0.085	1	0.0036	-0.00092	-6.2e-05	-0.002
dst_bytes	0.013	0.0036	1	-0.0036	0.00035	0.013
wrong_fragment	-0.01	-0.00092	-0.0036	1	-0.00057	-0.074
urgent	-0.00049	-6.2e-05	0.00035	-0.00057	1	0.0078
logged_in	-0.064	-0.002	0.013	-0.074	0.0078	1

Fig 5. 1 Possible Combinations of mapped features in train_20% dataset

Figure 5. 1 shows the comparison between detection rates and false alarm rates on all possible combinations of new mapped features in Train20 % training set. According to this study, the highest detection rate belongs to the combination of the first two attributes (dst_host_same_srv_rate, dst_host_diff_srv_rate). The result shows that after dimensional reduction the feature value comes in between -1 to 1.

5.1 Coefficient Co-relation Measure

In this study, dimensional reduction has been done by performing a filter co-relation method. Five new features (*fm1, fm2, fm3, fm4, fm5*) have been obtained after the dimensionality reduction.

Table 5. 1 Assessment of new mapped attributes dependency by correlation co efficient measure in train_20%

Features	*fm1*	*fm2*	*fm3*	*fm4*	*fm5*
fm1	1	0	0	0	0
fm2	0	1	0	0	0
fm3	0	0	1	0	0
fm4	0	0	0	1	0
fm5	0	0	0	0	1

These new features have been obtained by evaluating the train_20% dataset. In table 5. 1, assumptions have been made of the new mapped attributes.

Table 5. 2 Assessment of new mapped attributes dependency by filter correlation measure in train+

Features	*fm1*	*fm2*	*fm3*	*fm4*	*fm5*
fm1	1	0.085	0.013	-0.001	-0.01
fm2	0.085	1	0.0036	-9e0.05	-0.00092
fm3	0.013	0.0036	1	-0.00035	-0.0036
fm4	-0.01	-9e.05	-0.00035	1	0.00081
fm5	-0.01	-0.00092	-0.0036	-0.00081	1

Table 5. 2, shows the co-relation coefficients which have been derived from the train+ dataset after performing the filter method.

5.2 NSL-KDD Classes and Features

Table 5. 3 NSL-KDD Dataset Normal and malicious feature vector similarity

Features Vector	Label
0, udp, other, SF, 146, 0, 0, 0, 0, 0, 0, 0.08, 0.15, 0, 255, 1, 0, 0.6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 13, 1, 0, 0, 0, 0, 0.88, 0, 0, 0, 0, 0	Normal
0, tcp, private, SO, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 166, 9, 1, 1, 0, 0, 0.05, 0.06, 0, 255, 9, 0.04, 0.05, 0, 0, 1, 1, 0, 0	U2R

Table 5. 3 shows the feature values of the NSL-KDD dataset. Difference in features in between normal and anomalous attacks has been shown here.

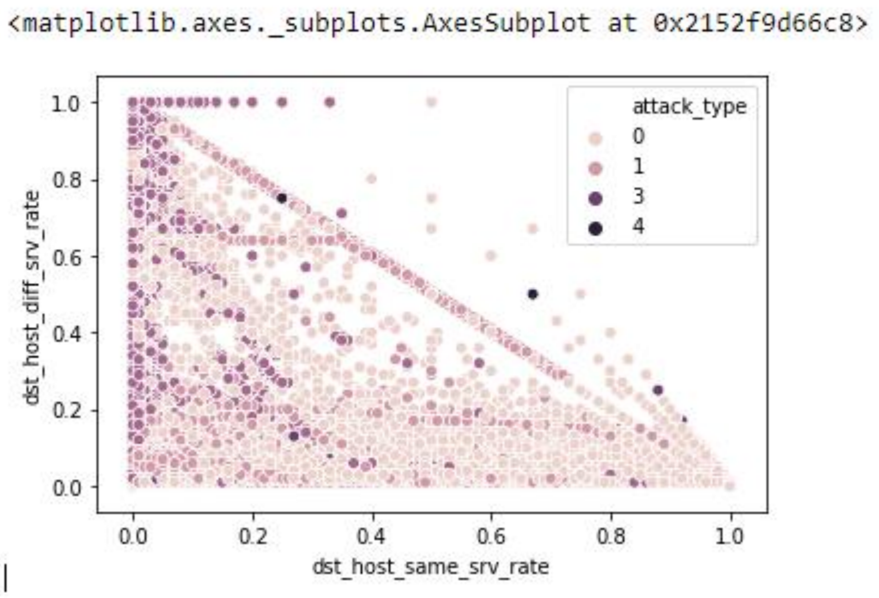


Figure 5. 2 Generated image of the total number of attacks in the test+ dataset

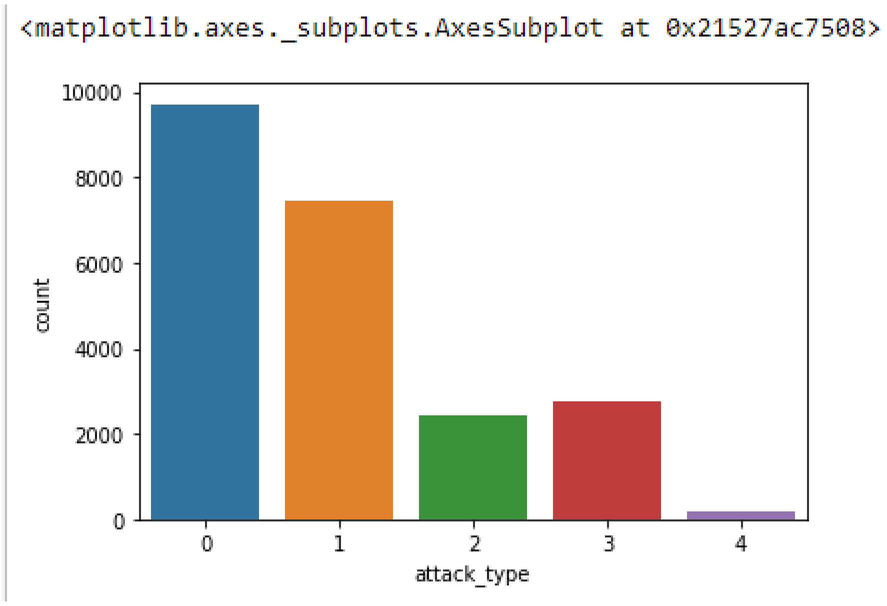
Figure 5. 2 shows the different attack types that are present in the test+ dataset. It also reflects on the total number of the specific attack types. Here 0 stands for DoS, 1 for Probe, 3 for U2R and 4 for R2L. The generated image is the visible explanation of the attack types.

As Fig 5. 4 represents the disintegrated rates of mapped test set which is not very much different from the training set. Another important issue that insinuates from this figure is that the rare and dangerous attack type like u2R and R2L are so involved with normal behaviors

Table 5. 4 NSL-KDD Data Classes Distribution

Dataset	Total Records	Normal	Probe	Dos	U2R	R2L
Train_20%	25192	13449	2289	9234	11	209
Train+	125973	67343	11656	45927	52	995
Test+	22544	9711	2421	7458	67	2887

Table 5. 4 represents the number of attacks in the whole dataset. It is visible that the biggest number of attacks are present in the test data set and the train_20% possesses the lowest number of attacks.



```
<matplotlib.axes._subplots.AxesSubplot at 0x21527ac7508>
```

Fig 5. 3 Generated diagram of the number of attacks for train+ dataset

From table 5. 4, it was established that train+ possesses the largest number of attacks. Fig 5. 3 is the graphical representation of the number of attacks in train+ dataset. Here, 0 stand for normal, 1 for DoS, 2 for Probe, 3 for U2R and 4 for R2L.

5.3 Comparison Between the Proposed Model and the Existing Models

To evaluate the proposed model, the proposed method has been implemented in Python. The proposed model works in two layers using two different machine learning classifiers.

Table 5. 5 Comparison between anomalous instances detection rate% of the first level and the refined level of classifications

Level	Probe	DoS	U2R	R2L
First Level of Classification	98	97	11	9
Refined Level of Classification	88	95	69	54

In table 5. 5, it can be seen that after applying SVM in the first layer, the gained detection rate of the common attack types such as DoS and Probe has been satisfying. But the detection rate of the rare ones has been poor. In the second layer the results have changed drastically. After applying Random forest classifier, the detection rates of all the attack types have been satisfactory.

Table 5. 6 Multiclass Classification Detection Rates% comparison to existing Models

Method	Normal	Probe	DoS	U2R	R2L
Proposed Model	99	88	95	69	54
Two-Tier Anomaly Detection Model	94.56	79.76	84.68	67.16	34.81

In table 5. 6, a comparison has been shown in between the proposed model and a similar co existing model. The latter one consists of the same type of methodology and steps but with different dimensionality reduction approach and classifiers. They have used LDA for feature selection. The classifier that they had used are Naïve Bayes and KNN- Classifier. The obtained result was satisfactory as the detection rate of rare type of attacks are pretty high. The only drawback was that the study lacked in gaining a higher detection rate against normal attacks. But it can be seen in the proposed model that by using different classifiers and dimension reduction techniques, it is possible to gain a higher detection rate against both common and rare type of attacks.

Table 5. 7 Confusion matrix of existing models which had low false alarm and undesirable detection rate (%) against the rare attacks versus proposed model

Model	Normal	Probe	DoS	U2R	R2L
The proposed model	99	88	95	69	54
SVM with BIRCH clustering [11]	99.3	99.5	97.5	28.8	19.7
SVM with Random Forest [20]	92.99	63.81	96.32	34.48	45.1
Two Tier with Naïve Bayes and KNN [4]	94.56	79.76	84.68	67.16	34.81

A comparison in between the co-existing IDS models and the proposed model has been shown in table 5. 7. It is visible that the proposed model has better detection rate against all of the existing ones.

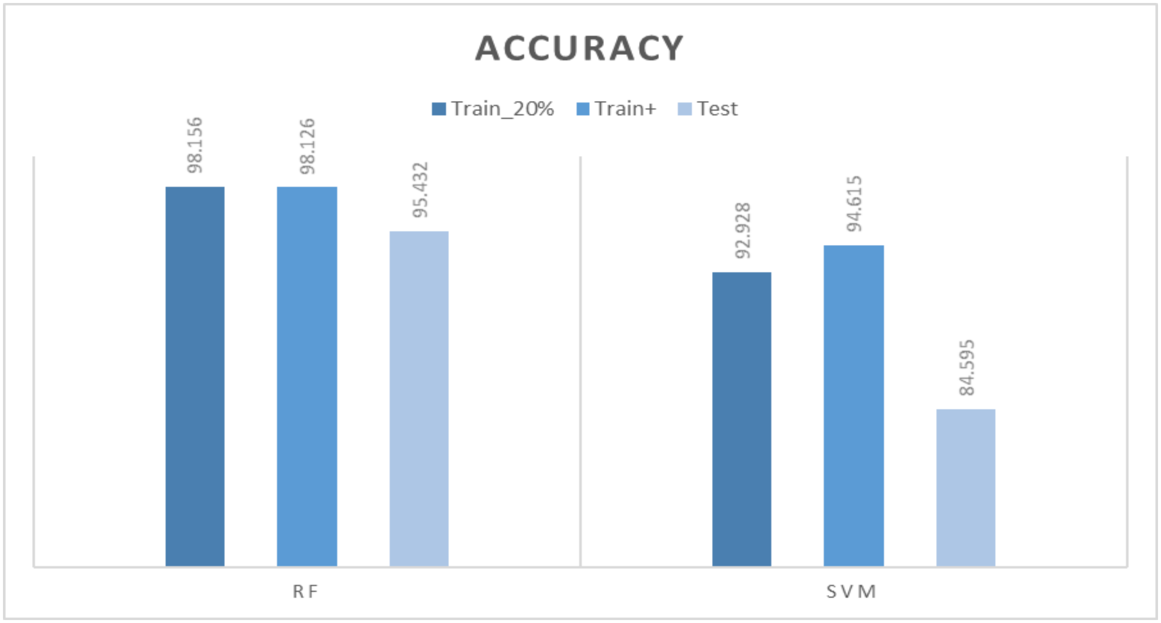


Fig 5. 4 Bar diagram for the accuracy of the proposed model

Fig 5. 4 shows the accuracy of the proposed model for test+, train+ and train_20% dataset. The results show that in the second layer where Random forest was applied, the accuracy increased in comparison with the first layer. In the first layer, SVM was applied and the accuracy for the test data was 84.595%. but after applying random forest, accuracy increase to 95.432%.

Table 5. 8 Single-layer Vs. multi-layer binary classification comparison (%) result evaluated by Test+

Method	Training set	detection rate	False alarm rate
Proposed model	Train 20 %	93.89	0.237
Na¨ıve Bayes (Tavallaee et al. 2009)	Train 20 %	76.56	N/A
Random forest (Tavallaee et al. 2009)	Train 20 %	80.67	N/A
SVM (Tavallaee et al. 2009)	Train 20 %	69.52	N/A
Decision trees (J48) (Tavallaee et al. 2009)	Train 20 %	81.05	N/A
Proposed model	Train$^+$	82	5.43
SOM IDS (Ibrahim et al. 2013)	Train$^+$	75.49	N/A
Feature selection with SVM IDS (Pervez and Md Farid 2014)	Train$^+$	82	15

In table 5. 8, a comparison has been shown between the proposed model and the existing one layered models. It can be seen that the overall detection rate of the proposed two layered model is far better than the existing one layered models. The false alarm rate is also comparatively better in the proposed model making it even more desirable.

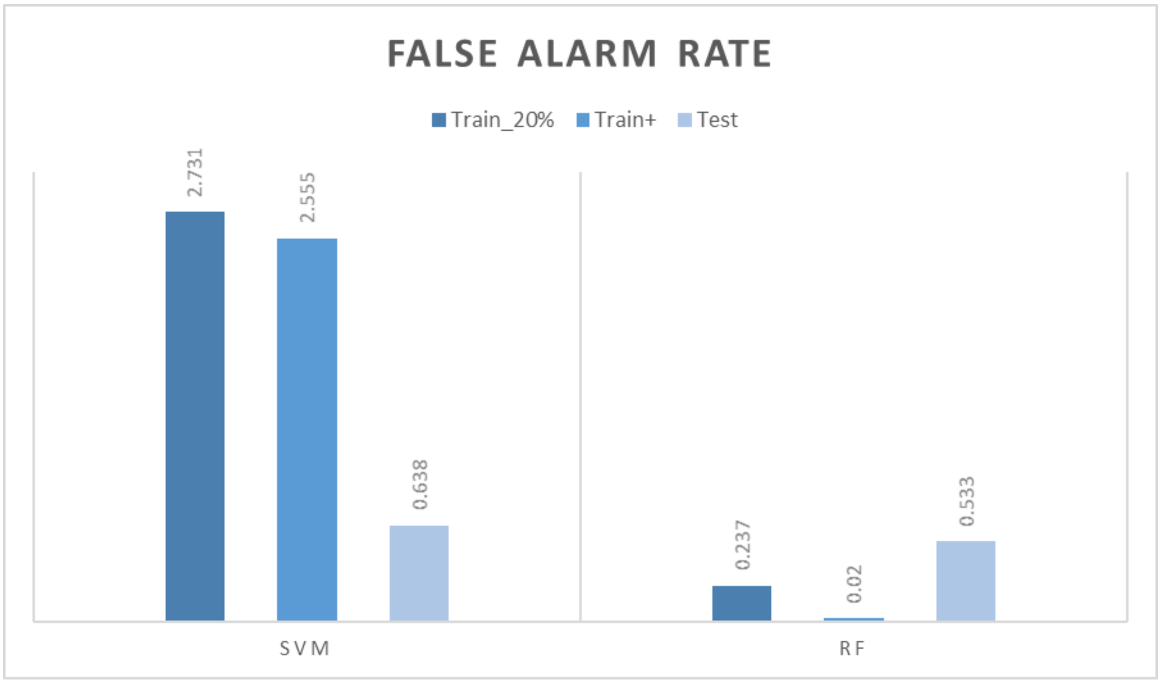


Fig 5. 5 False Alarm Rate difference in the two layers

Analyzing Fig 5. 5, it can be seen that in the first layer (after applying SVM as the classifier), the false alarm rate was less than the coexisting models. But in the second layer the rate decreased even more making the proposed model more efficient to use. In the second layer, the proposed model obtained 0.237 % false alarm using Train_20 % dataset and 0.0204 % in using Train+ as training set and 0.533% in using Test+ respectively for testing the proposed model.

Table 5. 9 41 features of NSL-KDD Dataset

Number	Feature	Type of feature	Number	Feature	Type of feature
1	Duration	numeric	22	Is_guest_login	nominal
2	Protocol_type	nominal	23	Count	numeric
3	Service	nominal	24	Srv_count	numeric
4	Flag	nominal	25	Serror_rate	numeric
5	Src_bytes	numeric	26	Srv_serror_rate	numeric
6	Dst_bytes	numeric	27	Rerror_rate	numeric
7	Land	nominal	28	Srv_rerror_rate	numeric
8	Wrong_fragment	numeric	29	Same_srv_rate	numeric
9	Urgent	numeric	30	Diff_srv_rate	numeric
10	Hot	numeric	31	Srv_diff_host_rate	numeric
11	Num_failed_logins	numeric	32	Dst_host_count	numeric
12	Logged_in	nominal	33	Dst_host_srv_count	numeric
13	Num_compromised	numeric	34	Dst_host_same_srv_rate	numeric
14	Root_shell	numeric	35	Dst_host_diff_srv_rate	numeric
15	Su_attempted	numeric	36	Dst_host_same_src_port_rate	numeric
16	Num_root	numeric	37	Dst_host_srv_diff_host_rate	numeric
17	Num_file_creations	numeric	38	Dst_host_serror_rate	numeric
18	Num_shells	numeric	39	Dst_host_srv_serror_rate	numeric
19	Num_access_files	numeric	40	Dst_host_rerror_rate	numeric
20	Num_outbound_cmds	numeric	41	Dst_host_srv_rerror_rate	numeric
21	Is_host_login	nominal			

Table 5. 9 shows the features that has been used in the proposed model. The NSL-KDD dataset consists of 41 features. Amongst them, five have been used for each iteration after implementing the feature selection procedure.

5.4 Discussion

After analyzing the derived detection and false alarm rates it is safe to state that other works that had been done prior to this were either efficient in detecting common and seen attacks or the rare attacks. Even the multiple layered works are prone to detecting either common or rare ones but not the both. So, designing an IDS capable of detecting all sorts of attacks were necessary. And it could be seen that changing feature selection method and altering algorithms improve the system to a great extent.

Using dimensionality reduction approach reduces the features thus reduces the complexity. And less complexity makes a system more cost efficient. Adding filter co-relation method to the proposed model reduced its complexity and thus was able to reduce its computational cost.

The classifiers used in the two layers of the proposed model are SVM and Random Forest. Both of the algorithms have comparatively lower complexity which adds to system having low computational cost.

False alarms could be regarded as a great disadvantage to any intrusion detection system. So, getting a low false alarm rate is really important in an IDS. The proposed model has a comparatively low false alarm rate which adds as an advantage to the model.

The proposed model is capable of distinguishing between normal and anomalous behavior. It is also able to detect both common less dangerous attacks and unseen rare attacks which makes the proposed model efficient enough.

Accuracy is an important fact for any machine learning approach. Accuracy defines the working ability. In this study, it can be seen that the accuracy in the second layer for train_20% is 98.15 which is a decent percentage. So, by analyzing the accuracy, it could be said that the proposed model is dependable.

Overall, the detection rate of the proposed model is 93.89% which is better than most other coexisting models. The detection rate for the specific attack types are 99%, 95%, 88%, 69%, 54% respectively for Normal, DoS, Probe, U2R and R2L. it could be seen from the result section that the rates are higher than the others making the proposed model the better one.

5.5 Limitation

The proposed model was able to gain a higher detection rate in comparison with most of the coexisting models but the final output was not that much praiseworthy. Detection rate for the common types of attacks was less than some of the coexisting models which needs to be improved in the proposed model. Another limitation is that the proposed model was not able to use SMOTE techniques which could have helped to get better results.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this study, the writers have explored the field of intrusion detection system and proposed a two layered model that could work against common and rare attacks. The challenges of designing a working model cable of detecting attacks has been described in the study. The implementation was done by a machine learning parametric approach using various algorithms and techniques.

The proposed model trains and preprocesses data, selects features for dimensional reduction and the applies classifiers in two layers to classify the data. In the process detection rate, accuracy and false alarm rates are derived. After that the derived rates are compared with the other coexisting models to prove the worth of the proposed one.

Finally, it is safe to state that the aim of the proposed model has been fulfilled as it has higher detection rate for all types of attacks, lower false alarm rate and lower computational cost in comparison with the coexisting models. It seems to be an efficient solution for the organizations and industries to keep their server away from anomalous activities thus ensure their server safety. In near future it might be possible to get even better result by applying better techniques, but at present it seems to be a coherent solution for designing an effective intrusion detection system (IDS).

6.2 Future Work

In the near future, the aim of the proposed model is to use non parametric techniques to obtain more useful features. Another future work is to use fuzzy clustering techniques to improve the detection rate by separating the normal instances from the anomalous ones.

BIBLIOGRAPHY

[1] D. E. Denning, “An Intrusion-Detection Model,” *IEEE Trans. Softw. Eng.*, 1987, doi: 10.1109/TSE.1987.232894.

[2] S. Axelsson, “Base-rate fallacy and its implications for the difficulty of intrusion detection,” 1999, doi: 10.1145/319709.319710.

[3] Z. Tan, A. Jamdagni, X. He, and P. Nanda, “Network intrusion detection based on LDA for payload feature selection,” *2010 IEEE Globecom Work. GC’10*, pp. 1545–1549, 2010, doi: 10.1109/GLOCOMW.2010.5700198.

[4] H. H. Pajouh, G. H. Dastghaibyfard, and S. Hashemi, “Two-tier network anomaly detection model: a machine learning approach,” *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, 2017, doi: 10.1007/s10844-015-0388-x.

[5] K. Leung and C. Leckie, “Unsupervised anomaly detection in network intrusion detection using clusters,” *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 38, no. January, pp. 333–342, 2005.

[6] P. K. Chan, M. V. Mahoney, and M. H. Arshad, “Learning Rules and Clusters for Anomaly Detection in Network Traffic,” *Manag. Cyber Threat.*, pp. 81–99, 2005, doi: 10.1007/0-387-24230-9_3.

[7] J. Zhang and M. Zulkernine, “Anomaly based network intrusion detection with unsupervised outlier detection,” *IEEE Int. Conf. Commun.*, vol. 5, no. c, pp. 2388–2393, 2006, doi: 10.1109/ICC.2006.255127.

[8] A. N. Toosi and M. Kahani, “A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers,” *Comput. Commun.*, vol. 30, no. 10, pp. 2201–2212, 2007, doi: 10.1016/j.comcom.2007.05.002.

[9] H. Lu and J. Xu, “Three-level hybrid intrusion detection system,” *Proc. - 2009 Int. Conf. Inf. Eng. Comput. Sci. ICIECS 2009*, 2009, doi: 10.1109/ICIECS.2009.5366474.

[10] M. Panda, A. Abraham, and M. R. Patra, “Discriminative multinomial naïve bayes for network intrusion detection,” 2010, doi: 10.1109/ISIAS.2010.5604193.

[11] S. J. Horng *et al.*, “A novel intrusion detection system based on hierarchical clustering and support vector machines,” *Expert Syst. Appl.*, vol. 38, no. 1, pp. 306–313, 2011, doi: 10.1016/j.eswa.2010.06.066.

[12] L. Breiman, “Random forests,” *Mach. Learn.*, 2001, doi: 10.1023/A:1010933404324.

[13] P. Krömer, J. Platoš, V. Snášel, and A. Abraham, “Fuzzy classification by evolutionary algorithms,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 313–318, 2011, doi: 10.1109/ICSMC.2011.6083684.

[14] T. Idé and H. Kashima, “Eigenspace-based anomaly detection in computer systems,” 2004, doi: 10.1145/1014052.1014102.

[15] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*. 2016.

[16] V. Shmatikov and M. H. Wang, “Security against probe-response attacks in collaborative intrusion detection,” *Proc. 2007 Work. Large Scale Attack Defense, LSAD ’07*, pp. 129–136, 2007, doi: 10.1145/1352664.1352673.

[17] A. Alharbi, “Denial-of-Service , Probing , User to Root (U2R) & Remote to User (R2L) Attack Detection using Hidden Markov Models,” vol. 07, no. 05, pp. 204–210, 2018.

[18] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” 2009, doi: 10.1109/CISDA.2009.5356528.

[19] J. Mchugh, “Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory,” *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 4, pp. 262–294, 2000, doi: 10.1145/382912.382923.

[20] M. A. M. Hasan, M. Nasser, S. Ahmad, and K. I. Molla, “Feature Selection for Intrusion Detection Using Random Forest,” *J. Inf. Secur.*, vol. 07, no. 03, pp. 129–140, 2016, doi: 10.4236/jis.2016.73009.

[21] M. A. M. Hasan, M. Nasser, B. Pal, and S. Ahmad, “Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS),” *J. Intell. Learn. Syst. Appl.*, vol. 06, no. 01, pp. 45–52, 2014, doi: 10.4236/jilsa.2014.61005.

[22] V. Svetnik, A. Liaw, C. Tong, J. Christopher Culberson, R. P. Sheridan, and B. P. Feuston, “Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling,” *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, 2003, doi: 10.1021/ci034160g.

[23] S. Y. Jiang and L. X. Wang, “Efficient feature selection based on correlation measure between continuous and discrete features,” *Inf. Process. Lett.*, vol. 116, no. 2, pp. 203–215, 2016, doi: 10.1016/j.ipl.2015.07.005.

[24] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, “Technique - ML - A Practical Guide to Support Vector Classification,” no. 1, pp. 1–16, 2010.

[25] N. Farnaaz and M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System,” *Procedia Comput. Sci.*, vol. 89, pp. 213–217, 2016, doi: 10.1016/j.procs.2016.06.047.

APPENDIX A

LIST OF ACRONYMS

DoS	Denial of Service
U2R	User to Root
R2L	Root to Local
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
NIDS	Network-Based Intrusion Detection System
HIDS	Host-Based Delivery Network
RF	Random Forest
SVM	Support Vector Machine
FM	Filter Method
FSS	Feature Selection