Intrusion Detection System using Machine Learning Algorithm

Submitted in partial fulfillment for the award of the degree of

Masters of Computer Applications

By

ABHIJEET GIRI
18MCA0088

Under the guidance of
Dr. Brindha K

School of Information Technology and Engineering
(SITE)
VIT, Vellore.

May, 2020.

EXECUTIVE SUMMARY

Now a days, it is very hard to prevent security breaches using current technologies. In this manner, the outcome is that Intrusion Detection turns into a significant issue in security of system and computer forensics. To ensure that communication of information is safe, various systems for detecting intrusions are developed that may have several restrictions in intrusion detection. The rules of encoding intrusion is very time taking and also conditional upon the idea of similar intrusions. Detecting attacks and preventing computers through these is a leading topic for research among researchers in this era. In this project we will implement and will highlight Intrusion Detection on the attacks mentioned above. Threats to these types are different and are potentially devastating. Till now, different procedures have been proposed by the scientists for the IDS, some of which are AI, DNA arrangement, Pattern coordinating, data mining are used as expertise to learn about the attack and its different types and even happen with some matching attacks types when its came across us in the future. In this project three type of machine learning algorithms RF, KNN, and SVM are used. Here, we have classified the problem based on features that have been selected as a parameter for evaluation. We have likewise utilized model assessment and determination strategies like accuracy, precision, recall, f-score along with Confusion Matrix which is a device utilized for assessing performance. And as a result we have shown the accuracy of our method for each type of attack.

LIST OF ABBREVIATIONS

IDS	Intrusion Detection System
DOS	Denial of Service
U2R	User to Root
R2L	Remote to Local
SVM	Support Vector Machine
RF	Random Forest
KNN	K- Nearest Neighbors
DM	Data Mining
RFE	Recursive Feature Elimination
IPS	Intrusion Prevention System
CM	Confusion Matrix

1. INTRODUCTION

1.1. Objective

With the immense extension in services of network-related areas and transmission of private information over these unsecured networks makes though task for providing network security. Except for this large number of technologies used for protection as like as encrypting information, access control, and preventing access of intrusion are widely used to secure neuromorphic systems, except these, still there are large number of untraceable intrusions. IDS is widely used to monitor network activities in automatic way plays very important character in security of network. An IDS has a place with a system that saves eyes on arrange traffic for recognizing farfetched exercises and alarms the system when any dubious exercises are watched. Though distinguishing anomaly and enumerating is one of the underlying stages, few IDS can initialize steps for detecting abnormal traffic and malicious activity when they are in action, which includes traffic blockage that arrived from a doubtful IP address. Despite that IDS authenticates networks from harmful activities, and it is also liable to fake alarms or fake activities. As a result, a firm needs to calibrate IDS output while installing. Properly configuring IDS for recognizing for that cause networking usual congestion seems very comparable for possible harmful activities. An Intrusion Prevention System (IPS) always recognizes network packets for that cause of harming network congestion. Although an "Intrusion Detection System" reacts to likely harmful congestion by blocking the congestion and releasing alert notifications, "Intrusion Prevention Systems" reacts to congestion by refusing the likely harmful packets. At present, an enormous number of rule-based IDSs systems are highly relying on carrying out rules marked by experts in the security fields. Since, the amount of network congestion is enormous, the encoding command are expensive as well as slow. However, the rules are modified by security people or they have deployed currently developed rules used a particular command-driven script. To get over the disadvantages of rules-based systems, a figure of IDSs implements Machine learning algorithm. Machine learning is the way toward breaking down of large datasets to find out the pattern and models to derive related information from it. We can efficiently get patterns or information's from dataset using Machine learning algorithm, to find out general network activities profiles to detect anomaly, and to make model for classifiers for detecting malicious network attacks. To make a system more flexible and deployable.

1.2 Motivation

Developing an IDS system with absolutely secure and with less false alarm is not possible with the use of the existing system available because the system can't handle the broad set of data and only handle limited types of attacks. In the previously used system pattern recognition method is mostly used for IDS, but it takes a lot of time to recognition of intruder and most of the time its fails to detect the intruder. In this project some of the machine learning algorithms are used to detect the IDS with high accuracy of detection of an intruder with less time, and the proposed system detects many types of attacks that are divided into four subcategories.

1.3 Background

1.3.1 Related Work

- **Unsupervised learning approach for network intrusion detection system (IDS) using auto encoders.**

The author deal with unsupervised machine learning algorithm for intrusion detection previous many paper was developed in many algorithms but in this paper break all the accuracy of previous algorithms. The author compare this paper with cluster based intrusion detection in cluster based algorithm gives the accuracy 80% but this paper gives the accuracy of 91.70% that is very big difference of accuracy label so this model gives more accurate result then previous. In this paper NSL-KDD dataset was used, 85% of training dataset and 15% of test dataset was used in this paper. The author was used autoencoders algorithm in this paper that is an unsupervised machine learning algorithm.

- **Multi layer intrusion detection system with Extra Trees feature selection,extreme learning machine, and softmax aggregation.**

The author uses ExtraTreeClassifier for selecting the effective features of each types of attack and also ELM was used for detect each type of attack separately. After that they uses the ELM output is used to combine with softmax layer refines the result for more accuracy.In this paper the ExtraTreeClassifier was used to select the features the classifier gives the rank and score to each features. The author uses the model for two types if dataset.

- **Combining Best Features Selection Using Three Classifiers in Intrusion Detection System.**

The author uses the feature selection method that removes the unrelated attributes to increase the performance of the classification algorithms to increasing the accuracy. In this paper there are some different method are used which are gain ratio, information gain, and correlation. These techniques are used for selecting and ranking the attributes only six attributes are selected out of 41 attributes. The features where tested on some different classifiers they are Naïve bays, KNN, and neural network. The naïve bays classifier give less accuracy other than two classifiers KNN gives higher accuracy.

- **Performance enhancement of deep neural network using feature selection and preprocessing for intrusion detection.**

The author deal with network intrusion detection system using deep neural network to get the higher accuracy rate using feature selection and layer configuration method. In this paper the author uses the NSL-KDD dataset. Pearson correlation is used for selecting the attribute by selecting from total attributes the main motive of attributes selection is remove the less effective features and avoid over fitting. In this paper the author conclude that many layer configurations only extended the learning time and did not provide high accuracy due to overfitting.

- **A Novel Hierarchical Intrusion Detection System based on Decision Tree and Rules-based Models.**

The author deal with a novel intrusion detection system by using some different classifier methodologies which are based on decision tree and rule bases concept. They uses three classifier that are REP tree, JRip algorithm and Forest PA. The first two classifier take features as a input and the third gives the output. The evaluation is made on real time data set CICIDS2017. The hierarchical model gives the maximum DR for 7 types of attacks that is 94.45% and highest accuracy with 96.66%.

- **RF-Based Network IDS System.**

In this paper summarize three frameworks that are based on the data mining frameworks for the detection of intrusion in the network. In this paper, the random forest algorithm is used for anomaly or outliers, hybrid, and misuse detection. In this paper, the author implemented the random forest algorithm to generate the patterns of the intrusion that can easily address the problem of rule-based systems. With the help of training data, the RF method automatically produces the examples rather than manual rules. The method is proposed in this paper, are doing on the JAVA by using the WEKA condition and the Fortan program. The creator likewise assessed the usage over various datasets which are acquired from the dataset of KDDCUP99 and the outcomes which are gotten from the directed test show that the methodologies utilized in this paper performed better than the KDDCUP99 results.

The author thread detection architecture, the instructions of patterns are generated in the disconnected stage, and the system detects the intrusions naturally in the actual time by using the generated design. For the upgrading of the accuracy of the methodology, the author uses the attribute-selection methodology and optimizes the variables of the RF algorithm. Some easier methods are also used in this paper so that the figure of minority intrusions that are in the substructure can be increased. Since the observation of misuse cannot observe the novel intrusions, a new methodology was proposed by the user in unsupervised anomaly detection.

- **DOS, Probe &Remote to User (R2L) Attack Detection using Genetic Algorithm.**

Artificial Intelligence methods are getting more importance at present era due to its ability of acquiring knowledge and it's progress, that make this more accurate and effective to look out on the large number of untraceable attacks. Genetic Algorithm methodology is used for detection of Denial of Service, Remote and Probing is proposed on user attacks. The primary aim of the proposed approach is to maximize the detections of DoS, R2L and Probing attack to acquire minimum false or fake positive rate. From all intrusion types found in testing dataset, it is expected that more than 97% of the intrusion is detected by using this approach. Thus we can easily say that this way must be useful for the detection of attack in present era where attack methodologies are changing every day. By dynamically updating the rules of firewall log data, thus we can effectively use this method against new types of attacks.

- **Random Forest Modeling for Network Intrusion Detection System.**

The Random Forest (RF) algorithm is implemented by the author for the detection of four different kind of attacks are DOS, probe, user to root and R2L. Author takes on 10 different cross validations proposed on classification. To reduce dimensionality, feature selection method is implemented on the dataset and also implemented to reduce unnecessary and inapplicable features. To overcome the problems arise due to information gain, author implemented symmetrical uncertainty of attributes. NSL KDD data set is used to evaluate proposed approach. Author also compared Modeling of Random forest is compared by author with j48 classier with regard to accuracy, DR, FAR and MCC. Author's proposed model shows that accuracy, MCB and DR for four different attacks are increased under their experimental results. In respect of future work, author's suppose to implement evolutionary computation to improve accuracy of the classifier for selecting features of the intrusions.

- **Anomaly Intrusion Detection System using Random Forests and k-Nearest Neighbor.**

The author has presented a data-mining techniques survey that had proposed for the anomaly intrusion detection systems enhancement. In this paper, for classifying the intrusion, the author implies classification methods on the DARPA dataset. The results which are obtained by the author shows the performance of the Random Forest Classifier is better than other classifiers. But for the Random Forest classifier, the execution time is more than other classifiers.

- **Feature Selection for Intrusion Detection Using Random Forest.**

A Random Forest model for Intrusion Detection System (IDS) has been introduce which aims for improving the performance of Intrusion detection by reducing the attributes of the input. In the circumstances of the real-world applications, attributes that are in smaller numbers are constantly advanced in terms of both the data management and growing time.

The values which are obtained show the ability of the Random Forest classifier with eliminated characteristics (25 attributes) that produces results more accurate comparing from the Random Forest classifier that has all (41 attributes). Further, time is taken to process 25 attributes with Random Forest is less than the developing time of Random Forest has 41 attributes. Researches in attributes selection and intrusion detection with the help of the Random Forest classifier is still an ongoing process regarding its good performance. The author says that the findings in this paper will be very useful in the field of research of attribute selection and classifier. Those things can also be applied to use Random Forest in a meaningful way so that the performance rate can be maximized and the false positive rate can be minimized.

- **Random Forest Algorithm in Intrusion Detection System: A Survey**.

The author explains Random Forest Algorithm, the number of trees in the forest and the results from them are directly related, i.e. the more trees, the more accurate the result. It is important to note that, decision-making using the gain or gain approach is not the same as creating a random forest. This paper presented an overview of the random forest algorithm and a survey of various techniques proposed by several researchers has also been summarized.

- **An Investigation on Intrusion Detection System using machine Learning.**

The author deal with NSL-KDD dataset was used to check performance of machine learning algorithms for the intrusion detection system. In this paper, the author has used a feature section method for better accuracy because all the features are not so much affective on the construction of model. Hence, the author eliminates this type of feature and compare it without elimination and after elimination. Two models are used for prediction of intrusion detection that is Support Vector Machine (SVM) and Random Forest(RF). Random Forest gives good performance than SVM before features elimination, but after feature elimination SVM provides better performance.

- **Differential Evolution Wrapper Feature Selection for Intrusion Detection System.**

In this paper, the Wrapper feature selection method used with a differential evolution technique for the intrusion detection system. This method helps the author to find minimum features by eliminating the features that are not affected by the performance of the model. In this proposed system, NSL-KDD data set was used and gave an accuracy of 87% and 80% using binary classes and five classes. The proposed method was done in the weka tool. Four steps are done in the hole System, first step preparing NSL-Dataset then feature selection using DE algorithm after that ELM classifier was used then the final step was done that is performance evaluation.

- **Attribute Selection and Ensemble Classifier based Novel Approach to Intrusion Detection System.**

The author deal with ranker-based attribute evaluation technique used for reducing the number of attributes after that they evaluate the model using IBk(K-NN), Random Tree, REP Tree, j48graft, and Random Forest classifier. The model uses NSL-KDD data set for performing the model. In this proposed system, the author uses different classifiers that is Function classifier, Bayes classifier, Lazy classifier, Meta classifier, Rules classifier, and Trees classifier. The dataset contains 39 different types of attacks that are divided into four types DOS, Probe, U2R, and R2L, there are ten types of DOS attacks, six types of Probe, sixteen types of R2L, and seven types of U2R. The system gives an accuracy of 99%.

- **Network Intrusion Detection System using Data Mining Techniques.**

In this paper, the author proposed an intelligent intrusion detection system using the AODE methodology used for the finding of different types of attacks. To checking the performance of the system, the author uses NSL-KDD dataset with low FAR and high DR. In this proposed system, the author divided the data set into four types of attack that DOS, Probe, U2R, and R2L. After that, ten cross-validations is applied for classification. They compare their proposed system with naive bays algorithm, and the accuracy was their system is 96%, and naive bays is 89%.

- **Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark.**

In this paper distributed random forest machine learning algorithm was used to detect intrusion from high-speed traffic data. They used a random forest classification methodology and adopted it to apache spark for real-time finding from the distributed processing system. The proposed system gives 97.8% accuracy.

- **Feature Reduction in Flow-Based Intrusion Detection System.**

In this paper, the author uses an effective features reduction method for the intrusion detection system. The technique extracts performance-based features with a machine learning algorithm. They used the JRip rule-based classification to evaluate the performance of the elements, and they also used 10-fold cross-validation. The proposed system uses the label flow-based CICIDS2017 dataset. The dataset having 85 attributes, and the algorithm selects the most relevant feature that is only 18 features.

2. PROJECT DESCRIPTION AND GOALS

2.1 Problem Definition

To distinguish the exercises of the system traffic, the interruption and solid are troublesome and require a lot of tedious. An analyst must check all the data that broad and extensive to find the sequence of intrusion on the network connection is very difficult. Accordingly, it needs a way that can distinguish organize interruption to mirror the present system deals. In this venture, three methods are used to discover intrusion characteristics for IDS using random forest, support vector machine, and k-nearest neighbor's machine learning algorithms proposed. The technique used to generate rules by classification using the above algorithms. These rules can decide intrusion characteristics than to execute in the firewall arrangement rules as a counteraction.

2.2 Existing System

There are many variety of existing system is available for intrusion detection but mostly used system are signature-based system and anomaly-based system. The signature-based system is work for that types of threads or attacks that are already known by the system that can be detected, but if new types attacks are arrives then it's can't be detected by the system. The system is detect the threads using searching the series of pattern that takes a lots of time. The more advance system you want to developed using signature based you need more CPU load. On the other hand the anomaly based system works with system behavior its checks all arrival connection and its behavior with baseline that is known as accepted behavior of network. In this system the most disadvantage is every time they have to define the rules of accepted networks and analyze the accuracy and its need more hardware to stabiles the system.

2.3 Proposed System

An IDS recognition system screen and network traffic for unauthorized behavior and issues alert. In intrusion detection system helps us to monitor the unauthorized activities or any types of rules violation if this type of situation happened then they report it to the system administrator. There are many types of IDS is present that is network-based IDS, host-based IDS, Perimeter

IDS and VM-based IDS are present. In this project, I proposed a network-based IDS using the KDDCUP99 dataset. The dataset has 42 attributes. To evaluate the dataset, various types of algorithms of machine learning are used. The first step of the project is data processing in this step One-Hot-Encoding method is used to convert categorical features to form a matrix of categorical features along with binary values. Then split the data set into different types of attacks that are DOS, Probing, U2R, and R2L. After that features selection method used to eliminate the features using recursive feature elimination technique its select the most related 13 features. The final step is to build models in this project, three types of model are make that Random forest, Support vector machine, and KNN and evaluate the dataset using all features and using selected features.

2.4 Advantages of Proposed System

In this project IDS system was developed by Machine Learning algorithms using KDDCUP99 dataset. The proposed system can learn different types of attacks patterns of network to identify the intruder. The system can identify the new types of attacks, existing systems can't identify new attacks. The proposed system is analyses the data that is known as training data and predict the output using this its can produce the correct outcomes for labeled data. Its takes less times compare to existing system and the rate of false alarm is very less and its gives higher accuracy rate compare to other

2.5 Scope and Goal

The scope of the project is in the field of network security. The study of intrusion detection system is a very good topic for research and verity of direction can developed in further. Developing this types of system is a valuable contribution in the part of intrusion detection.

The goal of the project is to develop a system that’s detect the threads and attacks more accurately with less failures and detect new types of attacks. The project also should take less times and less CPU utilization for detection and it’s should also work for huge network traffic.

3. TECHNICAL SPECIFICATION

3.1 External Interface Requirement

3.1.1 Hardware Specification

- Processor requirement: Intel i5 6th generation and higher version with 2GHz.
- Hard Drive: 500 GB.
- Memory/RAM: 8 GB.
- Graphics Memory: 2 GB.

3.1.2 Software Specification

- Anaconda Navigator- All experiments are done in Python IDE called Anaconda.. Anaconda is a open source tool its having many data manipulation tool in it that are jupyter lab, jupyter notebook, R-programming and many more in this project the Spyder that is included in Anaconda are used for data analysis. Python 3.6 is used in this project

3.2 Machine Learning

Machine learning is the fastest-developing innovation of information science. The input of the teaching is training data representing experience. AI is an investigation of a computer algorithm that improve consequently through understanding. It construct a mathematical model using training data to make guess or decisions and tests the model using testing data. DM is the parallel types of study that falls under the same category. Machine learning is divide into two kinds of knowledge that is a supervised learning algorithm and an unsupervised learning algorithm. The supervised learning Algorithms investigates the training data to deliver learning,, which used for mapping a new example. Supervised learning helps us to solve two types of problem that is a classification and regression problem. Supervised learning having an input(x) and output(y) and using this input-output mapping the function y=f(x) after the mapping when you have new input data(x), you can predict output(y). When the output variable is categorical data like "red" or "green" or "diseases," then we have to use classification. When the output had a real value like "height" or weight," then we have to use a regression problem. The unsupervised learning used to solve clustering and association. When the data were having only input variable(x) and no parallel output variable, then we are used this algorithm.

When we want to discover to grouping the customer by purchasing behavior, then we are using the clustering problem on another side when we have to find the data like if the customer buys the product x then also buys y to discover this type of rule, we are using the association problem. The problem which falls between supervised learning and unsupervised learning is called semi-supervised learning. In this study, there is a huge no of input facts (x) and some of the facts labeled output(y). In this project many machine learning library are used that are:-

- **Scikit-learn-**

Scikit-learn additionally it called by sklearn is a free program or library of machine learning for python programming. Scikit-learn venture began on Google by Summer of code venture by David Cournapeau. It can work with various kinds of a calculation like order, relapse, and clustering calculation incorporate SVM, RF, and K-means Clustering, and so forth. It works with python numerical and logical library with Numpy and Scipy.

- **Pandas-**

Pandas is a data manipulation tool that was establish by wes Mckinneey. Panda is an open-source library that gives significant level data taking care of and examination devices. Pandas can understand three category of data structure that is series, data frame, and panel. Series is a 1 dimensional data labeled by a equivalent type of array. Its size was immutable. Data outline is a two-dimensional data; it has the size-variable plain structure with mostly heterogeneously composed sections. The panel is a three-dimensional data having a size-mutable array. In any case, utilizing a pandas data structure, it was very simple and all the more straight forward to work with these kinds of data. All pandas are values mutable values can be changed accept series. Pandas can be installed by installing Anaconda, a python package. Pandas are automatically to install by default using Anaconda.

- **Numpy-**

Numpy was establish by Jim Huginin. Numpy come with python packages, NumPy stands for Numerical Python. It is an free python library. Numpy is a group of multi-dimensional array object that procces the array in runtime.

Using NumPy, we can do the following operation that is mathematics operation and logical operations on the array, Fourier transformation, pattern for shape manipulation, and we can also do linear algebra using NumPy because NumPy has inbuild function for linear algebra. Numpy is utilized alongside packages like logical python and matplotlib. Numpy can be installed by installing Anaconda, a python package. Numpy is automatically to install by default using Anaconda. Numpy has a most significant item that is a N-dimensional exhibit type called ndarray.It depicts the assortment of information of a similar sort. Each thing of ndarray takes a similar size of block in the memory. Numpy support numerical type of data more than python that is bool, intc, uint32, complex, etc.

4. DESIGN APPROACHES AND DETAILS

4.1 Design Approach.

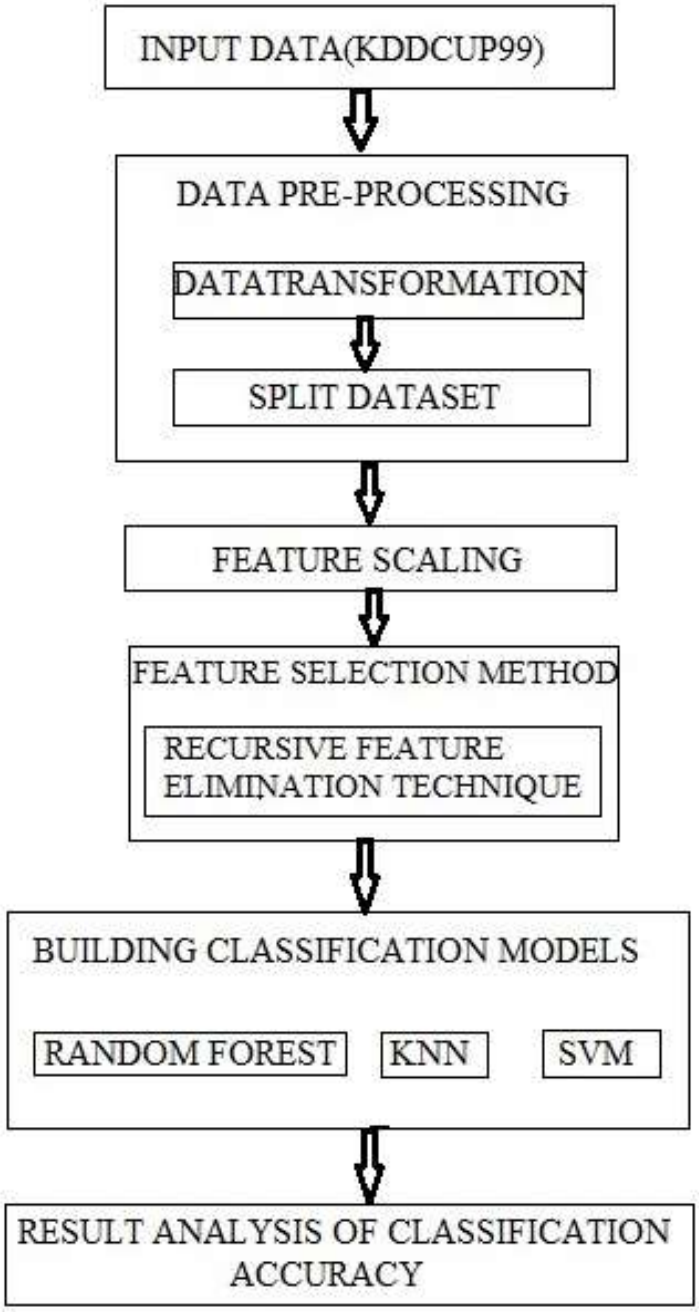


Fig4.1: Workflow diagram of proposed method.

- **Data preprocessing**: - In this step, One-Hot-Encoding method is used to convert categorical features to form a matrix of categorical features.in this dataset three columns or attributes are contains categorical data that is protocal_type, service and flags. Now, the features are transformed using LabelEncoder, to transform every category to a number. Three column are categorical in this dataset. We replace the label columns with new label columns. After that we divided the dataset into 5 sub categories each of different attack type. Here, we provide a dummy name to each attack type as 0=Normal, 1=DOS, 2=R2L, 3=U2R, 4=Probe.

- **Feature Scaling**:-It is the most important thing before trained the model with any dataset. In feature scaling technique used to standardized the independent features presented in data in a range. It is used because of handle the highly magnitude values. In this project we split the 4 attacks categories into x and y data frames that is done on the previous stage x is the data frame of features and y is the outcome variable.

- **Feature Selection**: - Feature selection step is correlated to data pre-processing step where irrelevant features are removed which increases the accuracy. Feature Selection refers to identification of features that are strongly correlated to problem which are useful in prediction of class. In this project one of the most important and effective feature selection technique is used that is recursive feature elimination technique it is falls under the wrapper feature selection categories. After using this technique we get the most effective 13-features with each attacks categories.

- **Building Model:-** To get a high accurate classifier which deals with real time data, a high performing model selection is required. In this project three most efficient machine learning algorithm is used for building model that is random forest (RF), KNN (K-Nearest Neighbors) and SVM (Support Vector Machine). Here, classifiers are trained for all features and most effective 13-features using trained dataset for trained the dataset i built two models for every algorithm one for all features and another for 13-features. To perform this model classification we used the predefined function of python called **RandomForestClassifier (), KNeighborsClassifier() and SVM()** classifier**.** Here, classification model is built for all types of attack.

- **Experiments** - All experiments are done in Python IDE called Anaconda using the Spyder that is included in Anaconda. Python 3.6 is used in this project. I used KDD'99 dataset for analysis which consist of 42 attributes where the last attribute having class label. To evaluate the performance of each classifier we construct confusion matrix to find accuracy, precision, recall and f- measure.

4.2 Module Description:-

- **Random Forest**: random forest algorithm is proposed in this paper as a new framework using data-mining technique in hybrid detection, misuse and anomaly detection. Classification and regression way of approach is used within random forest algorithm that act as an effective technique in data mining. Now a day's random forests algorithm are used for different applications. Random Forest algorithms are extensively used in probability estimation and prediction. It has been put in to forecast, and. However, the methodology has not been applied in naturally intrusion detection. In the system we are proposing, the component used for misused are the random forests algorithms for classifying intrusion detection, while the exception component is based on the outlier detection mechanism of the algorithm. Multiple decision trees are built under Random forest and it combines all of them together to generate an accurate and stable prediction.

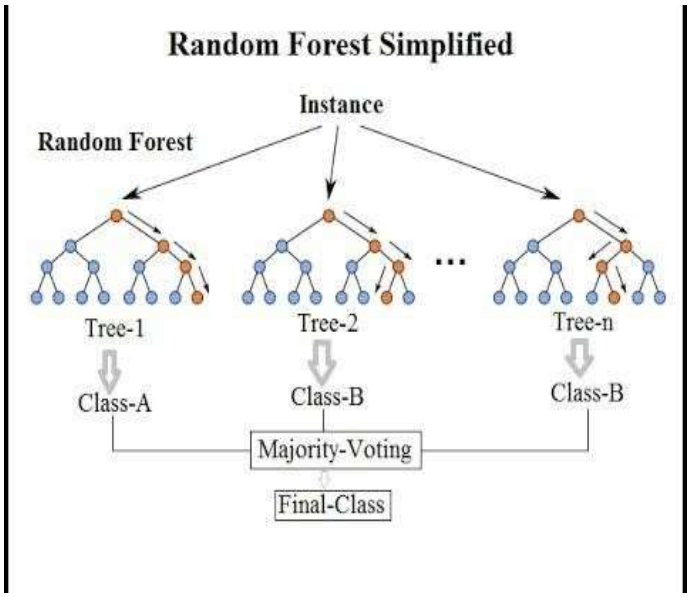


Fig 4.2:-Showing Random Forest in architecture.

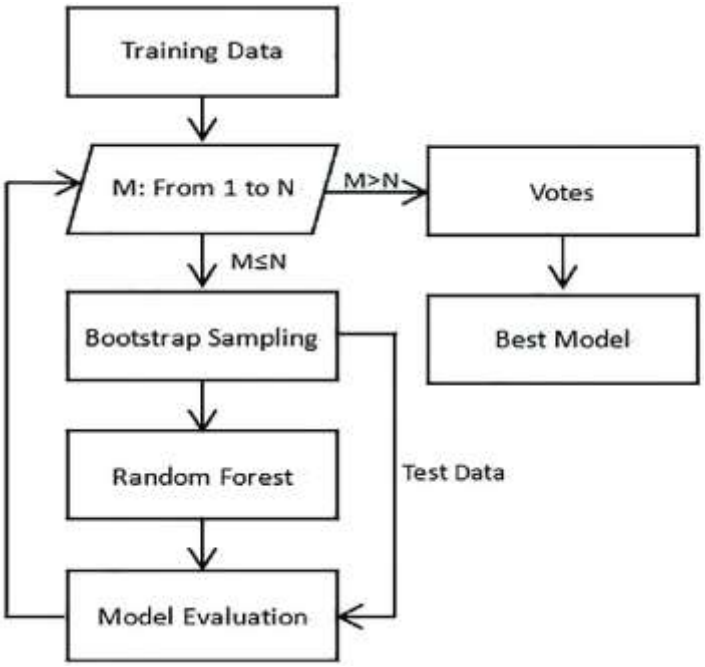


Fig 4.3: Workflow diagram of Random Forest

Advantage:

- To take care of grouping and relapse issues, random forest calculation is consistently the primary decision.
- Random forests are incredibly adaptable and have extremely high precision
- Random forests have very smaller fluctuation than a single decision tree. It proved that it works correctly for a huge range of data items than single decision trees.
- Random forest do not want produce of the input data. You don't need to scale the information.
- Even we lose a large set of data, this algorithm maintains accuracy.

Disadvantage:

- Intricacy is one of the primary impediment of Random forests calculation. Construction of random forest is much harder, complex and time consuming than decision tree.
- Requirement of computation is more in random forest algorithm. It always hard to get an instinctive grip of bonding present in input data when we get a large collection of decision tree.
- Predicting intrusion by using random forests methodology is always takes more time-taking than other algorithms.

- **K-Nearest Neighbors**:- K nearest neighbor is defined as a simple supervised algorithm which can train itself with all the given cases provided by the supervisor and then classify it to a new test case on previous features based on similarity measure (e.g., distance functions such as Euclidean or Hamilton distance). KNN algorithm can also be used for clustering problems too which is an unsupervised learning algorithm. For the most part all the cases are recognized by a dominant part vote of its nearest neighbors, with the case being assigned to the class generally regular among its K nearest neighbors are estimated by the separation work. On the off chance that K = 1, than the case is just assigned to the class of its nearest neighbors.

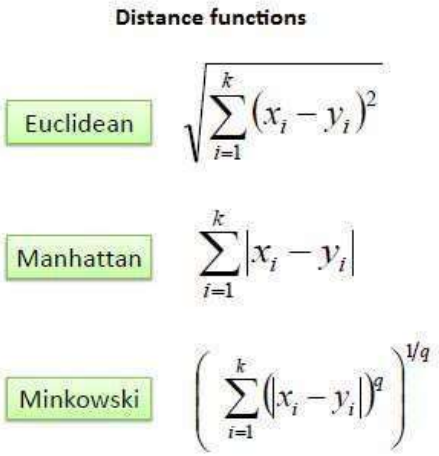


Fig 4.4: Different type of distance function

It should likewise be striking here that each of the three distance measures are legitimate just for the consistent factors. For the event of all out factors the Hamming distance ought to be utilized. It likewise gives the issue of normalization of the numerical factors lying somewhere in the range of 0 and 1 when the numerical and unmitigated qualities are blended in the dataset".

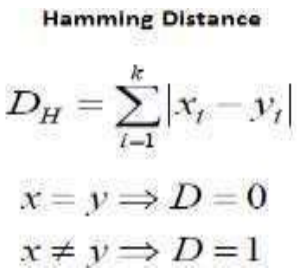


Fig 4.5: Formula of Hamming Distance

- **Support Vector Machine**:-The motivation behind the support vector machine algorithm is finding in a N-dimensional space a hyperplane, here 'n' is the number of attributes that plainly distinguishes the data points. There exist many combinations of hyperplanes that could be chosen in order to separate the different classes of data points. Our initiative of finding a maximum margin plane for example the most elevated distance between any two data points of both the classes. Increasing the margin to maximum distance gives us some reinforcement so that data in near future can be classified with more level of confidence.

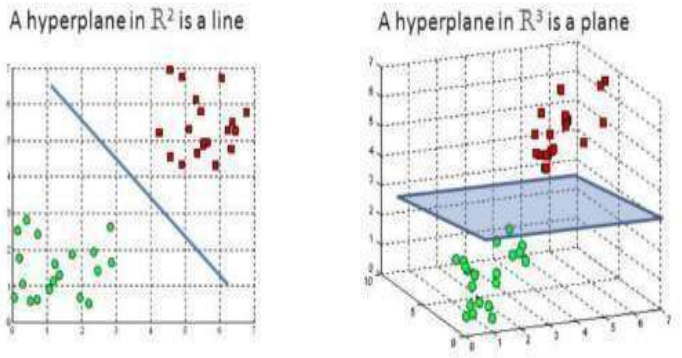


Fig 4.6: Hyperplanes in 2D and 3D Hyperspace

Hyperplanes are used as decision boundaries which help in classifying the data points. Data points which falls on the opposite of the hyperplane is considered of different classes. Therefore the number of features represents the dimension of Hyperplanes. When two input features are considered, that results the hyperplane in just a line. When three input features are taken, then the hyperplane acts as a 2D plane. Situations where number of features is more than 3, it becomes difficult to predict. Support vectors are that data point which are nearer to the hyperplane and makes its impact on hyperplane's position and orientation. We make full use of the margins of a classifiers by implementing these support vectors. The position of the hyperplane changes when we erase the consecutive support vectors. These points help in the implementation of our SVM.

4.3 Dataset Description: -

The data used for training the models is KDDCup99 Data Set collected from Kaggle repositories. The following dataset seems to contain good quality of data with less noisy and disrupted frequency of variations. KDDCup99 data contains 42 data attributes shown in (Table 4.1) that are generally responsible for detection the type of Intrusion and the total number of training tuple provided in the dataset are **125973.** The dataset is identify four types of attacks that are DoS, Probe, R2L and U2R.

Definition of four types of attacks:-

- **Denial of Service Attack (DOS):** A Denial-of-Service (DOS) is a type of attack in which the attacker attempt to prevent the legal users to access their services. This is a type of attack that means to shut down a network or machine that makes the users unable to access their particular files or accounts etc. DOS attack is accomplished by sending the information that implies a crash or by flooding the target with the traffic. In both cases, DOS attack stops the authorized users of the service or resources. There are some general methods of DOS attacks that are flooding services or crashing services. Flood attacks happen when the system receives a huge amount of traffic to buffer for the server, as a result, the system goes slows down and eventually goes stop. The popular flood attacks are Buffer Overflow attacks, ICMP flood, and SYN flood. Problems that can happen by flood attacks are Ineffective services, Interruption of network traffic, connection interface.

- **Probing Attack:** Probing in a type of attack in which the machine or a device that is connected to the network, is scanned by the hacker, to determine the vulnerabilities or any kind of weakness that present in the device so it can be exploite later that can compromise the system. This kind of technique is commonly used in data mining techniques like Saint, Port Sweep, MSCAN, NMAP, etc. According to network security, a probe is a kind of attack by which the attacker can get access to the system files with the help of a known or weak point in the system of the computer.

- **User to Root Attack (U2R):** It is a type of attack, in which an attacker who has an account in a computer system is an expert in misusing or using his or her privileges by creating weakness in computer mechanisms, by creating a bug in the operating system or software's that are installed on the system.

- **Remote to Local Attack (R2L):** Remote to Local (R2L) is a type of attack in which an intruder gains access, either as a user or root of the system, from the remote system through a network. From the majority R2L attacks, the attacker shatters into the computer system via the Internet.

Table 4.1: The table contains name of 42 attributes present in the dataset.

Attribute	1.”Duratin" 2.”protcol_type" 3."service" 4."flag" 5."src_byte" 6."dst_byte" 7."land" 8."wrong_fragment" 9."urgenty" 10."hoter" 11."num_failed_logins" 12. "lgged_in" 13. "num_compromised", 14."root_shell” 15. "su_attempted" 16."num_root" 17."num_file_creations" 18."num_shells" 19."num_access_files" 20."num_outbound_cmds", 21. "is_hst_login" 22."is_guest_login" 23."count" 24."srv_count" 25."serror_rate" 26."srvi_serror_rate" 27."rerror_rate" 28."srv_rerror_rate" 29."same_srv_rate" 30."diff_srv_rate" 31."srv_diff_hst_rate" 32."dst_hst_count" 33. "dst_hst_srv_count", 34."dst_hst_same_srv_rate" 35."dst_hst_diff_srv_rate" 36."dst_hst_same_src_port_rate" 37."dst_hst_srv_diff_hst_rate" 38."dst_hst_serror_rate", 39. "dst_hst_srv_serror_rate" 40. "dst_hst_rerror_rate" 41."dst_hst_srv_rerror_rate" 42."label"

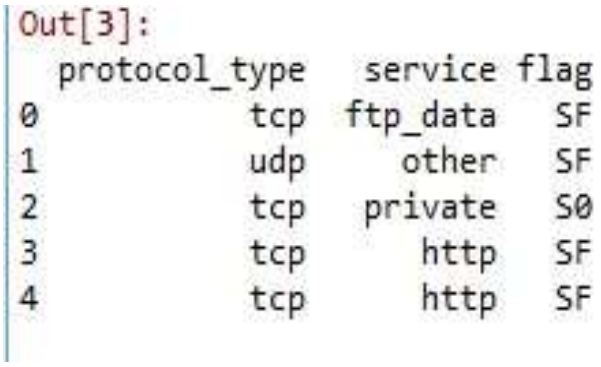
0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0
0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0
0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	remote_job	S0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0

Fig 4.7: Snapshot of Training Dataset.

2	tcp	ftp_data	SF	12983	0	0	0	0	0	0	0	0	0	0	0
0	icmp	eco_i	SF	20	0	0	0	0	0	0	0	0	0	0	0
1	tcp	telnet	RSTO	0	15	0	0	0	0	0	0	0	0	0	0
0	tcp	http	SF	267	14515	0	0	0	0	0	1	0	0	0	0
0	tcp	smtp	SF	1022	387	0	0	0	0	0	1	0	0	0	0
0	tcp	telnet	SF	129	174	0	0	0	0	1	0	0	0	0	0
0	tcp	http	SF	327	467	0	0	0	0	0	1	0	0	0	0
0	tcp	ftp	SF	26	157	0	0	0	0	1	0	0	0	0	0
0	tcp	telnet	SF	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	smtp	SF	616	330	0	0	0	0	0	1	0	0	0	0
0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0
0	tcp	telnet	S0	0	0	0	0	0	0	0	0	0	0	0	0
37	tcp	telnet	SF	773	364200	0	0	0	0	0	1	0	0	0	0

Fig 4.8: Snapshot of Testing Dataset.

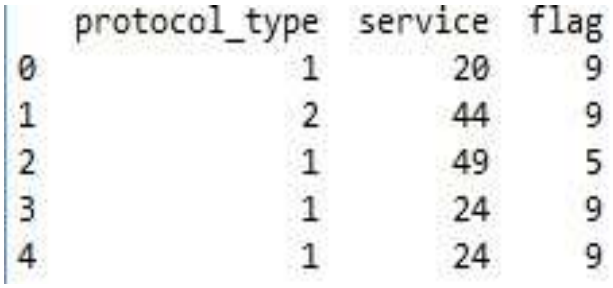
In the above Fig 4.8,Fig 4.9 we can see that in the dataset snapshot there are three attributes that are categorical so the categorical data cannot used for evaluation so we have to convert the attribute in this project using one-hot encoding method is used for convert the attributes. We can see in the below Fig3, Fig4 having snapshots of after and before the dataset using one-hot encoder method.



Out[3]:

	protocol_type	service	flag
0	tcp	ftp_data	SF
1	udp	other	SF
2	tcp	private	S0
3	tcp	http	SF
4	tcp	http	SF

Fig 4.9: Three attributes having categorical data.



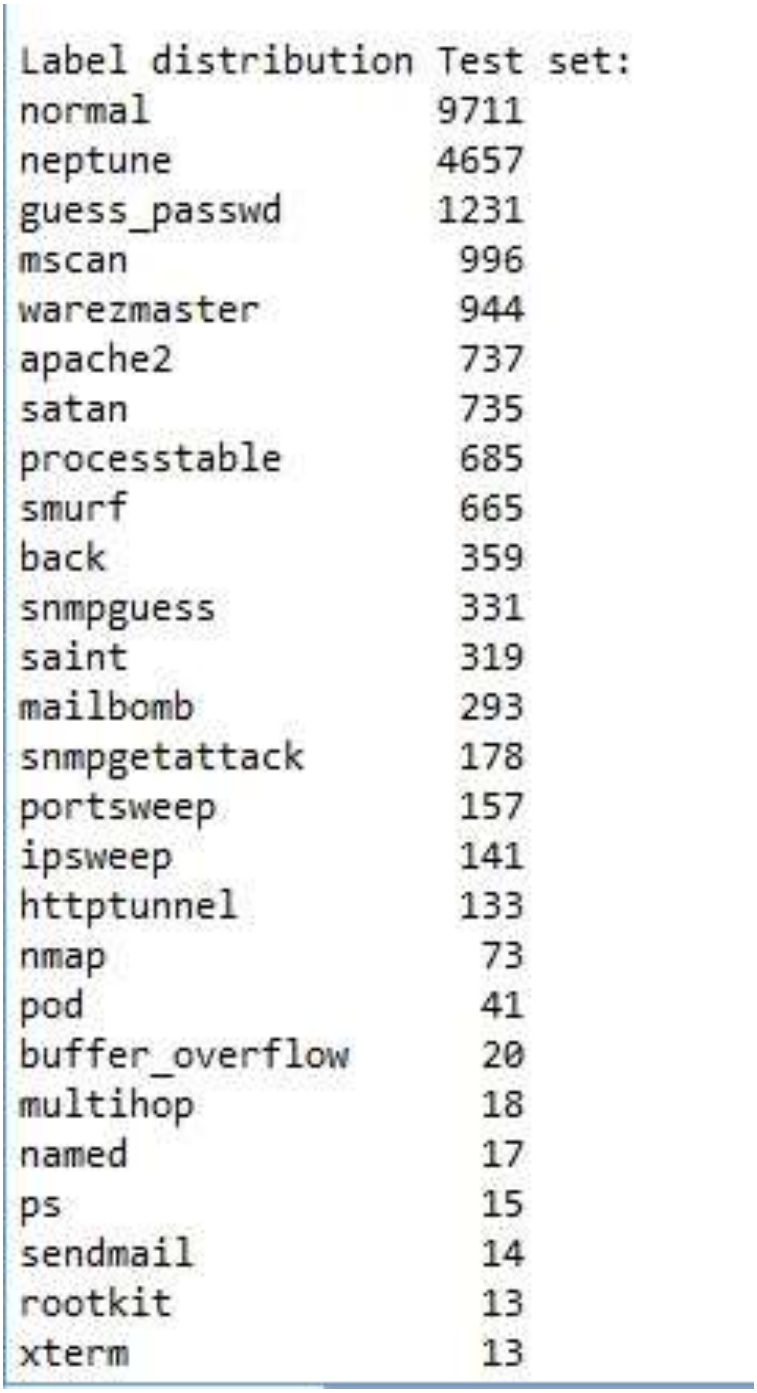
	protocol_type	service	flag
0	1	20	9
1	2	44	9
2	1	49	5
3	1	24	9
4	1	24	9

Fig 4.10: Converted three categorical data.

The Dataset having label attribute that we can see in the table1, the label attribute present in 42 number. The label attribute is having different types of attacks we can see in the below Fig that are divided into four categories of attacks. We use this label attribute as class for classification. But the problem is it having different types of attacks so we cannot use it for class so convert the label into 5 sub categories for evaluate that is 0=Normal, 1=DoS, 2=Probe, 3=U2R and 4=R2L the conversion is shown in the below table.

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

Fig 4.11: Number of rows in different kind of attack present in the training dataset.



```
Label distribution Test set:
normal             9711
neptune            4657
guess_passwd       1231
mscan               996
warezmaster         944
apache2             737
satan               735
processtable        685
smurf               665
back                359
snmpguess           331
saint               319
mailbomb            293
snmpgetattack       178
portsweep           157
ipsweep             141
httptunnel          133
nmap                 73
pod                  41
buffer_overflow      20
multihop             18
named                17
ps                   15
sendmail             14
rootkit              13
xterm                13
```

Fig 4.12: Number of rows in different kind of attack present in the testing dataset.

Table 4.2: Split the class label into 5-sub categories.

Attacks Type	**Class Name**	
normal	Normal	0
nptune	DOS	1
bak		
Lnd		
Pod		
Smurf		
Teardrop		
Maibomb		
apahe2		
Prcesstable		
Udpstorm		
Wom		
ipsweep	Probe	2
nmap		
saint		
satan		
portsweep		
mscan		
ftp_write	R2L	3
httptunnel		
xsnoop		
xlock'		
snmpguess		
snmpgetattack		
named		
sendmail		
warezmaster		
warezclien		
multihop		
phf		
imap		
spy		
guess_passwd		
loadmodule	U2R	4
xterm		
sqlattack		
buffer_overflow		
rootkit		
ps		
perl		

4.4 CODE

- **Give the name to the dataset and print the dimension of data set using pandas library.**

```
# -*- coding: utf-8 -*-
"""
Created on Tue Jan 21 20:58:22 2020
@author: Abhijeet
"""
import pandas as pds
import numpy as np
import sys
import sklearn
import io
import random
column_name= ["duration","protocal_types","servce","flags","srvc_byte",
    "dst_bytes","lands","wrng_fragmentt","urgnt","hoot","nums_filed_logins",
    "logged_in","num_compromiseds","root_shello","su_attempte","nm_rootin",
    "num_file_creations","num_shells","num_access_fies","num_outbound_cmds",
    "is_host_login","is_guest_logins","cont","srv_count","srror_rate",
    "srv_serrr_rates","rerrorr_rates","srv_rerrosr_rates","same_srv_rate",
    "diff_srvc_rates","srv_diff_hst_rates","dst_hst_counts","dst_host_srv_count",
    "dst_host_same_srv_rates","dst_hst_diff_srvc_rates","dst_host_same_srvc_port_rate",
    "dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_hst_srvc_serror_rate",
    "dst_hst_rerror_rate","dst_hst_srv_rerrr_rate","label"]
df = pds.read_csv("KDDTrain+_2.csv", header=None, names = col_names)
df_test = pds.read_csv("KDDTest+_2.csv", header=None, names = col_name)
print('Dimensions of the Training set:',df.shape)
print('Dimensions of the Test set:',df_test.shape)
df.head(5)
```

- **Feature selection using Recursive feature elimination technique(RFE) using Sklearn library.**

```
from sklean.feature_selection import RFE
from sklean.ensemble import RandomForestsClassifier
clf = RandomForestsClassifier(n_estimators=10,n_jobs=2)
rfe = RFE(estimator=clf, n_features_to_select=13, step=1)
rfe.fit(X_DoS, Y_DoS.astype(int))
X_rfeDoS=rfe.transform(X_DoS)
true=rfe.support_
rfecolindex_DoS=[i for i, x in enumerate(true) if x]
rfecolname_DoS=list(colNames[i] for i in rfecolindex_DoS)
rfe.fit(X_Probe, Y_Probe.astype(int))
X_rfeProbe=rfe.transform(X_Probe)
true=rfe.support_
rfecolindex_Probe=[i for i, x in enumerate(true) if x]
rfecolname_Probe=list(colNames[i] for i in rfecolindex_Probe)
rfe.fit(X_R2L, Y_R2L.astype(int))
X_rfeR2L=rfe.transform(X_R2L)
true=rfe.support_
rfecolindex_R2L=[i for i, x in enumerate(true) if x]
rfecolname_R2L=list(colNames[i] for i in rfecolindex_R2L)
rfe.fit(X_U2R, Y_U2R.astype(int))
X_rfeU2R=rfe.transform(X_U2R)
true=rfe.support_
rfecolindex_U2R=[i for i, x in enumerate(true) if x]
rfecolname_U2R=list(colNames[i] for i in rfecolindex_U2R)
print('Features selected for DoS:',rfecolname_DoS)
print()
print('Features selected for Probe:',rfecolname_Probe)
print()
print('Features selected for R2L:',rfecolname_R2L)
print()
print('Features selected for U2R:',rfecolname_U2R)
```

- **Building model for all feature and using 13-feature using RF, KNN and SVM algorithms.**

```
# all features
clf_DoS=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_Probe=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_R2L=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_U2R=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_DoS.fit(X_DoS, Y_DoS.astype(int))
clf_Probe.fit(X_Probe, Y_Probe.astype(int))
clf_R2L.fit(X_R2L, Y_R2L.astype(int))
clf_U2R.fit(X_U2R, Y_U2R.astype(int))
# selected features
clf_rfeDoS=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_rfeProbe=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_rfeR2L=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_rfeU2R=RandomForestClassifier(n_estimators=10,n_jobs=2)
clf_rfeDoS.fit(X_rfeDoS, Y_DoS.astype(int))
clf_rfeProbe.fit(X_rfeProbe, Y_Probe.astype(int))
clf_rfeR2L.fit(X_rfeR2L, Y_R2L.astype(int))
clf_rfeU2R.fit(X_rfeU2R, Y_U2R.astype(int))
#KNeighbors
from sklearn.neighbors import KNeighborsClassifier
clf_KNN_DoS=KNeighborsClassifier()
clf_KNN_Probe=KNeighborsClassifier()
clf_KNN_R2L=KNeighborsClassifier()
clf_KNN_U2R=KNeighborsClassifier()
clf_KNN_DoS.fit(X_DoS, Y_DoS.astype(int))
clf_KNN_Probe.fit(X_Probe, Y_Probe.astype(int))
clf_KNN_R2L.fit(X_R2L, Y_R2L.astype(int))
clf_KNN_U2R.fit(X_U2R, Y_U2R.astype(int))
# selected features
from sklearn.neighbors import KNeighborsClassifier
clf_rfeDoS=KNeighborsClassifier()
```

```
clf_rfeProbe=KNeighborsClassifier()
clf_rfeR2L=KNeighborsClassifier()
clf_rfeU2R=KNeighborsClassifier()
clf_rfeDoS.fit(X_rfeDoS, Y_DoS.astype(int))
clf_rfeProbe.fit(X_rfeProbe, Y_Probe.astype(int))
clf_rfeR2L.fit(X_rfeR2L, Y_R2L.astype(int))
clf_rfeU2R.fit(X_rfeU2R, Y_U2R.astype(int))

#SVM
from sklean.svm import SVC
clfx_SVM_DoS=SVC (kernel='linear', C=1.0, random_state=0)
clfx_SVM_Probe=SVC (kernel='linear', C=1.0, random_state=0)
clfx_SVM_R2L=SVC (kernel='linear', C=1.0, random_state=0)
clfx_SVM_U2R=SVC (kernel='linear', C=1.0, random_state=0)
clfx_SVM_DoS.fit ( X_DoS, Y_DoS.astype(int))
clfx_SVM_Probe.fi t(X_Probe, Y_Probe.astype(int))
clfx_SVM_R2L.fit(X_R2L, Y_R2L.astype(int))
clfx_SVM_U2R.fit(X_U2R, Y_U2R.astype(int))
# selected features
from sklearn.svm import SVC
clf_rfeDoS=SVC(kernel='linear', C=1.0, random_state=0)
clf_rfeProbe=SVC(kernel='linear', C=1.0, random_state=0)
clf_rfeR2L=SVC(kernel='linear', C=1.0, random_state=0)
clf_rfeU2R=SVC(kernel='linear', C=1.0, random_state=0)
clf_rfeDoS.fit(X_rfeDoS, Y_DoS.astype(int))
clf_rfeProbe.fit(X_rfeProbe, Y_Probe.astype(int))
clf_rfeR2L.fit(X_rfeR2L, Y_R2L.astype(int))
clf_rfeU2R.fit(X_rfeU2R, Y_U2R.astype(int))
```

5. SCHEDULE, TASK AND MILESTONES

The work of the project was schedule in parts that are finding the effective data set from the repository then preprocess the dataset finding the best feature reduction method with best machine learning algorithm.

Task of the project is convert the categorical data into numeric data there are some missing data so replace the missing data with mean of the row and we working with large dataset so the finding of effective attribute is the tough task for this project.

5.1 Milestones

5.1.1 Types of Intrusion detection System.

There are many types of intrusion detection system is present that are anomaly based, signature based, host based, network based and many more every intrusion detection system having some advantages and disadvantages as well. There are many technology used to developed an IDS system that is pattern matching, datamining, machine learning, and many more using with this many feature reduction technology is also used to enhance the performance of the system.

5.1.2Importance of Intrusion Detection.

In today's life use of network is very common for every one using this we all books tickets, doing shopping and also we payment using this if the intrusion detection system is not attached in the administration system the network or the ticket provider can't be avail to find is the intruder and threads. Many times intruder sends a huge amount of request to the system that they want to crash so if the system can't find it then the system is crashed IDS system find it and reject the request or send message to the administrator.

5.1.3Recent trends on Intrusion Detection.

There are many hardware and Software Company developed intrusion detection system and they adopt latest technology the company are mainly dell, cisco, HP, IBM and many more. Recently research using deep learning to develop more secure and accurate IDS.

6. PROJECT DEMOSTRATION

Project start with learning the dataset that was used to evaluate. The dataset is having two parts that are training and testing dataset. The training dataset having 125973 dataset and the testing data have 22544 dataset. The dataset having 42 columns. The dataset having three categorical attributes that is protocol type, service, and flags. All the categorical data is converted to numerical data using one-hot encoder method using dummy variable they store the three attribute data and convert that to numerical data then replace that attributes with new numerical data attribute. The dataset having a class level that contains 40 different types of attacks. The attacks are divided into 4 sub categories that is Dos, Probe, U2R, and R2L. The next part of the project is feature scaling in this part there are many attribute that having high magnitude if we used to evaluated the model using this high magnitude data then the output and the accuracy of model come very less from what we expected so, we convert that high magnitude data into a specific range. After the feature scaling part is over we go for feature selection in this part we select the most relevant features from the total no of features. The dataset having 42 features from that we select 13 most effective features using recursive features elimination technique. The recursive feature elimination is one of the most effective methods for feature elimination. It is a machine learning features that is using with any machine learning algorithms. The final step is building model in this project we build three model using machine learning classification algorithms that is Random forest, KNN, and SVM. After building the model, we have to test the model in this project, i used 4 methods for testing the model that is precision, recall, F-measure, and accuracy. All the experiment was done by python anaconda using Spyder data analyzing tool that is an open-source tool.

- **Loading the dataset and giving name to every attribute in the dataset , print the dimension of testing and training data set, and print the first five rows of dataset you can see in the below fig 5.1.**

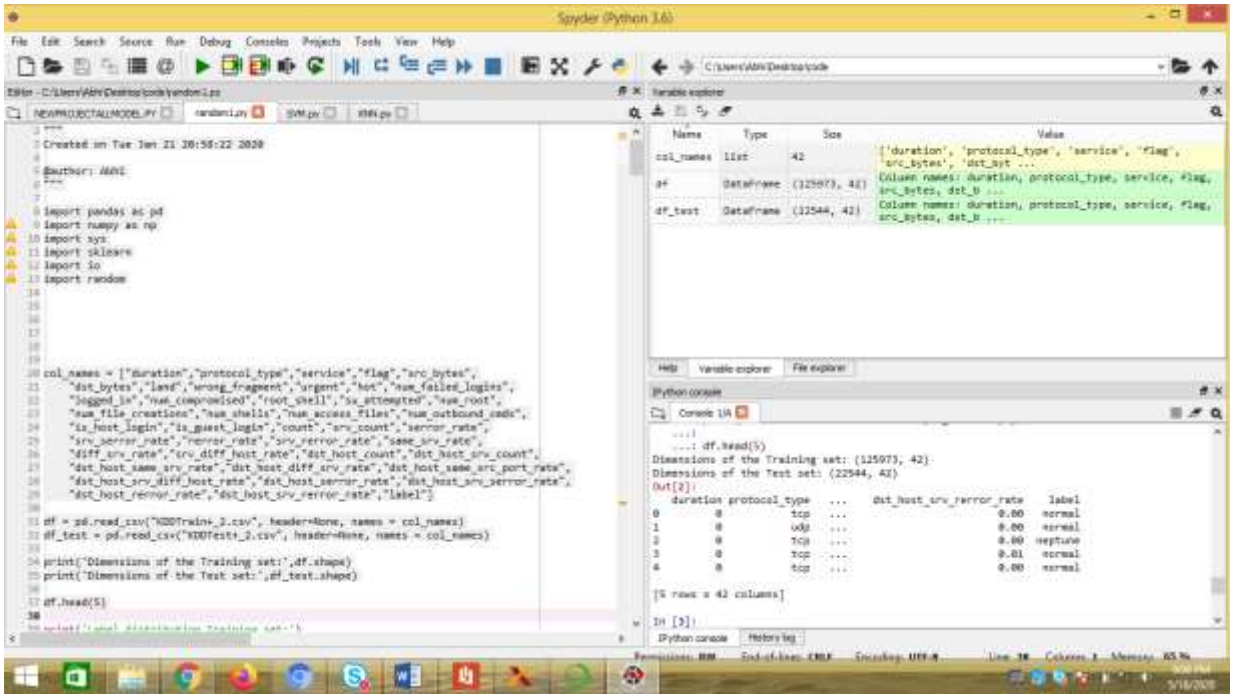


Fig 6.1: Loading and printing dataset.

In the below fig5.2 shown the selecting feature using RFE and printing the dimension of every types of attacks.

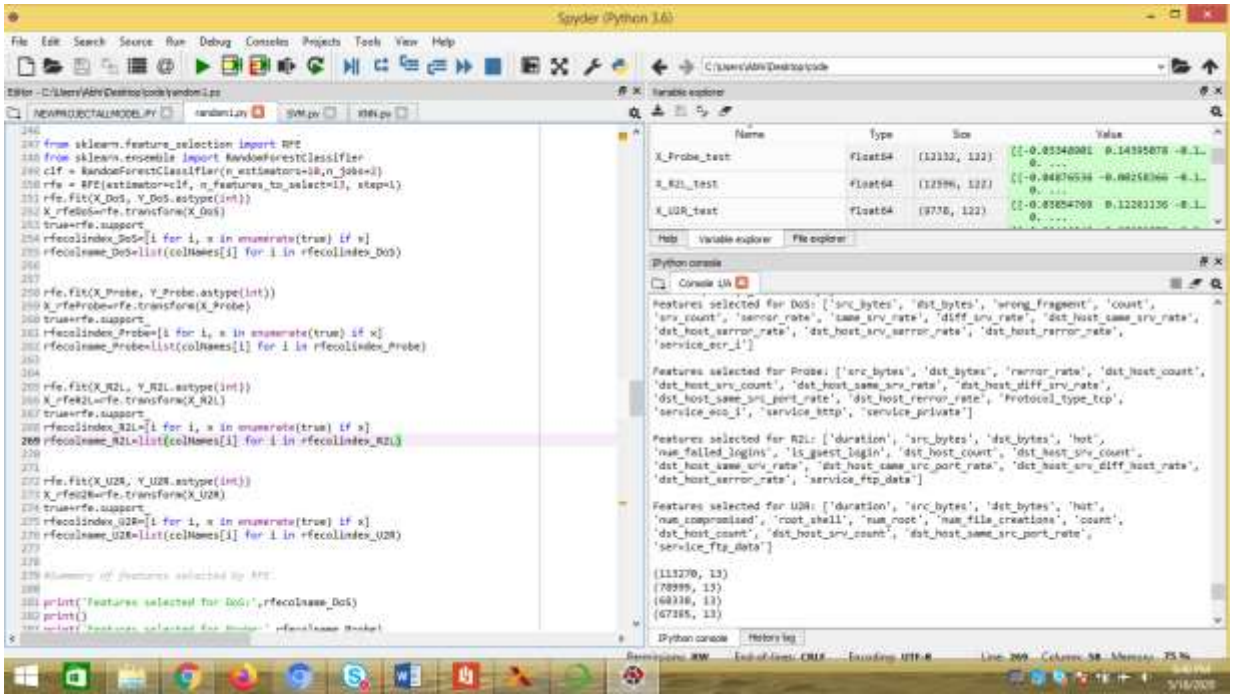


Fig 6.2: Selecting the feature and print the dimension

- **Testing the dataset using confusion matrix, accuracy, precision, recall, and F-measure.**

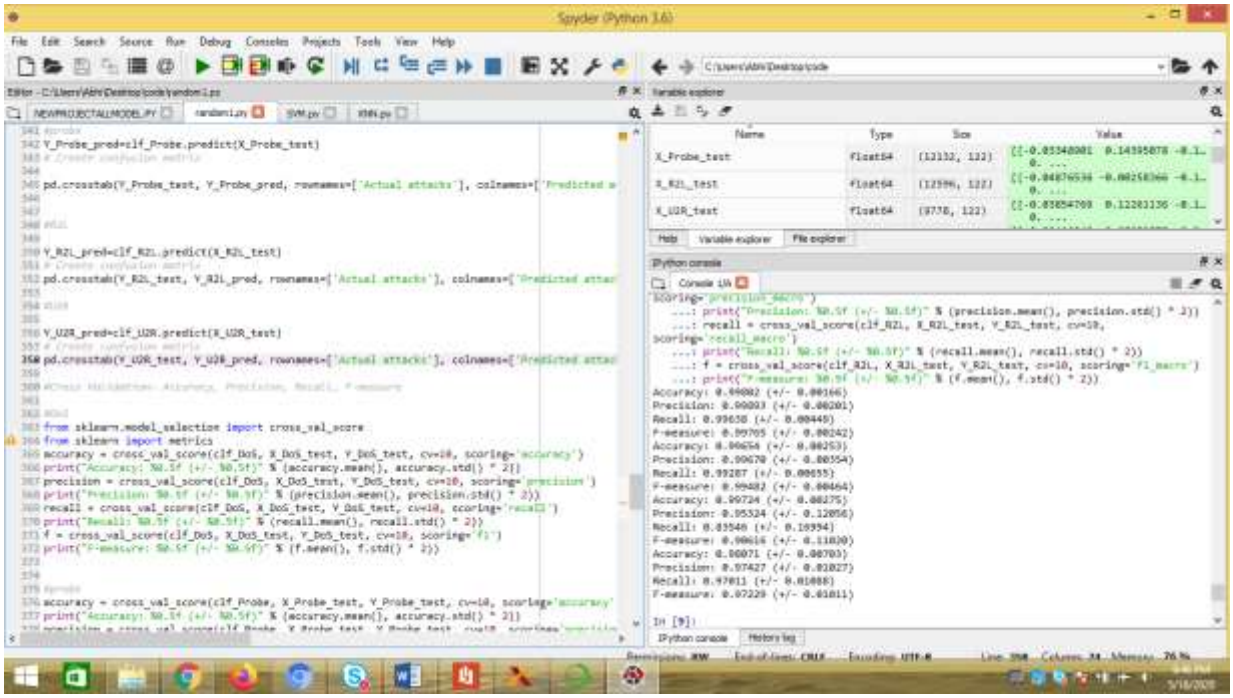


Fig 6.3: finding accuracy, precision, recall and F-measure

7. RESULT

All experiments are done in Python IDE called Anaconda using the Spyder that is included in Anaconda. Python 3.6 is used in this project. In this Project KDDCUP99 dataset is used for analysis which consist of 42 attributes where the last attribute having class label, using this dataset we are build 2 models for every classifier in this project three classifier is used that is RF, SVM and KNN. Two model have using different features one model used the total no of features and one is for only selected features for selecting feature RFE algorithm is used this is also known as recursive feature elimination using this method select only 13 features that are shown in the below table 6.1.To evaluate the working of each classifier and each model we construct confusion matrix to find accuracy, precision, recall and f- measure.

Table 6.1: Feature selected by RFE.

Feature Selected using Recursive Feature Elimination Technique		
No.	**Types**	**Selected Features**
1.	DOS	'src_byte', 'dst_byte', 'wrongs_fragment', 'num_compromised', 'count', 'srv_count', 'serror_rate', 'same_srvc_rate', 'diff_srvc_rate', 'dst_hst_serror_rate', 'Protocol_types_cmp', 'services_ecr_i', 'service_private'
2.	Probe	'src_byte', 'dst_byte', 'count', 'dst_host_count', 'dst_host_srv_count', 'dst_hst_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_hst_same_src_port_rate', 'dst_host_srv_diff_hst_rate', 'dst_hst_rerror_rate', 'Protocol_types_tcp', 'servcice_eco_i', 'service_private'
3.	R2L	'duration', 'src_byte', 'dst_byte', 'hot', 'num_faled_logins', 'is_guest_login', 'count', 'dst_hst_count', 'dst_host_srv_count', 'dst_hst_same_srv_rate', 'dst_host_same_src_port_rate', 'dst_hst_srvc_diff_host_rates', 'service_ftp_data'
4.	U2R	duration', 'src_byte', 'dst_byte', 'hot', 'num_compromised', 'root_shell', 'num_file_creations', 'count', 'dst_host_count', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_hst_same_srcc_port_rates', 'service_ftpp_data'

Confusion matrix

A CM can be defined as a precis of the effects primarily based at the predictions on the category trouble. The quantity of correct as well as wrong predictions is summarized with the matter values and each elegance is damaged down which is likewise the for the confusion matrix. It shows when the prediction is made, how the class version is pressured. It gives the errors in addition to sorts of errors which are being made by means of a classifier.

- True Positive (TP): Observational is positives, and the predicted value to be positive.
- False Negative (FN): Observational is positives, but the predicted value to be negative.
- True Negative (TN): Observational is negatives, and the predicted value to be negative.
- False Positive (FP): Observational is negatives, but the predicted value is positive.

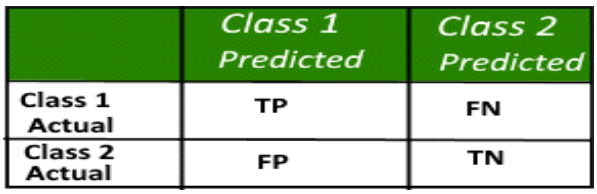


Fig 7.1: Confusion-matrix table.

Experimental results after the analysis of KDDCUP99 dataset in Python IDE(Anaconda).

Confusion Matrix using all features for each category using the classifier **Random Forest (RF):-**

Table 7.2: Confusion-Matrix of DOS Using RF.

Predicted / Actual	Normal	Attack
Normal	9551	160
Attack	1183	1238

Table 7.3: Confusion -matrix of Probe using RF.

Predicted / Actual	Normal	Attack
Normal	9711	0
Attack	7452	8

Table 7.4: Confusion Matrix of R2L using RF.

Predicted / Actual	Normal	attack
Normal	9711	0
Attack	2885	0

Table 7.5: Confusion Matrix of R2L using RF.

Predicted / actual	Normal	Attack
Normal	9711	0
Attack	67	0

Confusion Matrix using all features for each category using the classifier **K-Nearest Neighbors (KNN) :-**

Table 7.6: Confusion Matrix of DOS using KNN.

Predicted / Actual	Normal	Attack
Normal	9692	19
Attack	6879	581

Table 7.7: Confusion Matrix of Probe using KNN.

Predicted / Actual	Normal	Attack
Normal	9367	344
Attack	1232	1189

Table 7.8: Confusion Matrix of R2L using KNN.

Predicted / Actual	Normal	Attack
Normal	9711	0
Attack	2885	0

Table 7.9: Confusion Matrix of U2R using KNN.

Actual \ Predicted	Normal	Attack
Normal	9711	0
Attack	62	5

Confusion Matrix using all features for each category using the classifier **Support Vector Machine (SVM):-**

Table 7.10: Confusion Matrix of DOS using SVM.

Actual \ Predicted	Normal	Attack
Normal	9422	289
Attack	1573	5887

Table 7.11: Confusion Matrix of Probe using SVM.

Actual \ Predicted	Normal	Attack
Normal	9437	274
Attack	1272	1149

Table 7.12: Confusion Matrix of R2L using SVM.

Predicted / Actual	Normal	Attack
Normal	9711	0
Attack	65	2

Table 7.13: Confusion Matrix of U2R using SVM.

Predicted / Actual	Normal	Attack
Normal	9706	5
Attack	2883	2

Confusion Matrix using 13- features Selected by RFE for each category using the classifier **Random Forest (RF):-**

Table 7.14: 13-Features Confusion Matrix of DOS Using RF.

Predicted / Actual	Normal	Attack
Normal	9622	89
Attack	2330	5130

Table 6.15: 13-Features Confusion Matrix of Probe Using RF.

Predicted / Actual	Normal	Attack
Normal	9360	351
Attack	1292	1129

Table 7.16: 13-Features Confusion Matrix of R2L Using RF.

Predicted / Actual	Normal	Attack
Normal	9711	0
Attack	2884	1

Table 7.17: 13-Features Confusion Matrix of U2R Using RF.

Predicted / Actual	Normal	Attack
Normal	9711	0
Attack	60	7

Confusion Matrix using 13-features Selected by RFE for each category using the classifier **K-Nearest Neighbors (KNN) :-**

Table 7.18: 13-Features Confusion Matrix of DOS Using KNN.

Predicted / Actual	Normal	Attack
Normal	9632	79
Attack	5473	1987

Table 7.19: 13-Features Confusion Matrix of Probe Using KNN.

Predicted / Actual	Normal	Attack
Normal	9484	227
Attack	1386	1035

Table 7.20: 13-Features Confusion Matrix of R2L Using KNN.

Predicted / Actual	Normal	Attack
Normal	9707	4
Attack	2872	13

Table 7.21: 13-Features Confusion Matrix of U2R Using KNN.

Predicted / Actual	Normal	Attack
Normal	9708	3
Attack	51	16

Confusion Matrix using 13-features Selected by RFE for each category using the classifier **SVM (Support Vector Machine):-**

Table 7.22: 13-Features Confusion Matrix of DOS Using SVM.

Predicted / Actual	Normal	Attack
Normal	8821	890
Attack	1669	5791

Table 7.23: 13-Features Confusion Matrix of Probe Using SVM.

Predicted / Actual	Normal	Attack
Normal	9653	58
Attack	2203	218

Table 7.24: 13-Features Confusion Matrix of R2L Using SVM.

Predicted / Actual	Normal	Attack
Normal	9684	27
Attack	2768	117

Table 7.25: 13-Features Confusion Matrix of U2R Using SVM.

Predicted / Actual	Normal	Attack
Normal	9710	1
Attack	60	7

In the above tables confusion matrix is shown for every model of four types of attacks for calculation of accuracy, recall, precision and F-measure we need the confusion matrix, in the below tables accuracy, recall, precision is shown for every table.

Accuracy:

Accuracy can be said as the portion of total predictions that are correct. Generally, if a prediction gives an accuracy of 99 percent, that accuracy is said to be excellent or good or mediocre, poor or terrible depends on the situation. Accuracy can be find using the following equation:

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

Recall:

A recall is said to be the ratio of the entire range of efficaciously categorized positive samples to the full quantity of positive samples. If a recall is excessive, then it suggests that class has been effectively diagnosed.

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP+FP}}$$

Precision:

Precision can be calculated by the toatal ratio of correctly classified positive samples divided by the total ratio of predicted positive samples. If the precision is high then the sample which is labeled as positive will be indeed positive. The equation for precision is:

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP+FN}}$$

F-Measure:

To calculation for the F-measure we want recall and precision that helps us to find f-measure we can see in the below equation how the f-measure is calculated using recall and precision. F measure can be calculated by Harmonic Mean instead of Arithmetic Mean.

F measure always resides to the smaller value of the Recall or Precision.

$$\textit{F - measure} = \frac{2*\text{Recall}*\text{Precision}}{\text{Recall} + \text{Precision}}$$

Comparing the three model using Accuracy, Precision, Recall and F- Measure Using all the Features:-

Table 7.26: Accuracy, recall, precision and F-measure is calculated using all features.

Random Forest				
	Accuracy	**Precision**	**Recall**	**F-Measure**
DOS	99.70	99.75	99.62	99.66
Probe	99.32	99.09	98.80	98.96
R2L	97.85	97.17	96.62	96.98
U2R	99.71	91.82	81.27	85.53
KNN				
	Accuracy	**Precision**	**Recall**	**F-Measure**
DOS	99.09	98.82	99.03	98.96
Probe	98.21	96.94	97.53	97.72
R2L	95.07	92.85	93.29	93.05
U2R	99.54	8.19	77.42	80.64
SVM				
	Accuracy	**Precision**	**Recall**	**F-Measure**
DOS	95.93	96.89	93.64	95.24
Probe	95.40	92.76	92.90	92.82
R2L	84.99	78.59	80.47	79.44
U2R	99.50	90.46	70.07	76.25

Comparing the three model using Accuracy, Precision, Recall and F- Measure using all 13-Features:-

Table 7.27: Accuracy, recall, precision and F-measure is calculated using 13-features

Random Forest				
	Accuracy	**Precision**	**Recall**	**F-Measure**
DOS	99.80	99.82	99.67	99.75
Probe	99.67	99.62	99.30	99.46
R2L	99.75	96.82	84.37	90.23
U2R	98.11	97.39	96.78	97.16
KNN				
DOS	99.71	99.67	99.66	99.67
Probe	99.07	98.60	98.50	98.55
R2L	96.73	95.31	95.48	95.38
U2R	99.70	93.28	84.83	87.75
SVM				
DOS	99.37	99.10	99.45	99.27
Probe	98.45	96.90	98.36	97.61
R2L	96.79	94.85	96.26	95.52
U2R	99.65	91.98	83.98	85.91

Cross validation score in comparison to number of features Using all the features:-

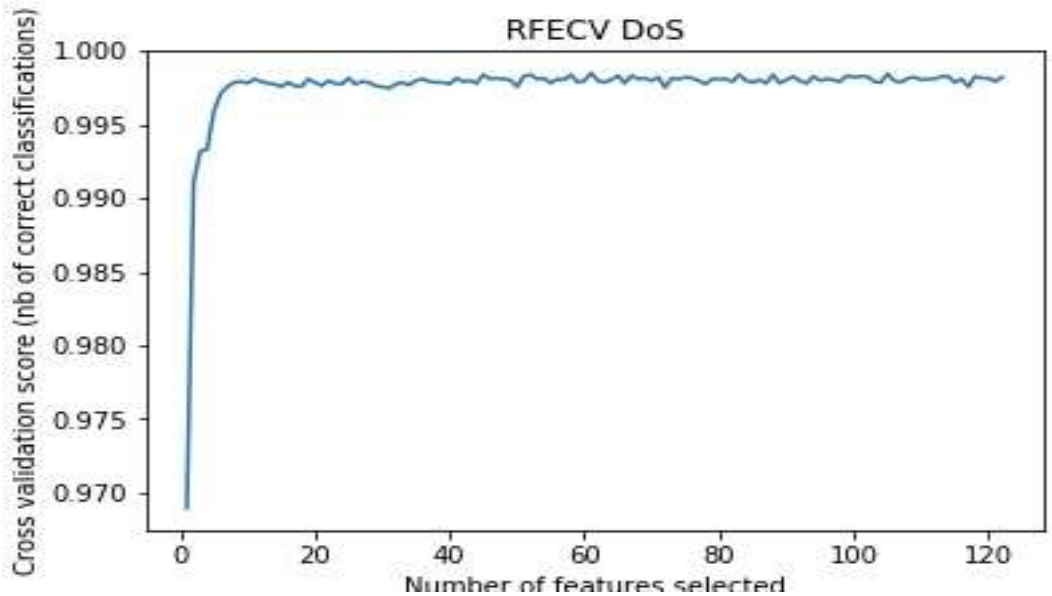


Fig 7.2: Comparison of cross-validation score with number of features using DOS.

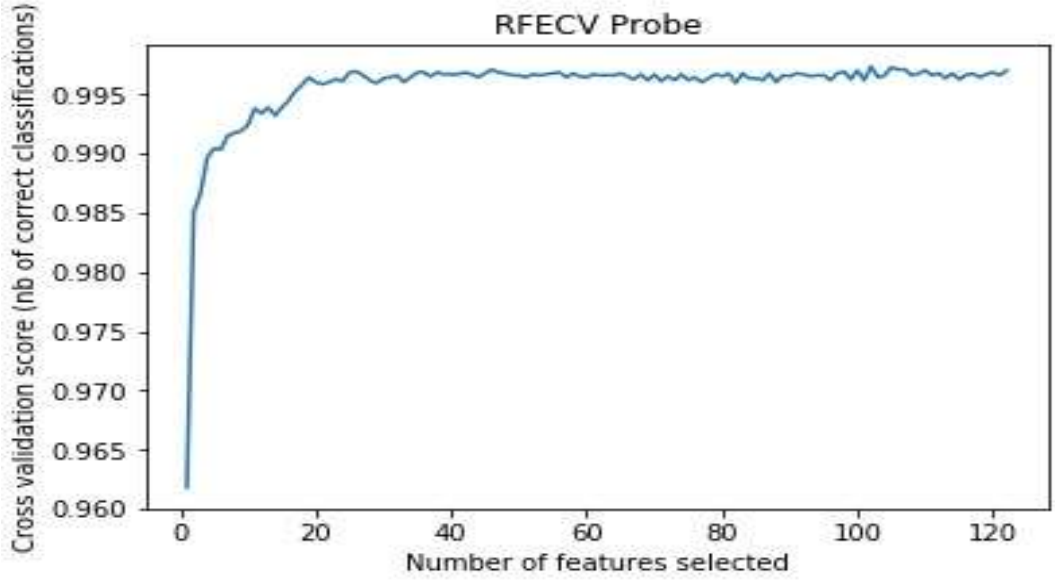


Fig 7.3: Comparison of cross-validation score with number of features using Probe.

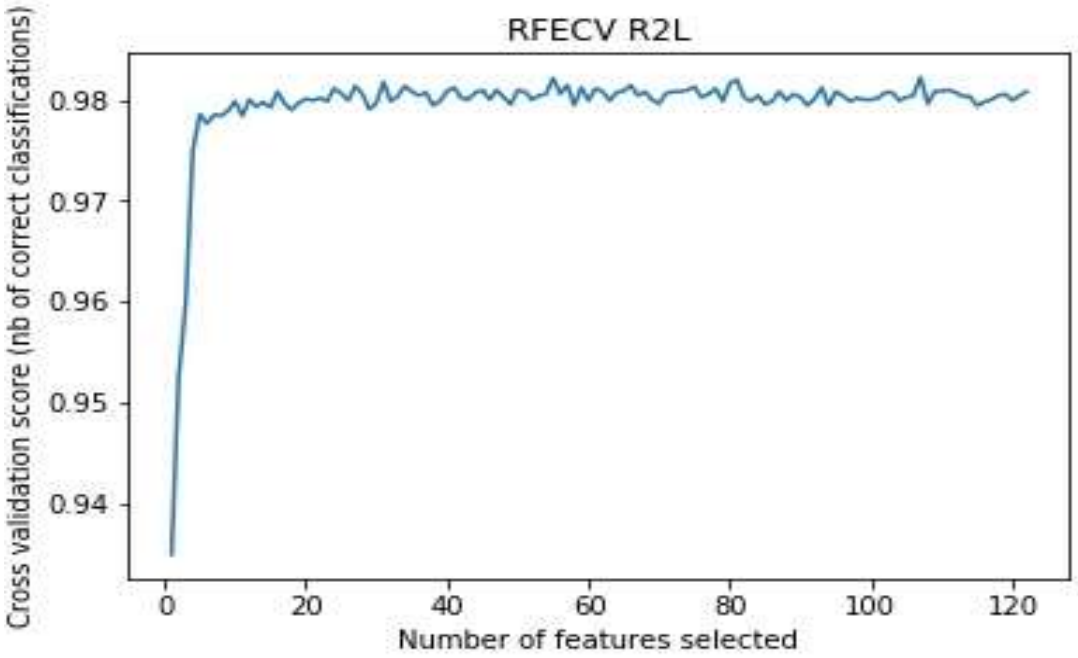


Fig 7.4: Comparison of cross-validation score with number of features using R2L.

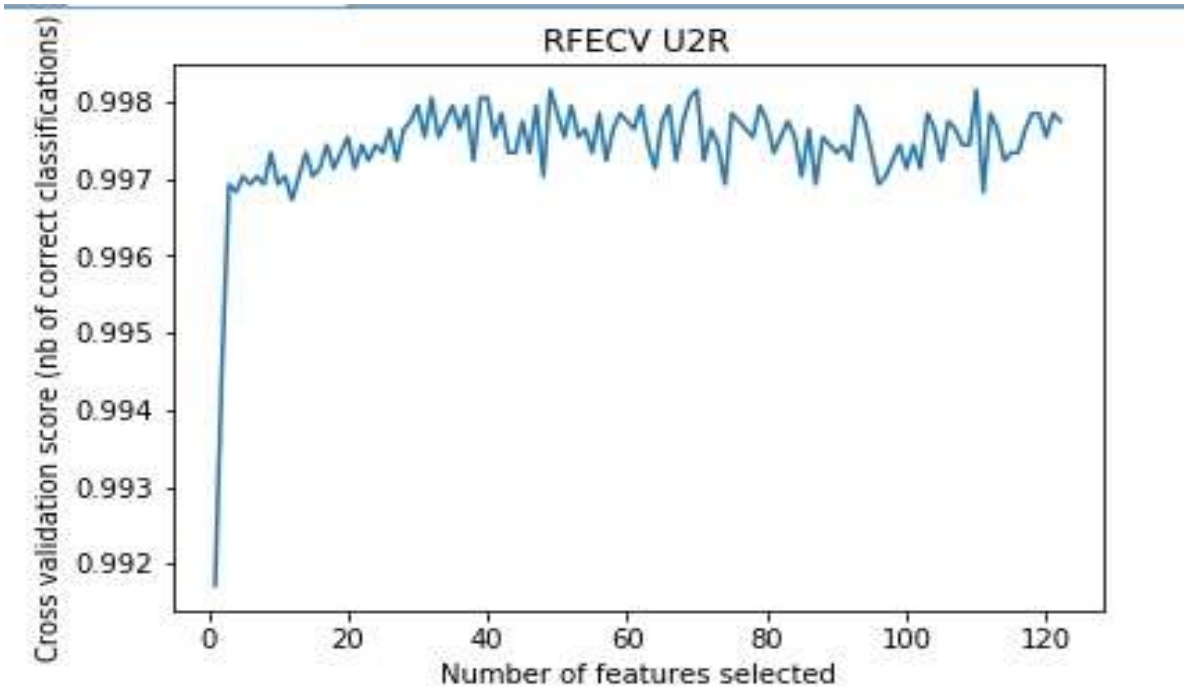


Fig 7.5: Comparison of cross-validation score with number of features using U2R.

Cross validation score in comparison to number of features Using 13-Features:-

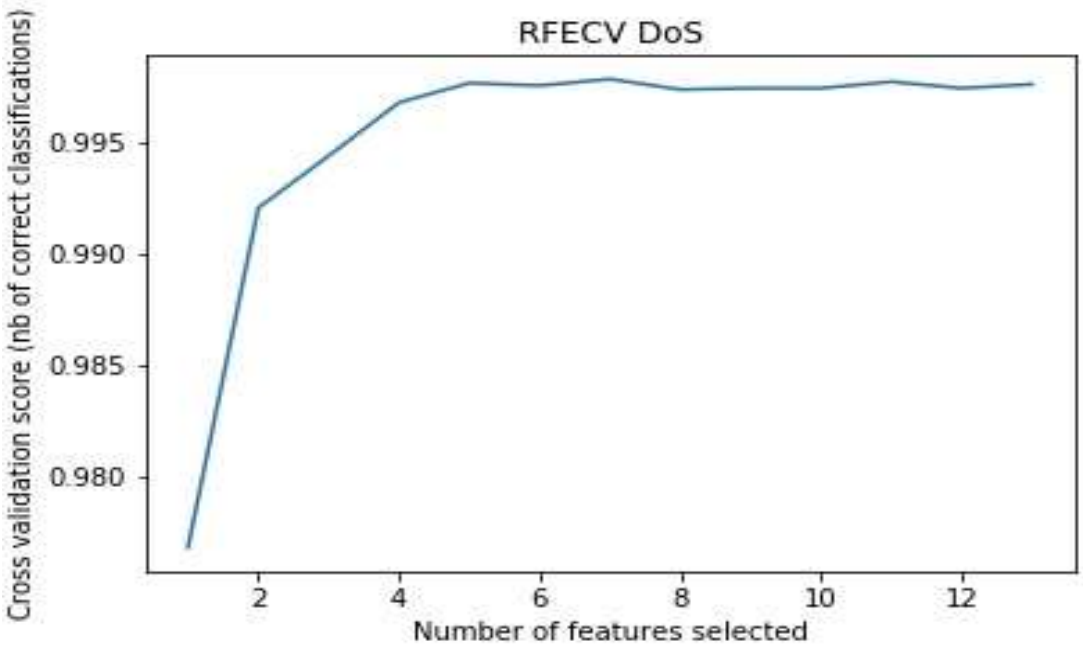


Fig 7.6: Comparison of cross-validation score with number of features using DOS.

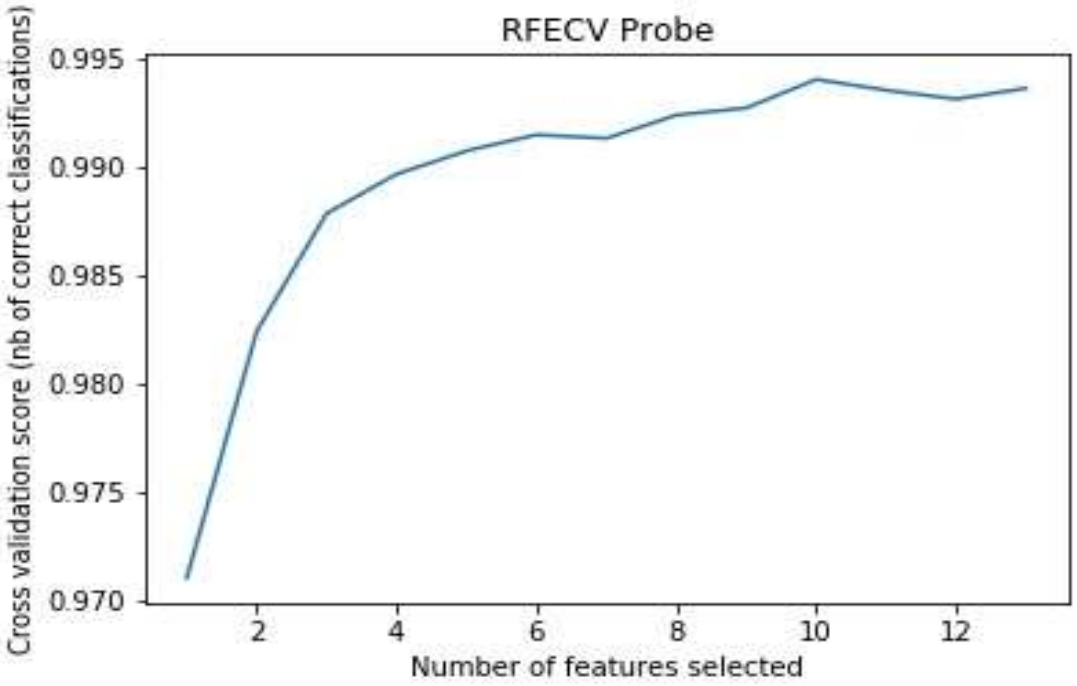


Fig 7.7: Comparison of cross-validation score with number of features using Probe.

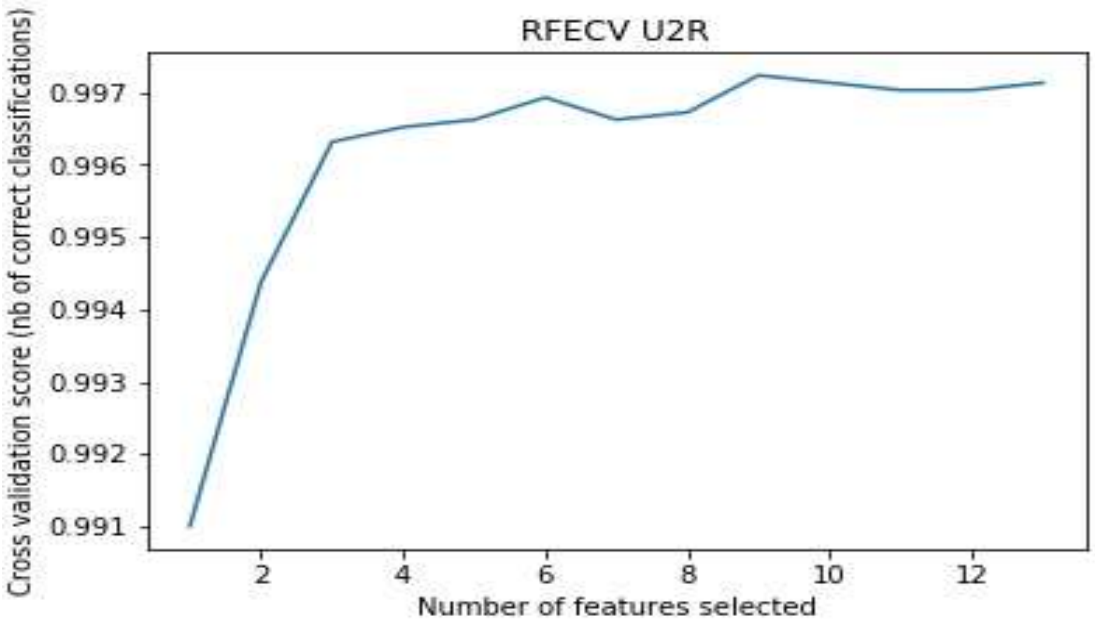


Fig 7.8: Comparison of cross-validation score with number of features using R2L.

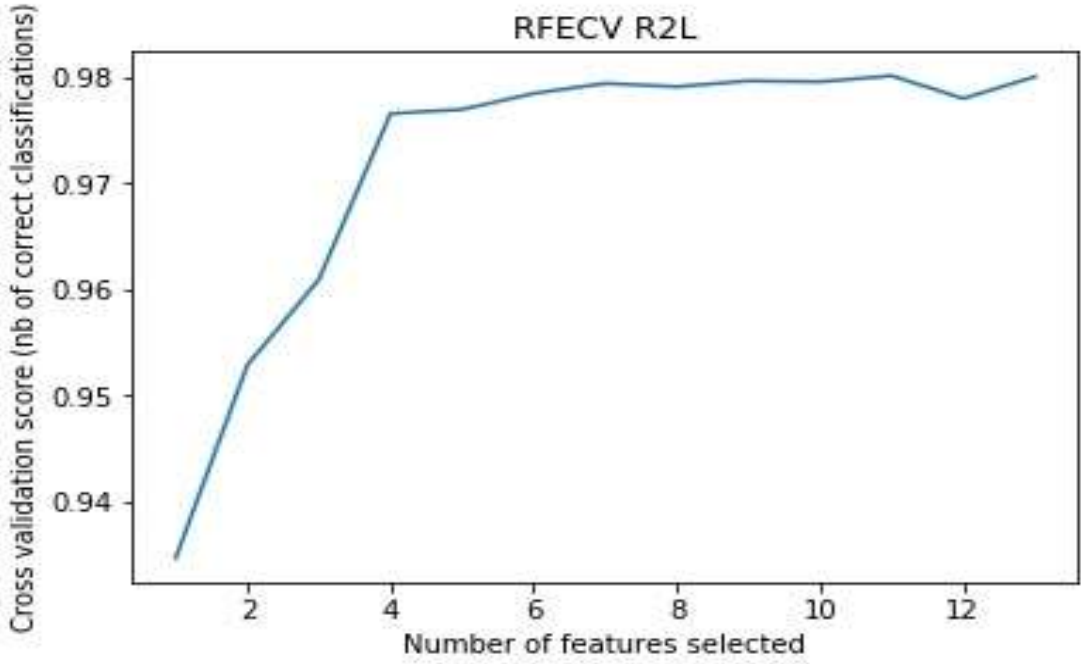


Fig 7.9: Comparison of cross-validation score with number of features using U2R.

8. SUMMARY

The project is dealing with IDS using machine learning algorithms. Three types of machine learning algorithms are used to finding four types of attacks that are DOS, Probe, R2L, and U2R, by using all the features and also by reducing the attribute using RFE. In this project, the KDDCUP99 dataset used to evaluate the model using three classifiers that are RF, KNN, and SVM, using each classifier two models are built one is for all attributes, and one is used for 13-attribute. When compare the accuracy, precision, recall, and f-measure using all the feature with using 13-features, using 13-features give a more accurate result. Comparing the three models with each other Random forest provides more accuracy then the other two models. In the future, if any better classification algorithm is proposed to increase the classifier accuracy, then that modification can be done in this project.

9. REFERENCES

[1] Hyunsung Choi, Mintae Kim, Gyubok Lee, Wooju Kim " Unsupervised learning approach for network intrusion detection system using autoencoders",The Journal of Supercomputng, 6 March 2019, pp-5597-5621.

[2] JiviteshSharma, CharulGir ,Ole-ChrstofferGranmo and MortenGoodwin" Multi-layer intrusion detection system with Extra Trees feature selection ,extreme learning machine ensemble, and softmax aggregation", EURASIP Journal on Informaton Security,2019, pp-1-16.

[3] Azar Abd Salih, Mawan Bahjat Abdulrazaq" Combining Best Features Selection Using Three Classifiers in Intrusion Detection System", 2019 International Conference on Advanced Scence and Engineering (ICOASE),2019, pp.94-99.

[4] Ju-ho woo, Joo-Yeop Song and Young-June choi,"Performance enhancement of deep neural network using feature selection and preprocessing for intrusion detection", International Conference on Artificial Intelligence in Information and Communcation (ICAIIC), pp.415-417,2019.IEEE**.**

[5] Ahmed Ahmim, Leandros Maglaras, Mohamed Amine Ferrag, Makhlouf Derdour, Helge Janicke, "A Novel Hierarchical Intrusion Detection System based on Decision Trees and Rules-based Models", 2019 15th Internatonal Conference on Distributed Computing in Sensor Systems (DCOSS), pp.228-233,2019, DOI 10.1109/DCOSS.2019.00059.

[6] Jiong Zhang, Mohammad Zulkernine, and Anwar Haque, "Random-Forests-Based Network Intrusion Detection Systems", Ieee Transactions On Systems, Man, And Cybernetics, VOL. 38,NO.5,September 2008.

[7] Swati Paliwal, Ravindra Gupta, "Denial-of-Service, Probing &Remote to User (R2L) Attack Detection using Genetic Algorithm", International Journal of Computer Applications, Volume 60– No.19, December 2012.

[8] Nabila Farnaaz, M. A. Jabbar, “Random Forest Modeling for Network Intrusion Detection System”, Twelfth International Multi-Conference on Information Processing (IMCIP),2016.

[9] Phyu Thi Htun, Kyaw Thet Khaing, “Anomaly Intrusion Detection System using Random Forests and k-Nearest Neighbor”, International Journal of P2P Network Trends and Technology (IJPTT), Volume 3 Issue 1 January to February,2013.

[10] Mehed Hasan, Mohammed Nasser, Shamm Ahmad, Khademul Islam Molla, "Feature Selection for Intrusion DetectionUsing Random Forest",Journal of Information Security , pp. 129-140 ,2016.

[11] Kritka Singh, Bharti Nagpal, “Random Forest Algorithm in Intrusion Detection System : A Survey”, Internatonal Journal of Scientific Research in Computer Science, Volume 3, pp. 673-676,2018.

[12] Ripon patgri,Udit Varshney, Tanya Akutota, and Rakesh Kunde," An Investigation on Intrusion Detection System using machine Learning", symposium Series on Computational Intelligence(SSCI), January 2019.

[13] Faezah Hamad Almasoudya, Wathq Laftah Al-Yaseenb, Ali Kadhum Idreesa, "Differential Evolution Wrapper Feature Selection for Intrusion Detection System" International Conference on computational Intelligence and data science (ICCIDS),2019.

[14] Kunala ,Mohit Dua,"Attribute Selection and Ensemble Classifier based Novel Approach to Intrusion Detection System" ,Internatonal Conference On computational intelligence and data-Science (ICCIDS),pp.2191-1=2199,2019.

[15] Amreen Sultana, M.A. Jabbar, " Network Intrusion Detection System using Data Mining Techniques, "International Conference on Applied and Theoretical Computing and communication technology (iCATccT),2016.

[16] Zhang, Hao Dai, Shumin Li, Yongdan Zhang, Wenjun,"Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark" International Performance Computing and communications conference (IPCCC), 2018.

[17] Patil, Gayatri V. Pachghare, K. Vinod. Kshirsagar, Deepak D," Feature Reduction in Flow-Based Intrusion Detection System." International Conference on Recent Trends in Electronics, Information & communication technology (RTEICT), May 2018.

[18] A. A. Aburomman and M. B. IbneReaz, "A novel SVM- kNNPSO ensemble method for intrusion detection system", Applied Soft Computing Journal", Vol. 38, pp. 360–372, 2016..

[19] Q. S. Qassim, A. M. Zin, and M. J. Ab Aziz, "Anomalies classification approach for network—based intrusion detection system", International Journal of Network Security, pp. 1159–1171,2016.

[20] A.R. Jakhale, G.A. Patil, "Anomaly Detection System by Mining Frequent Pattern using Data Mining Algorithm from Network Flow", "International Journal of Engineering Research and Technology, Vol. 3, No.1, January 2014, ISSN. 2278-0181.

[21] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection", IEEE Communications Surveys and Tutorials, Vol. 18, No. 2, pp. 1153–1176,2016.