VIT

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

Intrusion Detection System using Feature Selection and Decision Tree Classification Algorithm

CSE3502 Information Security and Management

WIN 2021-2022
Slot – F1

Team members

Harsh Agrawal 19BCE2360
Harshvardhan Singh Gahlaut 19BCE2372
Aman Yadav 19BCE2605

Submitted to
Professor Selvi M.

Abstract:

One of the most important aspects of security is to monitor the network traffic. The best way to accomplish this is to tap the traffic with an intrusion detection system Nowadays it is very important to maintain a high level security to ensure safe and trusted communication of information between various organizations. But secured data communication over internet and any other network is always under threat of intrusions and misuses. So Intrusion Detection Systems have become a needful component in terms of computer and network security. With the increasing amount of network throughput and security threat, the study of intrusion detection systems (IDSs) has received a lot of attention throughout the computer science field.

So to achieve this, we have 3 different types of detection techniques under IDS namely (i) Signature-based detection, (ii) Anomaly-based detection and (iii) Detection based on stateful protocol analysis. Signature-based detection- Depending on how they present themselves, several security threats can be identified.

Anomaly-based detection- This IDP approach works by looking for irregularities in a network traffic flow. Anomaly detection is performed by comparing a set profile of permissible traffic to the network's actual traffic. Detection based on stateful protocol analysis - The manufacturers of IDP devices define predefined profiles of protocol operation that describe everything that is permitted or not acceptable in the exchange of messages in a protocol. Unlike anomaly-based detection, which creates profiles based on hosts or unique network activity, stateful protocol analysis uses generic profiles established by equipment makers. The technique of comparing predefined operation profiles with the specific data flow of a protocol on the network is known as stateful protocol analysis. Most IDP systems employ many detection methods at the same time, resulting in a more comprehensive and precise detection method.

Keywords – classification; decision tree; features selection; intrusion detection system; NSL-KDD

Introduction:

In 1987 Dorothy E. Denning proposed intrusion detection as is an approach to counter the computer and networking attacksand misuses. Network Intrusion Detection Systems (NIDSs) are essential tools for the network system administrators to detect various security breaches inside an organization’s network. An NIDS monitors and analyzes the network traffic entering into or exiting from the network devices of an organization and raises alarms if an intrusion is observed. Machine Learning is one of the technique used in the IDS to detect attacks. Machine learning is concerned with the design and development of algorithms and methods that allow computer systems to autonomously acquire and integrate knowledge to continuously improve them to finish their tasks efficiently and effectively. In the project we have implemented a machine learning network intrusion detection system which classifies if a network is safe or else tells the type of attack the network is being subjected to from the following four attacks – (i)DOS (ii)Probe (iii)R2L (iv)U2R. A model is created and trained with the NSL-KDD dataset by using decision tree model. We have applied ANOVA F-test for univariate feature selection. With its aid we have tried to find a dependency between an attack and a feature and form a subset of the relevant features that fully represent the problem.

IDSs that are based on network traffic gather network traffic in order to detect intruders. Most of the time, these systems act as packet sniffers, reading incoming information and calculating particular metrics to determine whether a network has been hacked. TCP/IP, NetBEUI, and XNS, among other internet and proprietary protocols that manage messages between external and internal networks, are vulnerable to attack and require extra means to identify malicious events. Intrusion detection systems frequently struggle to cope with encrypted data and traffic from virtual private networks. Over 1Gbps speed is also a bottleneck, despite the fact that current and expensive network-based IDSs can work at this rate.

One of the most crucial components of a distributed intrusion detection architecture is cooperative agents. An agent is a piece of autonomous or semi-autonomous software that operates in the background and performs valuable activities for someone else.

An agent, in comparison to IDSs, is typically a piece of software that detects intrusions locally and communicates assault data to central analysis servers. The cooperative agents can build a network for data transmission and processing among themselves.

Multiple agents deployed across a network provide a more comprehensive picture of the network than a single IDS or centralised IDS.

Literature Survey:

- **Bayesian based intrusion detection system (Hesham Altwaijry, Saeed Algarny) –** In the project, the Bayesian method has been used as an engine to classify the data. This has been done with an objective to increase the accuracy to of R2L attack. The research results show that they could have results for R2L attack with a DR of 85.35% by using the three features: 23, 24 and 31 and a threshold value of 0.6. The CR is 76.69 because they used a low threshold value which reduces the accuracy of detection of normal records (TN) but increases the DR for R2L attack.

- **Modeling intrusion detection system using hybrid intelligent systems (Sandhya Peddabachigari ,Ajith Abraham) –** In this research, first the researchers have used decision tree(DT) and SVM models for intrusion detection and evaluated their performance based on the benchmark KDD Cup 99 Intrusion data. Then they designed a hybrid DT–SVM model and an ensemble approach with DT, SVM and DT–SVM models as base classifiers. Empirical results reveal that DT gives better or equal accuracy for Normal, Probe, U2R and R2L classes. The hybrid DT–SVM approach improves or delivers equal performance for all the classes when compared to a direct SVM approach. The Ensemble approach gave the best performance for Probe and R2L classes. The ensemble approach gave 100% accuracy for Probe class,and this suggests that if proper base classifiers are chosen 100% accuracy might be possible for other classes too.

- **Evaluation of Machine Learning Algorithms for Intrusion Detection System (Mohammad Almseidin, Maen Alzubi) –** In this paper, several experiments were performed and tested to evaluate the efficiency and the performance of the following machine learning classifiers: J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network. All the tests were based on the KDD intrusion detection dataset. The rate of the different type of the attacks in the KDD dataset are approximately 79% of DOS attacks, 19% of normal packets and 2% of other types of attacks (R2l, U2R and PROBE). In the experiments 148753 instances of records have been extracted as training data to build the training models for the selected machine learning classifiers. The experiments have demonstrated that there is no single machine learning algorithm which can handle efficiently all the types of attacks. All of the selected machine learning classifiers, except the MLP, were able to built their training models in an acceptable period of time. Furthermore, to save the availability and the confidentiality of the network resources, the true positive and the average accuracy rates alone are not sufficient to detect the intrusion. False negative and false positive rates are also needed to be taken into consideration.

- **A Deep Learning Approach for Network Intrusion Detection System (Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam) –** The given paper proposes two approaches for the evaluation of NIDS. The first approach, the training data is used for both training and testing either using n-fold cross-validation or splitting the training data into training, cross-validation, and test sets. NIDSs based on this approach achieved very high accuracy and less false-alarm rates. The second approach uses the training and test data separately for the training and testing. Since the training and test data were collected in different environments, the accuracy obtained using the second approach is not as high as in the first approach. Therefore, the result of the second approach has been emphasized for the accurate evaluation of the NIDS.

- **AN IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING GENETIC ALGORITHM (Mohammad Sazzadul Hoque, Md. Abdul Mukit and Md. Abu Naser Bikas) –** In this paper, an intrusion detection system has been successfully implemented using genetic algorithm to efficiently detect different types of network intrusions. The standard KDD99 benchmark dataset has been used to implement and measure the performance of the system and got reasonable detection rate. To measure the fitness of a chromosome the standard deviation equation with distance has been used.

- **Survey on Intrusion Detection System using Machine Learning Techniques (Sharmila Wagh, Vinod K. Pachghare) –** In this paper authors have presented an overview of machine learning technologies which are being utilized for the detection of attacks in IDS and system design of effective IDS. The security of information in computer based systems is a major concern to researchers. The work of IDS and methodologies which has been a major focus of information security related research. Machine learning is a vast and advanced field still relatively immature and definitely not optimized for IDS.

- **Deep Learning Approach For Intelligent Intrusion Detection System (Maneesha M, Savitha V, Jeevika S) –** The proposed architecture converts system call traces to a ngram vector representation model first. The input feature vectors are then reduced in size using a dimensionality reduction method that selects only those n-gram terms whose frequencies are greater than a predefined threshold value. Finally, deep learning is used to process the dimensionality reduced vectors in order to decide if the corresponding system call traces are natural and intrusive. The proposed framework reliably distinguishes the natural and intrusive device processes while minimizing overall computational overheating, according to experimental findings on the benchmark ADFA-LD dataset.

- **Random Forest Modeling for Network Intrusion Detection System (Nabila Farnaaz∗ and M. A. Jabbar) –** This paper deals the Random Forest (RF) algorithm to detect four types of attack like DOS, probe, U2R and R2L. The researcher adopted 10 cross-validation applied for classification. Feature selection is applied on the data set to reduce dimensionality and to remove redundant and irrelevant features. They applied symmetrical uncertainty of attributes which overcomes the problems of information gain. The proposed approach is evaluated using NSL KDD data set. They compared our random forest modeling with j48 classier in terms of accuracy, DR, FAR and MCC. Their experimental result prove that accuracy, DR and MCC for four types of attacks are increased by our proposed method.

- **Importance of Intrusion Detection System (Asmaa Shaker Ashoor) –** The research paper gives us a brief study of the intrusion detection system, its history and the different categories and classifications it has. The research paper is concluded with "An intrusion detection system is a part of the defensive operations that complements the defences such as firewalls, UTM etc. The intrusion detection system basically detects attack signs and then alerts. According to the detection methodology, intrusion detection systems are typically categorized as misuse detection and anomaly detection systems. The deployment perspective, they are be classified in network based or host based IDS. In current intrusion detection systems where information is collected from both network and host resources. In terms of performance, an intrusion detection system becomes more accurate as it detects more attacks and raises fewer false positive alarms."

- **Application of Machine Learning Approaches in Intrusion Detection System: A Survey (Nutan Farah Haq, Musharrat Rafni) –** This paper provides the overview of the research topic and describes a number of techniques for intrusion detection. It also represents a statistical overview of articles over the years on the algorithms that were frequently used, the datasets for each experiment and the consideration of feature selection step. And lastly it represents a statistical overview of articles over the years on the algorithms that were frequently used, the datasets for each experiment and the consideration of feature selection step.

- **A Neural Network Component for an Intrusion Detection System (Herve DEBAR, Monique BECKER, Didier SIBONI) –** In this paper, it is shown that neural networks can be used in an intrusion detection system. A user model is implemented, which is a complement of a statistical modal because neural network cannot handle all the available data adequately. The deviations to the normal behavior of the user seem to be diagnosed fairly quickly by the neural network. This capability is interesting since the goal of an intrusion detection system is to detect a potential intruder as soon as possible. The neural network doesn't solve the dimensioning and stability problems. The model is tested by putting it on a real website.

- **A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms (L.Dhanabal, Dr. S.P. Shantharajah) –** In this paper the analysis of the NSL-KDD data set is made by using various clustering algorithms available in the WEKA data mining tool. The NSL-KDD data set is analyzed and categorized into four different clusters depicting the four common different types of attacks. The analysis results on the NSL-KDD dataset show that it is a best candidate data set to simulate and test the performance of IDS. The CFS method for dimensionality reduction reduces the detection time and increase the accuracy rate. This analysis conducted on the NSL-KDD dataset with the help of figures and tables helps the researcher to have clear understanding of the dataset. It also brings to light that most of the attacks are launched using the inherent drawbacks of the TCP protocol.

- **D-SCIDS: Distributed soft computing intrusion detection system (Ajith Abraham, Ravi Jain) –** This paper presents a framework for Distributed Intrusion Detection Systems (DIDS) using several soft computing paradigms. It informs about the importance of feature reduction to model lightweight intrusion detection systems. It proposes a hybrid architecture involving ensemble and base classifiers for intrusion detection. LGP is the candidate for real time intrusion detection as it can be manipulated at the machine code level. The lightweight SCIDS is useful for distributed systems. The heavy weight are considered ideal for conventional static networks, wireless base stations etc.

- **An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks (Ozgur Depren, Murat Topallar, Emin Anarim) –** This paper proposes a novel Intrusion Detection System (IDS) architecture utilizing both anomaly and misuse detection approaches. This hybrid Intrusion Detection System architecture consists of an anomaly detection module, a misuse detection module and a decision support system combining the results of these two detection modules. The proposed anomaly detection module uses a Self-Organizing Map (SOM) structure to model normal behavior. Deviation from the normal behavior is classified as an attack. The proposed misuse detection module uses J.48 decision tree algorithm to classify various types of attacks. A rulebased Decision Support System (DSS) is also developed for interpreting the results of both anomaly and misuse detection modules.

- **The Architecture of a Network Level Intrusion Detection System (Richard Heady, George Luger) –** This paper presents the preliminary architecture of a network level intrusion detection system. The proposed system will monitor base level information in network packets learning the ‘normal’ patterns and announcing anomalies as they occur. The goal of the research was to determine the applicability of the intrusion detection technology, of that era, to detect network level intrusions.

- **Unsupervised learning techniques for an intrusion detection system (Stefano Zanero, Sergio M. Savaresi) –** The paper describes an innovative model of Anomaly Based Network Intrusion Detection System, totally based on unsupervised learning techniques. It describes the overall architecture of the system and proposed the empirical results on a test implementation. It proposes a two-tier innovative architecture for a network-based anomaly detection IDS, based on unsupervised learning algorithms. in the first tier of the system, an unsupervised clustering algorithm classifies the payload of the packets, observing one packet at a time and \compressing" it into a single byte of information. The second tier algorithm instead takes into consideration the anomalies, both in each single packet and in a sequence of packets. The results are to be considered qualitative until the full architecture of the system is integrated and tested.

- **The DIDS (Distributed Intrusion Detection System) Prototype (Steven R. Snapp, Stephen E. Smaha) –** The paper describes a prototype DIDS which generalizes the target environment in order to monitor multiple hosts connected via a network and the network itself. The DIDS components include the DIDS Director, a single Host Monitor per host, and a single LAN Monitor for each LAN segment of the monitored network. Information is gathered and processed locally by each distributed component, with important events and information transported to, and analyzed at, a central location. This architecture provides the capability to aggregate information from numerous different sources. The system is designed to work with any audit trail for mat as long as certain pieces of critical information are provided by the auditing mechanism.

- **Optimization of Intrusion Detection Systems Determined by Ameliorated HNADAM-SGD Algorithm (Shyla Shyla , Vishal Bhatnagar) –** In this paper, the IDS model is determined using the hybridization of the HNADAMSDG algorithm. The performance of the algorithm is compared with other classification algorithms by adapting feature selection and optimization techniques. The algorithm is used for testing and training of the UNSW-NB15 dataset. The HNADAM-SDG techniques are used to measure the relationship between the population of variables based on collected paired samples. The correlation of variables is determined to further compute the mathematical relationship, to predict the value of one variable based on another variable, and to observe the change in the value of variables. The best fit algorithm has a zero error rate to fit the data points. The error rate of the model varies depending on the size of the training samples. The performance is visualized using learning curves and AUC-ROC curve areas.

- **Haystack: An Intrusion detection system (Stephen E. Smaha) –** Haystack is a prototype system for the detection of intrusions in multi-user Air Force computer systems. Haystack reduces voluminous system audit trails to short summaries of user behaviors, anomalous events, and security incidents. Haystack is designed to be an operational utility for the System Security Officer to reduce enormous quantities of generally obscure audit trail data to short summaries of interpreted information for further investigation of potential computer intrusions. Because of the ambiguity of the data and the variety of possible interpretations for most user behaviors, the SSO remains the critical element in the investigative process. Haystack assists the SSO by providing "hunches," clues, and summaries of relevant data.

- **Intrusion Detection System for Internet of Things based on a Machine Learning approach (Chao Liang, Bharanidharan Shanmugam) –** This research aims to find an effective solution to security issues faced by the network environment of Internet of Things. This paper proposes an intrusion detection system model based on a multi-agent system, using blockchain and deep learning. The flexibility of a multi-agent system means that this new IDS can be used in IoT environments of different sizes. All actions of communication agents will be recorded on blockchain, which makes the system more secure from threats, including information tampering and information disclosure. Based on this model, this paper studies the application of a neural network in intrusion detection systems, and the simulation results show that the deep learning algorithm has a better performance than traditional methods. s. The simulation using the NSL-KDD dataset shows the high accuracy of DNN for intrusion detection on the transport layer of the IoT environment. The performance of the DNN model in distinguishing anomaly from normal is better than other machine learning methods, such as decision trees. Some rare attack types cannot yet be detected with enough accuracy, although the DNN model has a high accuracy rate in distinguishing the more common attack types.

Proposed Method:

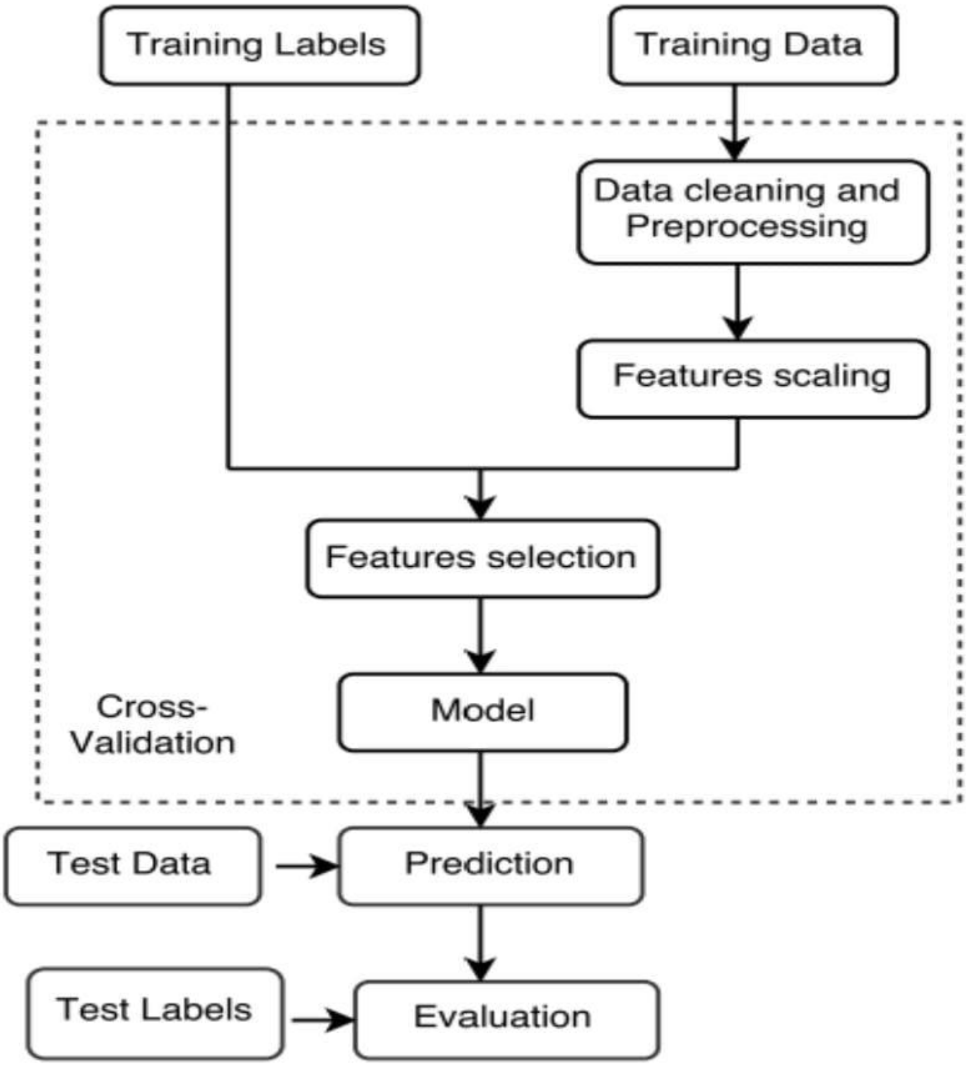
Step 1: Data Processing – All the categorical features are transformed into binary feature using One-Hot-Encoding. Requirement for One-Hot-encoding: "The input to this transformer should be a matrix of integers, denoting the values taken on by categorical (discrete) features. The output will be a sparse matrix where each column corresponds to one possible value of one feature. It is assumed that input features take on values in the range [0, n_values)." Therefore, the features first need to be transformed with LabelEncoder, to transform every category to a number.

Step 2: Feature Scaling – Features scaling is a common requirement of machine learning methods, to avoid that features with large values may weight too much on the final results. For each feature, calculate the average, subtract the mean value from the feature value, and divide the result by their standard deviation. After scaling, each feature will have a zero average, with a standard deviation of one.

Step 3: Feature Selection – Eliminate redundant and irrelevant data by selecting a subset of relevant features that fully represents the given problem. Univariate feature selection with ANOVA F-test. This analyzes each feature individually to detemine the strength of the relationship between the feature and labels. Using SecondPercentile method (sklearn.feature_selection) to select features based on percentile of the highest scores. When this subset is found: Recursive Feature Elimination (RFE) is applied.

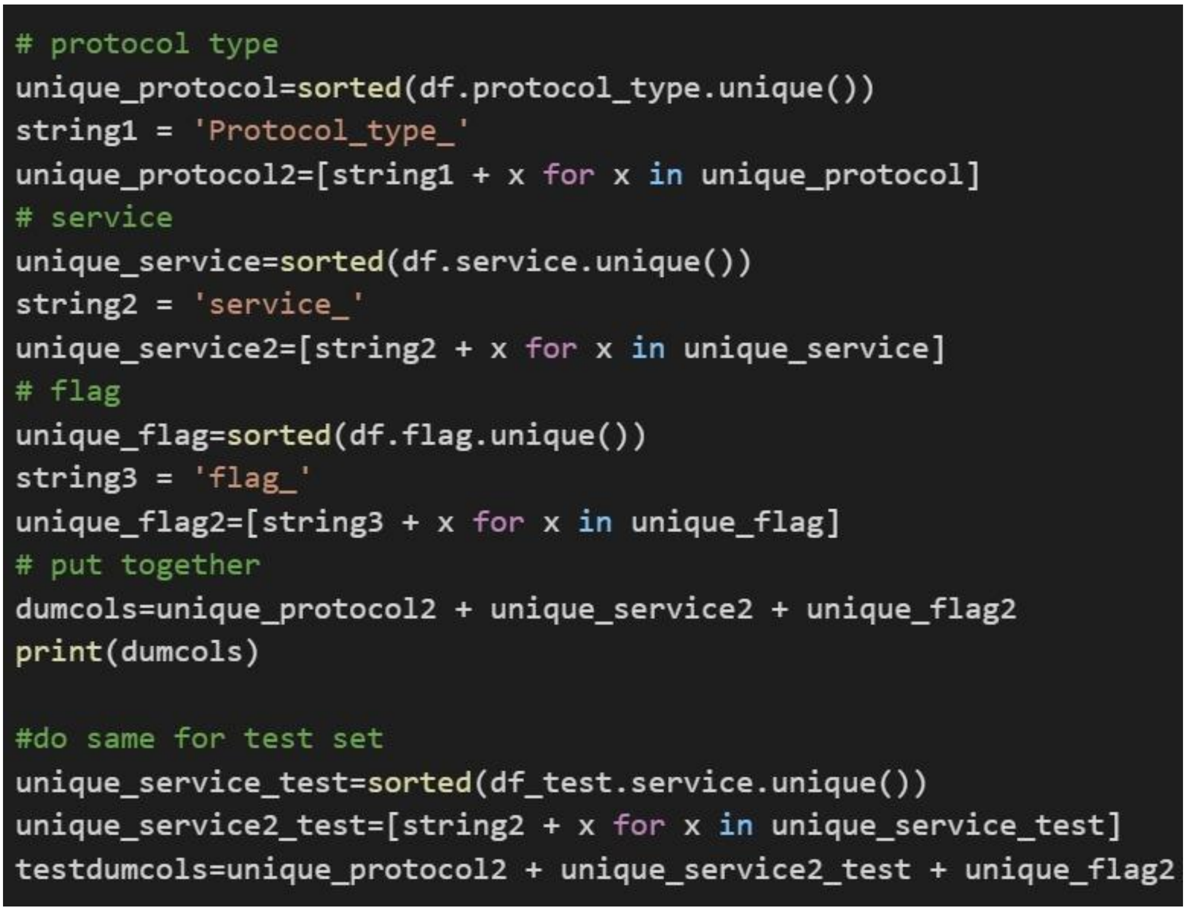
Step 4: Build the model – Decision tree model is built. Classifier is trained for reduced features.

Step 5: Prediction & Evaluation – Using the test data to make predictions of the model. Multiple scores are considered such as: accuracy score, confusion matrix and a cross validation is performed.



Implementation:

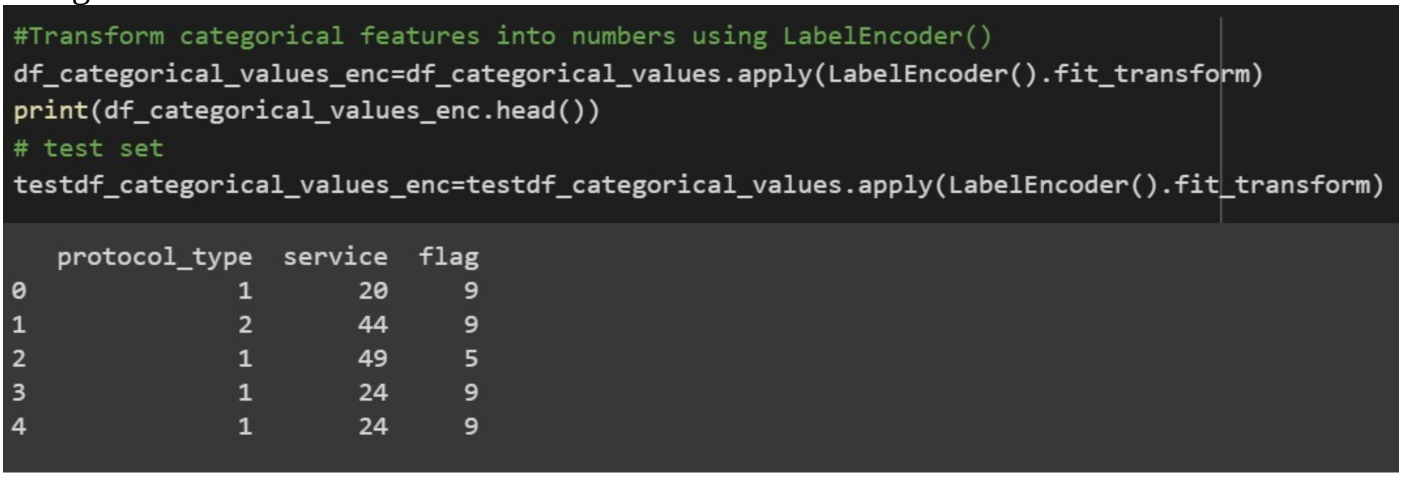
Protocol check –



```
# protocol type
unique_protocol=sorted(df.protocol_type.unique())
string1 = 'Protocol_type_'
unique_protocol2=[string1 + x for x in unique_protocol]
# service
unique_service=sorted(df.service.unique())
string2 = 'service_'
unique_service2=[string2 + x for x in unique_service]
# flag
unique_flag=sorted(df.flag.unique())
string3 = 'flag_'
unique_flag2=[string3 + x for x in unique_flag]
# put together
dumcols=unique_protocol2 + unique_service2 + unique_flag2
print(dumcols)

#do same for test set
unique_service_test=sorted(df_test.service.unique())
unique_service2_test=[string2 + x for x in unique_service_test]
testdumcols=unique_protocol2 + unique_service2_test + unique_flag2
```

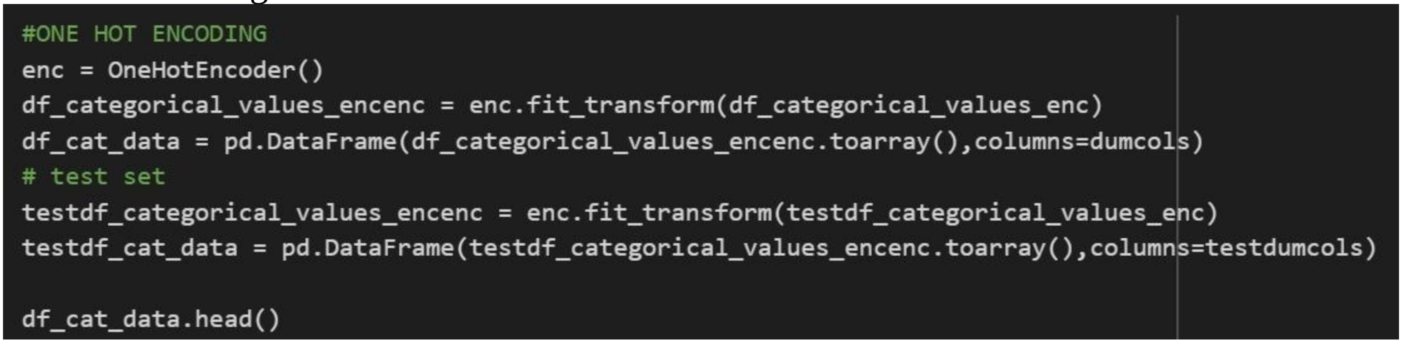
Categorical to numerical –



```
#Transform categorical features into numbers using LabelEncoder()
df_categorical_values_enc=df_categorical_values.apply(LabelEncoder().fit_transform)
print(df_categorical_values_enc.head())
# test set
testdf_categorical_values_enc=testdf_categorical_values.apply(LabelEncoder().fit_transform)
```

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

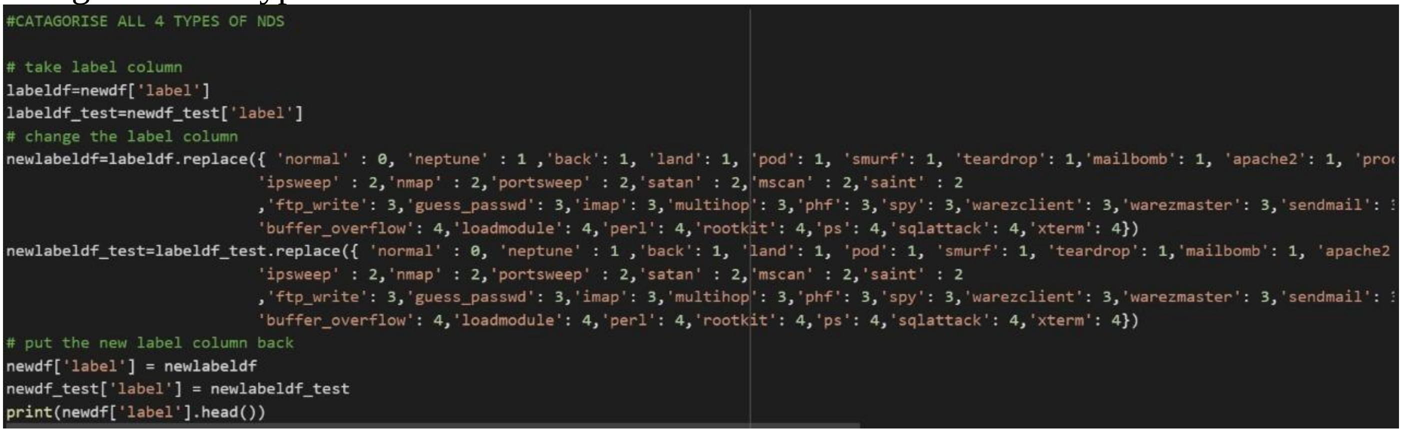
One hot encoding –



```
#ONE HOT ENCODING
enc = OneHotEncoder()
df_categorical_values_encenc = enc.fit_transform(df_categorical_values_enc)
df_cat_data = pd.DataFrame(df_categorical_values_encenc.toarray(),columns=dumcols)
# test set
testdf_categorical_values_encenc = enc.fit_transform(testdf_categorical_values_enc)
testdf_cat_data = pd.DataFrame(testdf_categorical_values_encenc.toarray(),columns=testdumcols)

df_cat_data.head()
```

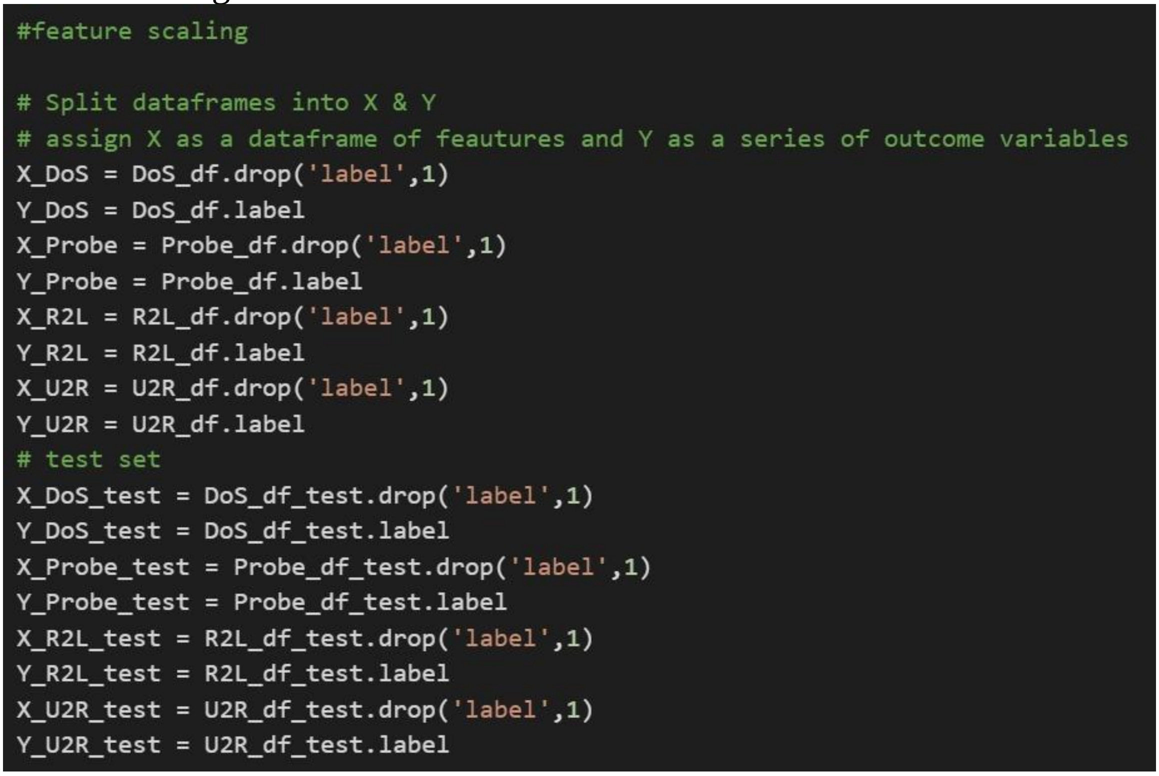
Categorize all 4 types –



```
#CATAGORISE ALL 4 TYPES OF NDS

# take label column
labeldf=newdf['label']
labeldf_test=newdf_test['label']
# change the label column
newlabeldf=labeldf.replace({ 'normal' : 0, 'neptune' : 1 ,'back': 1, 'land': 1, 'pod': 1, 'smurf': 1, 'teardrop': 1,'mailbomb': 1, 'apache2': 1, 'pro
                           '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':
                           'buffer_overflow': 4,'loadmodule': 4,'perl': 4,'rootkit': 4,'ps': 4,'sqlattack': 4,'xterm': 4})
newlabeldf_test=labeldf_test.replace({ 'normal' : 0, 'neptune' : 1 ,'back': 1, 'land': 1, 'pod': 1, 'smurf': 1, 'teardrop': 1,'mailbomb': 1, 'apache2
                           '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':
                           'buffer_overflow': 4,'loadmodule': 4,'perl': 4,'rootkit': 4,'ps': 4,'sqlattack': 4,'xterm': 4})
# put the new label column back
newdf['label'] = newlabeldf
newdf_test['label'] = newlabeldf_test
print(newdf['label'].head())
```

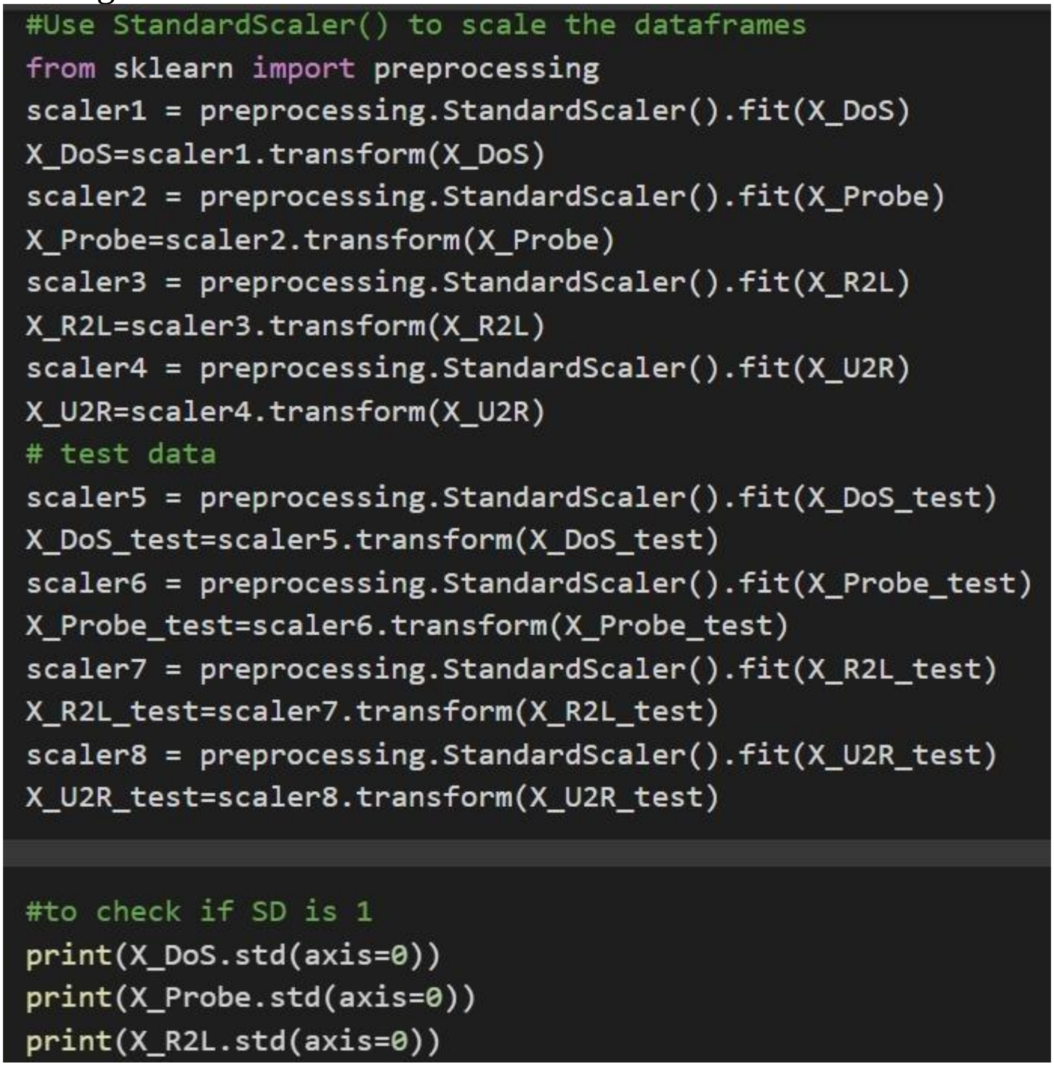
Feature scaling –



```
#feature scaling

# Split dataframes into X & Y
# assign X as a dataframe of feautures and Y as a series of outcome variables
X_DoS = DoS_df.drop('label',1)
Y_DoS = DoS_df.label
X_Probe = Probe_df.drop('label',1)
Y_Probe = Probe_df.label
X_R2L = R2L_df.drop('label',1)
Y_R2L = R2L_df.label
X_U2R = U2R_df.drop('label',1)
Y_U2R = U2R_df.label
# test set
X_DoS_test = DoS_df_test.drop('label',1)
Y_DoS_test = DoS_df_test.label
X_Probe_test = Probe_df_test.drop('label',1)
Y_Probe_test = Probe_df_test.label
X_R2L_test = R2L_df_test.drop('label',1)
Y_R2L_test = R2L_df_test.label
X_U2R_test = U2R_df_test.drop('label',1)
Y_U2R_test = U2R_df_test.label
```

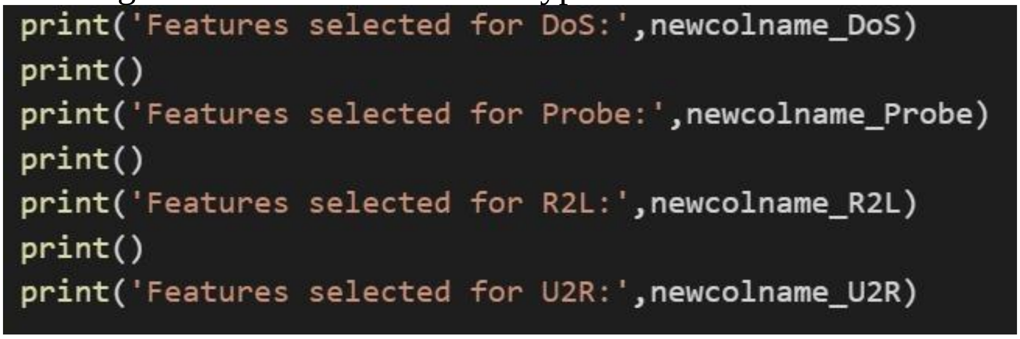
Scaling dataframe –



```
#Use StandardScaler() to scale the dataframes
from sklearn import preprocessing
scaler1 = preprocessing.StandardScaler().fit(X_DoS)
X_DoS=scaler1.transform(X_DoS)
scaler2 = preprocessing.StandardScaler().fit(X_Probe)
X_Probe=scaler2.transform(X_Probe)
scaler3 = preprocessing.StandardScaler().fit(X_R2L)
X_R2L=scaler3.transform(X_R2L)
scaler4 = preprocessing.StandardScaler().fit(X_U2R)
X_U2R=scaler4.transform(X_U2R)
# test data
scaler5 = preprocessing.StandardScaler().fit(X_DoS_test)
X_DoS_test=scaler5.transform(X_DoS_test)
scaler6 = preprocessing.StandardScaler().fit(X_Probe_test)
X_Probe_test=scaler6.transform(X_Probe_test)
scaler7 = preprocessing.StandardScaler().fit(X_R2L_test)
X_R2L_test=scaler7.transform(X_R2L_test)
scaler8 = preprocessing.StandardScaler().fit(X_U2R_test)
X_U2R_test=scaler8.transform(X_U2R_test)
```

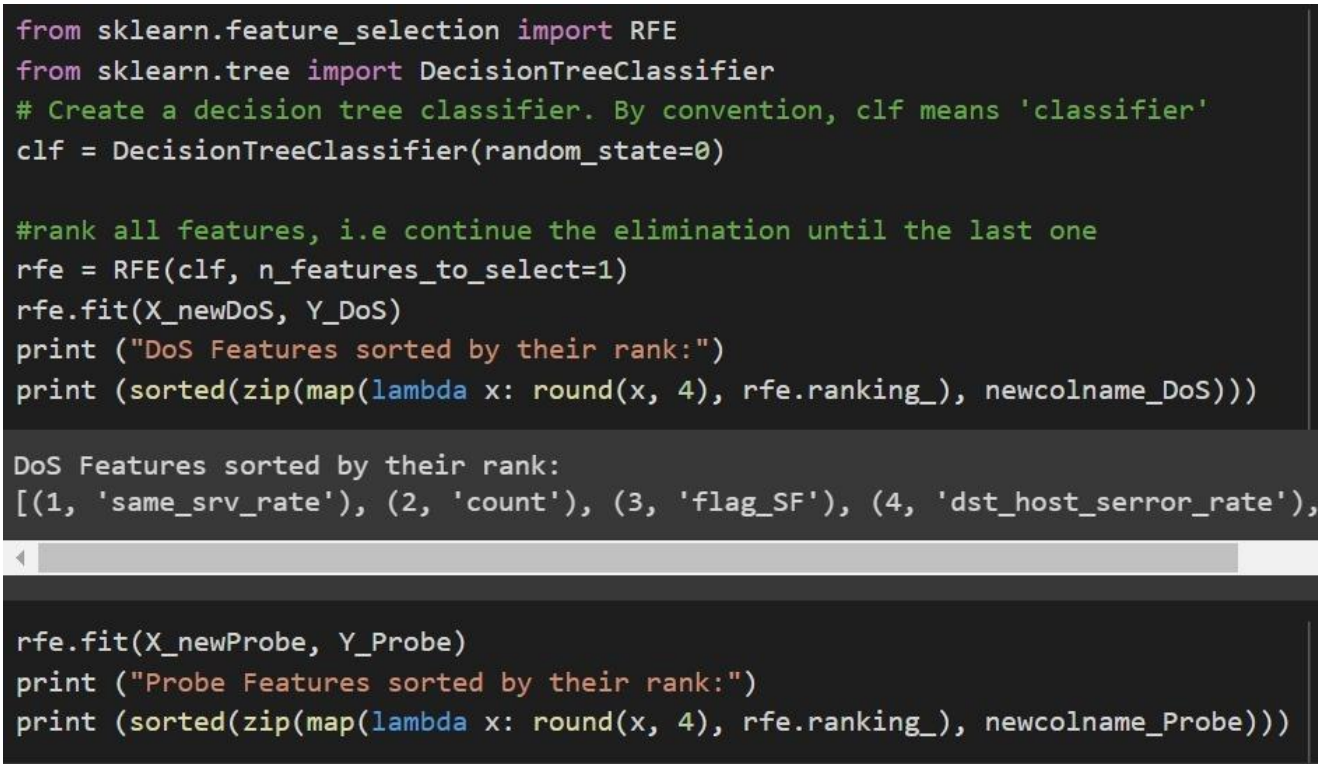
```
#to check if SD is 1
print(X_DoS.std(axis=0))
print(X_Probe.std(axis=0))
print(X_R2L.std(axis=0))
```

Showing selected features for all 4 types of attack –



```
print('Features selected for DoS:',newcolname_DoS)
print()
print('Features selected for Probe:',newcolname_Probe)
print()
print('Features selected for R2L:',newcolname_R2L)
print()
print('Features selected for U2R:',newcolname_U2R)
```

Decision tree classifier –



```
from sklearn.feature_selection import RFE
from sklearn.tree import DecisionTreeClassifier
# Create a decision tree classifier. By convention, clf means 'classifier'
clf = DecisionTreeClassifier(random_state=0)

#rank all features, i.e continue the elimination until the last one
rfe = RFE(clf, n_features_to_select=1)
rfe.fit(X_newDoS, Y_DoS)
print ("DoS Features sorted by their rank:")
print (sorted(zip(map(lambda x: round(x, 4), rfe.ranking_), newcolname_DoS)))
```

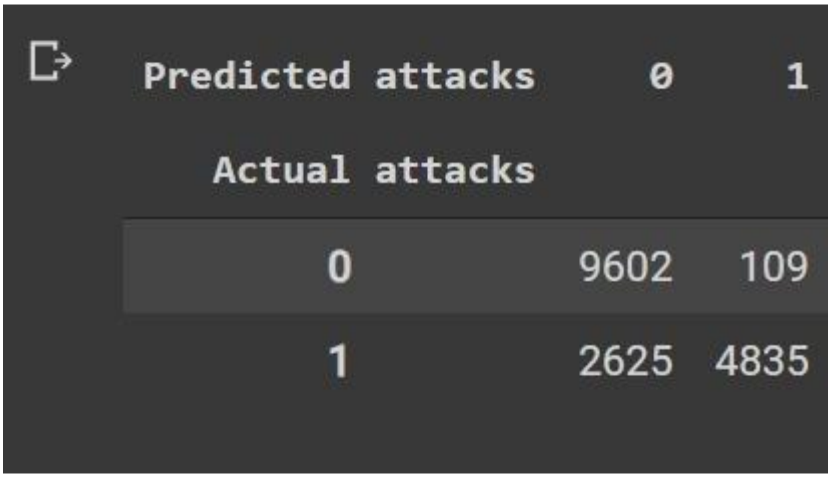
```
DoS Features sorted by their rank:
[(1, 'same_srv_rate'), (2, 'count'), (3, 'flag_SF'), (4, 'dst_host_serror_rate'),
```

```
rfe.fit(X_newProbe, Y_Probe)
print ("Probe Features sorted by their rank:")
print (sorted(zip(map(lambda x: round(x, 4), rfe.ranking_), newcolname_Probe)))
```

Results:

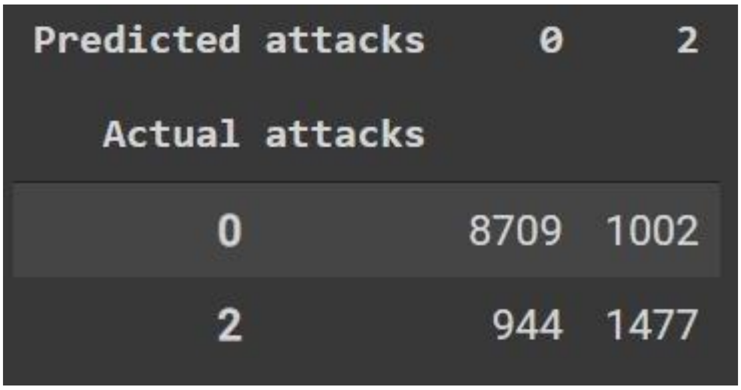
Confusion Matrices for each attack with 13 feature selection –

a. DOS



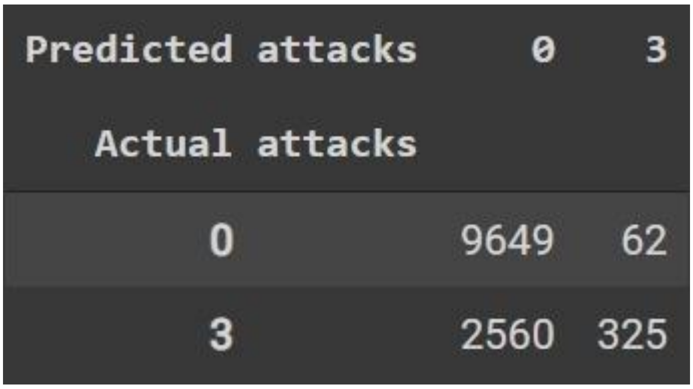
```
Predicted attacks     0     1
   Actual attacks
                0  9602   109
                1  2625  4835
```

b. PROBE



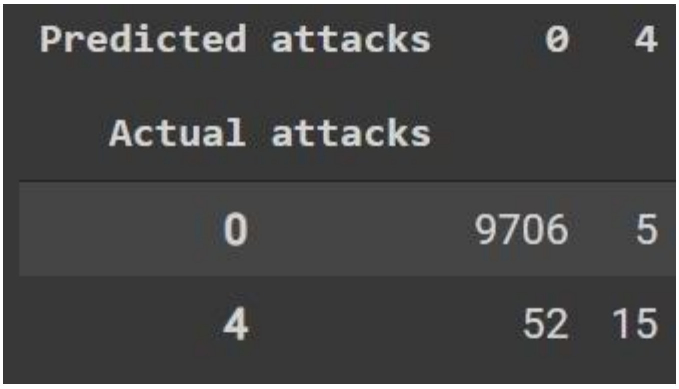
Predicted attacks	0	2
Actual attacks		
0	8709	1002
2	944	1477

c. R2L



Predicted attacks	0	3
Actual attacks		
0	9649	62
3	2560	325

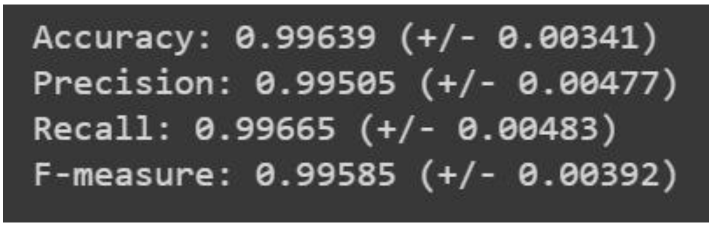
d. U2R



Predicted attacks	0	4
Actual attacks		
0	9706	5
4	52	15

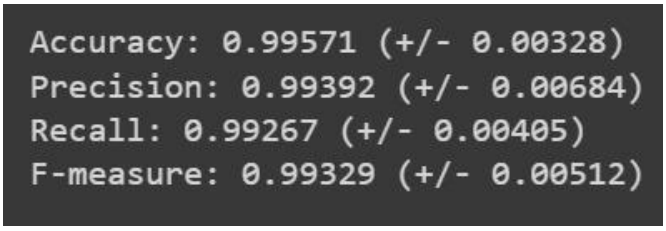
Accuracy –

a. DOS



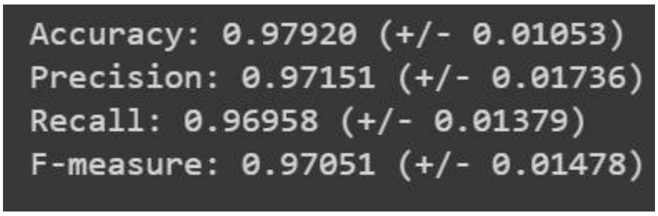
```
Accuracy: 0.99639 (+/- 0.00341)
Precision: 0.99505 (+/- 0.00477)
Recall: 0.99665 (+/- 0.00483)
F-measure: 0.99585 (+/- 0.00392)
```

b. PROBE



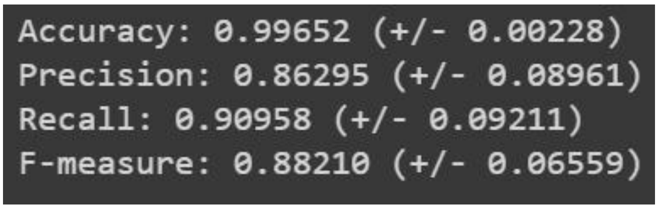
```
Accuracy: 0.99571 (+/- 0.00328)
Precision: 0.99392 (+/- 0.00684)
Recall: 0.99267 (+/- 0.00405)
F-measure: 0.99329 (+/- 0.00512)
```

c. R2L



```
Accuracy: 0.97920 (+/- 0.01053)
Precision: 0.97151 (+/- 0.01736)
Recall: 0.96958 (+/- 0.01379)
F-measure: 0.97051 (+/- 0.01478)
```

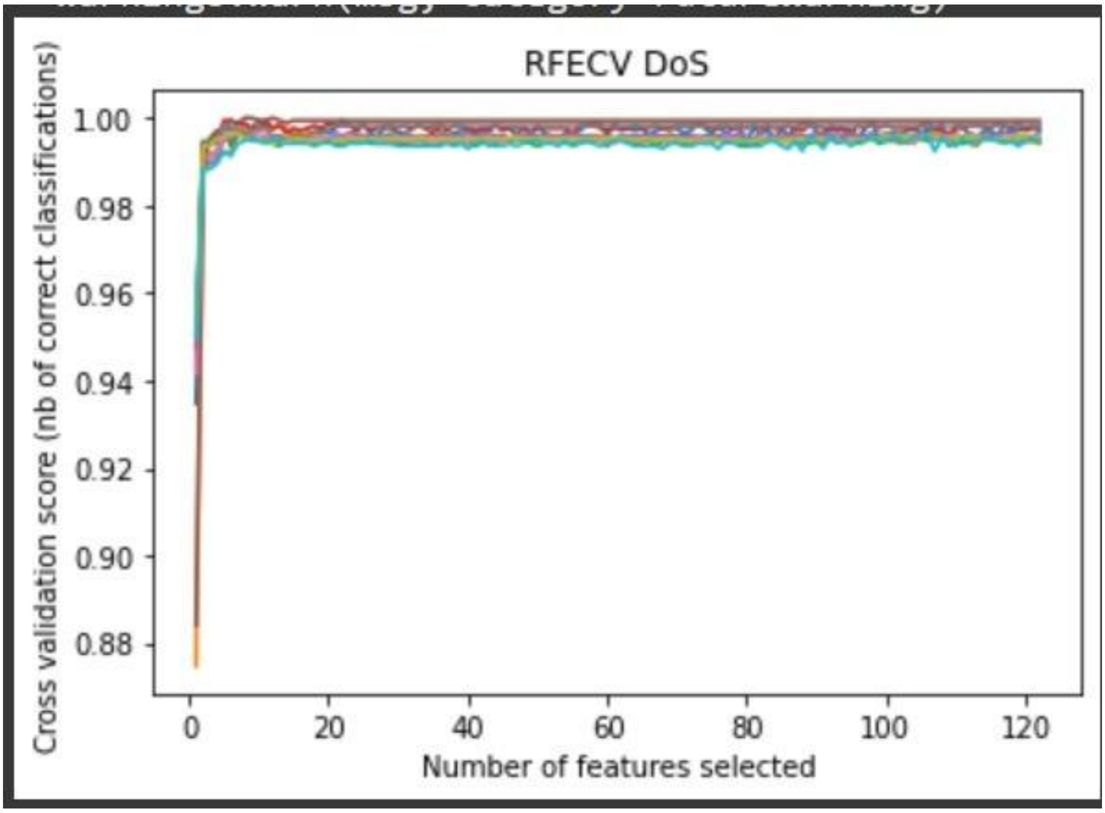
d. U2R



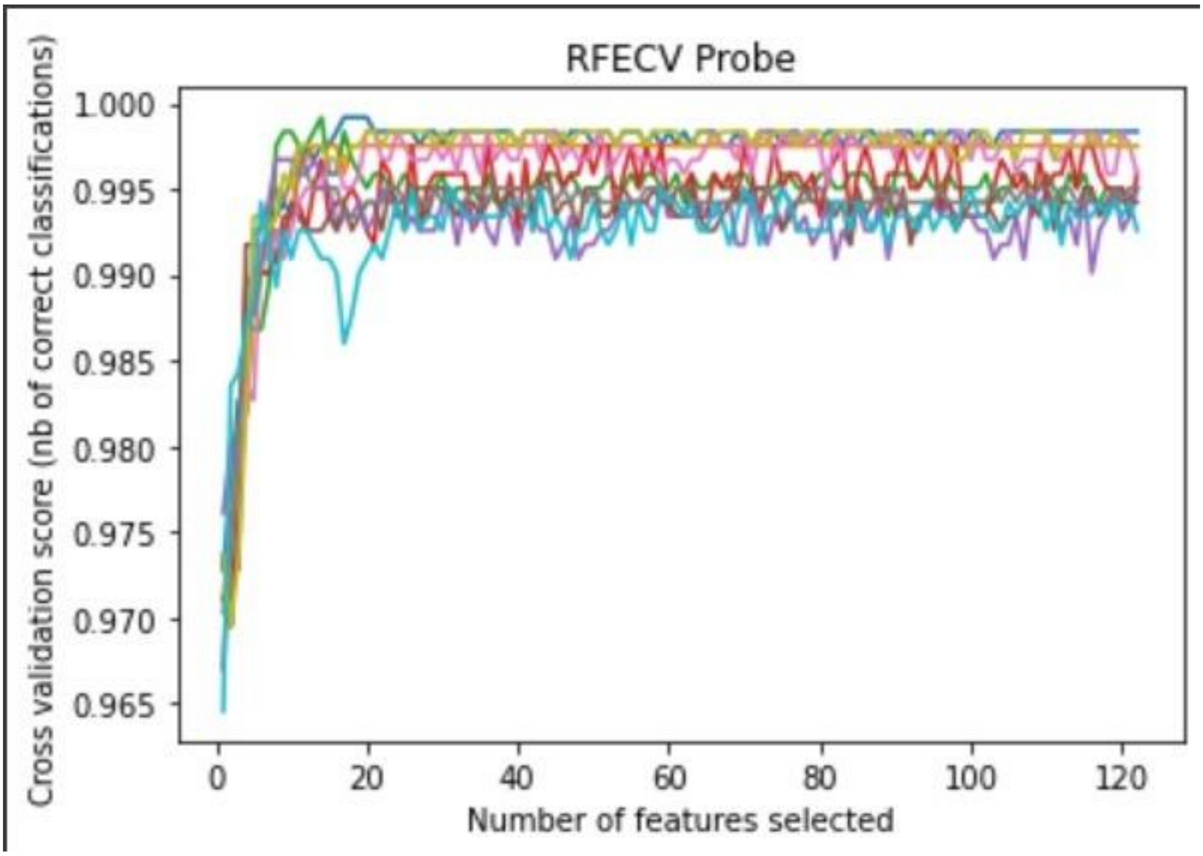
```
Accuracy: 0.99652 (+/- 0.00228)
Precision: 0.86295 (+/- 0.08961)
Recall: 0.90958 (+/- 0.09211)
F-measure: 0.88210 (+/- 0.06559)
```

Recursive Feature Elimination Cross Validation Graphs for each attack –

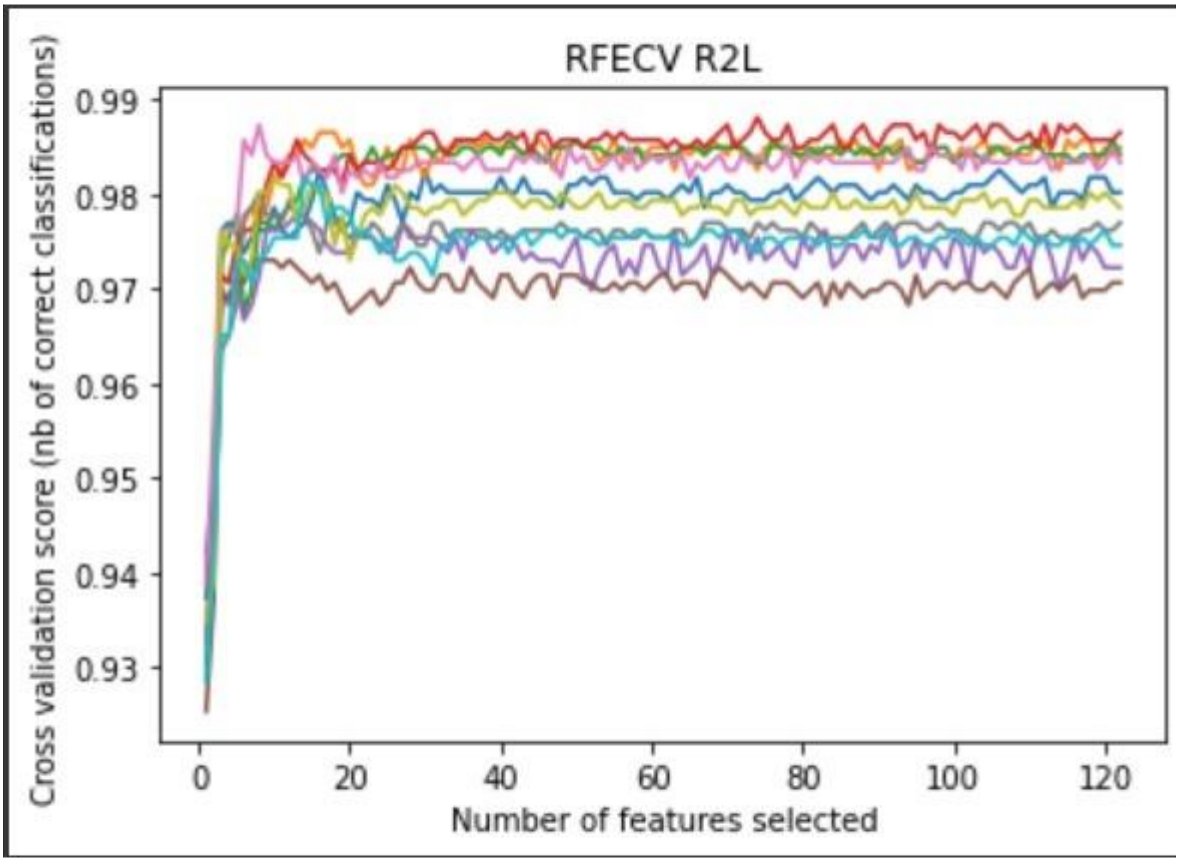
a. DOS



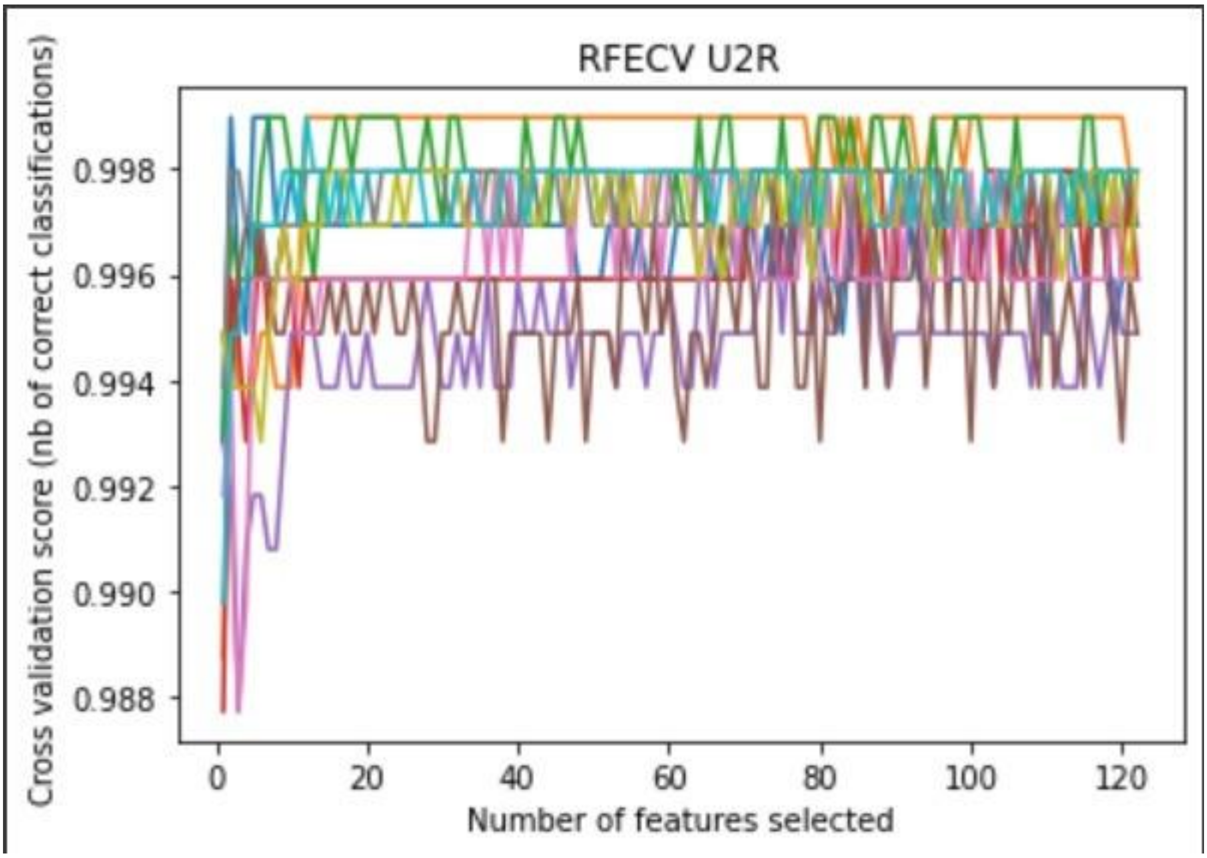
b. PROBE


RFECV Probe
Cross validation score (nb of correct classifications)
1.000
0.995
0.990
0.985
0.980
0.975
0.970
0.965
0
20
40
60
80
100
120
Number of features selected

c. R2L


RFECV R2L
Cross validation score (nb of correct classifications)
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0
20
40
60
80
100
120
Number of features selected

d. U2R



Conclusion:

We have successfully implemented an intrusion detection system using feature selection and decision tree classification algorithm which in case of a network intrusion detects the following attacks with the given accuracy –

- DOS with an accuracy of 99.6
- Probe attack with an accuracy of 99.5
- R2L with an accuracy of 97.9
- U2R with an accuracy of 99.6

References:

- Mohammad Sazzadul Hoque1 , Md. Abdul Mukit and Md. Abu Naser Bikas, "An Implementation of Intrusion Detection System Using Genetic Algorithm", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.2, March 2012.
- SandhyaPeddabachigari AjithAbraham crina Grosan , jhoson Thomas, **"**Modeling intrusion detection system using hybrid intelligent systems**",** Volume 30, Issue 1, January 2007
- Sharmila Kishor Wagh. Vinod K. Pachghare, Satish R. Kolhe, "Survey on Intrusion Detection System using Machine Learning Techniques ", Volume 78 – No.16, September 2013
- Quamar Niyaz, Weiqing Sun, Ahmad Y Javaid, and Mansoor Alam, "A Deep Learning Approach for Network Intrusion Detection System", ACM, 24 May 2016

- NabilaFarnaaz, M.A.Jabb,ar, “Random Forest Modeling for Network Intrusion Detection System”, Procedia Computer Science, Volume 89, 2016
- Asmaa Shaker Ashoor, “Importance of Intrusion Detection System (IDS)”, International Journal of Scientific & Engineering Research, Volume 2, 2010
- Nutan Farah Haq, Abdur Rahman Onik, Md. Avishek Khan Hridoy, “Application of Machine Learning Approaches in Intrusion Detection System: A Survey”, International Journal of Advanced Research in Artificial Intelligence(ijarai), Volume 4 Issue 3, 2015.
- Hervé Debar, M. Becker, D. Siboni, “A neural network component for an intrusion detection system”, IEEE Computer Society Symposium on Research in Security and Privacy, 06 August 2002
- L. Dhanabal , S. Shantharajah, “A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms”, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2015
- Ajith Abraham, Ravi Jain, Sugata Sanyal & Sang Yong Han, “SCIDS: A Soft Computing Intrusion Detection System”, volume 3326, International Workshop on Distributed Computing.
- Ozgur Depren, Murat Topallar, Emin Anarim, M Kemal Ciliz, “”, An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks, DBLP, Volume 29, Issue 4
- R. Heady, G. Luger, A. Maccabe, M. Servilla, “The architecture of a network level intrusion detection system”, 1990
- Stefano Zanero, Sergio M. Savaresi, “Unsupervised Learning Techniques for an Intrusion Detection System”, ACM symposium on Applied computing, 2004
- Steven R. Snapp, S. Smaha, Daniel M. Teal, T. Grance, “The DIDS (Distributed Intrusion Detection System) Prototype”, USENIX Summer, 42062636, 1992
- Shyla Shyla, Vishal Bhatnagar, Vikram Bali, Shivani Bali, “Optimization of Intrusion Detection Systems Determined by Ameliorated HNADAM-SGD Algorithm”, MDPI, 2022
- Hervé Debar, “An Introduction to Intrusion-Detection Systems”, 2009
- Chao Liang; Bharanidharan Shanmugam; Sami Azam; Mirjam Jonkman; Friso De Boer; Ganthan Narayansamy, “Intrusion Detection System for Internet of Things based on a Machine Learning approach”, International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), 2019
- Hesham Altwaijry, Saeed Algarny, “Bayesian Based Intrusion Detection System”, Elsevier B.V. , 2011
- Mohammad Almseidin, Maen Alzubi, Szilveszter Kovacs and Mouhammd Alkasassbeh, “Evaluation of Machine Learning Algorithms for Intrusion Detection System”, 2018
- Dr. Ahmad Y. Javaid, Quamar NiyazDr. Weiqing SunDr. Mansoor Alam, “A Deep Learning Approach for Network Intrusion Detection System”, 2015

THE END