International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8, Issue-1, May 2019

An Experimental Analysis and Performance Evaluation of Machine Learning Classifiers using Benchmark Dataset

Sukhvinder Singh, S K V Jayakumar

Abstract: ***In the world of system automation, organizations are really struggling to prevent/identify the different types of attacks or suspicious activities happening in the network because interconnected networks are widely spread across the globe. The organizations must have a virtuous traffic analysis system for fast identification of malicious traffic and isolating the attacker at the early stage, which demands an experimentally proven and analyzed robust traffic classification technique. There are many methods available to classify the traffic such as Port Number Based, Flow-Based and Packet-Based. Machine Learning is one of the state-of-the-art tools to solve classification problems in the real world. As it can attain better results as compared to traditional methods. In this paper, various variants of Machine Learning techniques under Support Vector Machine (SVM), Decision Tree and Ensemble is discussed. Further, Principal Component Analysis (PCA) is used for the dataset feature extraction and entrenched into the classifiers for traffic classification. The experimental analysis illustrates the highest accuracy rate of 99.7% for traffic classification by Bagged Tree classifier one of the variants of Ensemble as compared to other variants of SVM, Decision Tree, and Ensemble Classifiers.***

Index Terms**: *Decision Tree, Denial of Service, Ensemble, Machine Learning, Principal Component Analysis, Support Vector Machine.***

I. INTRODUCTION

In the 21st century, the exchange of information between different organizations and entities which may or may not be geographically dispersed is a regular need to fulfill the business objectives. The data should always originate and merge at authentic source and destination respectively. The information or data is exchanged over a public network infrastructure. This exchange of information using public channels creates security loopholes in the network. The organizations are facing real-time problems to prevent their network from the intrusion activities and it has become very difficult to classify the original data traffic and attack traffic coming into the network. This gives rise to the need for analyzing the traffic over the network.

Revised Manuscript Received on 30 May 2019.
***** Correspondence Author
Sukhvinder Singh*, School of Engineering and Technology, Department of Computer Science, Pondicherry University, Puducherry, India.
S K V Jayakumar, School of Engineering and Technology, Department of Computer Science, Pondicherry University, Puducherry, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

In the traditional network, various systems were designed to detect attack traffic in the network; one among these is an Intrusion Detection System (IDS). The IDS is used to identify the suspicious activities in the network and collects the information about the attacker and the victim system. It is generally of two types Behavioral Based and Signature-Based IDS[1]. Behavior-Based IDS also referred to as Statistical IDS, profile-based IDS (anomaly detection) and heuristics-based IDS[2][3]. It monitors normal activities and events on the system and scans for abnormal activities or events that are considered to be malicious activities. Working of Signature-Based IDS revolves around the database of known attack signatures. It examines the data and compares it with a signature pattern in the signature database. If an incident matches a signature, the IDS registers that an attack has happened and responds with an alert, alarm for necessary alteration to firewall configuration. With every pro there is a con, same is the case with Signature-based IDS. It works effectively when the vicinity of known attack methods is observed. But upgraded or altered versions of known attacks are often left undetected by this IDS. Therefore, Signature-based IDS is found effective only if its database is kept updated[4].To prevent malicious traffic entering into the network, the classification of incoming heterogeneous traffic flow needs to be analyzed properly. The classification techniques take the network traffic as input, analyzes it and help to identify the relevant features. The extracted features include Port Numbers, Type of Protocols, and Applications used in the network for various possible attacks[5]. Based on the traffic classification various actions like control, discovery, monitoring, and optimization can be done to improve the network performance. In this paper different variants of three Machine Learning techniques are used for the analysis of network traffic data as shown in Fig1. These Machine Learning classifiers are validated for the attack traffic classification using KDD cups dataset. In the first step, 20 features out of 42 features are extracted using the Principal Component Analysis (PCA)[6]. Later, these features are fed into different classifiers for evaluation and results. A tree structure in Fig1 shows the machine learning classifiers and its variants used in the experiment. The objective of this work is to find the superlative Machine Learning Technique for accurate identification of different attack flow and normal flow classes. Rest of the paper is organized as follows: Section II background of traffic classification techniques used earlier and discussion on Machine Learning Techniques used in the paper; Section III contains the description of Dataset; Section IV contains the detailed description on Experimental Analysis and Discussion; Section V reveals the Conclusion and Future Directions.

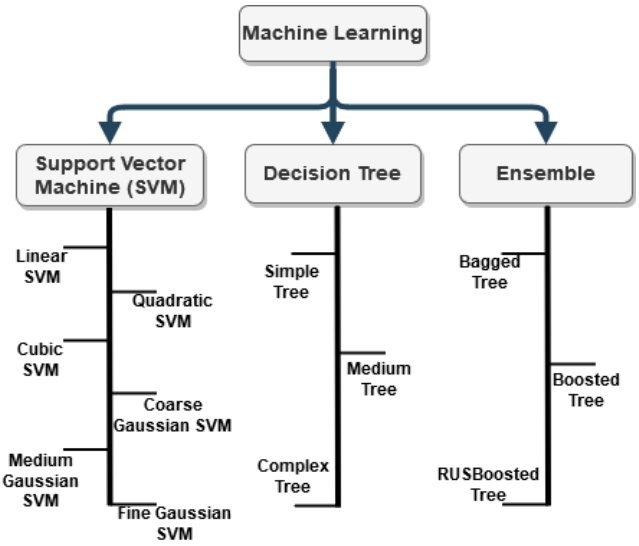


Fig 1 Variants of Machine Learning Techniques

II. BACKGROUND

In the backgrounds first, we will discuss the related work and then we discuss the Machine Learning Classifiers used for the classification of traffic flow[7].

A. Related Work

There are two kinds of DDoS detection techniques found namely, signature-based and anomaly-based detection techniques[8]. In Signature-based detection technique network behaviors are used whereas in anomaly-based detection the machine-learning techniques are used. The machine learning techniques commonly used for DDoS attack detection are an artificial neural network (ANN), Support Vector Machine (SVM), Fuzzy Logic, Decision tree, Evolutionary algorithm, Navies Bayes, and k-means clustering algorithms. The ANN is the one which is being used to detect both unknown and known DDoS attacks by the researchers [7]. In [7], it has been found that SVM learning can be used to prevent serious damages to the controllers. Two-phased machine learning classifier has been used in [9]. In the first phase, unsupervised learning has been used to collect the flow records of fifteen Internet applications. K-means clustering is brought in to find the unique flows generated by the applications. In the second phase, C5.0, decision tree data classes are used to train and test the derived flow, which will help to derive the flow. An Advance Support Vector Machine (ASVM) technique was proposed which is an enhancement of older Support Vector Machine (SVM) algorithm for attack detection [10]. Using this proposed ASVM the two types of flooding-based DDoS attacks were successfully detected with reduced training and testing time (i.e 97% faster) using volumetric and asymmetric features. In[9], DARPA dataset has been analyzed using SVM classifier with other classifiers for DDoS detection. It ensures the attack detection in the SDN controller with SVM classifier will have high accuracy and less false positive rate. In [10] Improved SVM was proposed for the multiclass classification problem. Here, multiclass problems were decomposed into a number of binary class problems. Each feature under binary class has assigned its own weights according to its importance for classification and results showed that this improved SVM is able to produce more stable performance than vanilla SVM. The mitigation strategies of malicious or suspicious traffic and its limitations in the Software Defined Network were presented in which traffic is classified into different class on the basis of the methodologies used for handling malicious traffic [13].

B. Machine Learning approach

Machine Learning techniques work well for Classification problems and outperform other handcrafted techniques[11]. In this work, traffic classification for Attack detection has been done using three Machine learning techniques. Techniques include SVM, Decision Tree and Ensemble[12][13][14]

1) Support Vector Machine

SVM is a supervised Machine Learning algorithm. It is a critical classifier correctly defined by a separating hyperplane. SVM is having various kernel function, based on this kernel function the SVM interpretation ability changes. Six kernel functions are used in our work i.e Linear, Quadratic, Cubic, Fine Gaussian, Medium Gaussian, and Coarse Gaussian

Linear SVM - SVM linear model is used to solve the classification problems. It creates a simple line to separate the data into classes. It is easy for the interpreter but it is less flexible and having low predictive accuracy.

Quadratic, Cubic SVM - It is nonlinear kernel and very hard to interpret but it can be more accurate than linear.

Fine Gaussian SVM- It allows rapid variations in the response function. Here, the kernel function can be scaled to sqrt(P)/4, where P is the number of predictors.

Medium Gaussian SVM– It gives a less flexible in the response function. The kernel scaling is set to sqrt(P).

Coarse Gaussian – It gives a rigid response function and kernel scaling is set to sqrt(P)*4.

2) Decision Tree

It is a tool for the decision support system that uses tree-based modeling for better decision making. It is easy to interpret, fast for fitting and prediction problems. Low memory is needed for training and testing. But the decision trees can also have low prediction rates based on the problem. The three types of Decision tree classifiers are used here; Simple Tree, Medium Tree, and Complex Tree

Simple Tree - Simple tree suffers from overfitting problem. To prevent this, the horizontal growth of the simple tree is encouraged. This tree is very flexible, using a few leaves we can distinguish between classes. Maximum 4 number of splits are permissible in this.

Medium Tree - In this, the different classes are distinguished using a maximum of 20 splits.

Complex Tree – Here, the tree size is huge and the model flexibility is also high. It does more finer distinctions between classes because many leaves are used here. In this maximum, 100 numbers of splits can be used.

3) Ensemble

This Model combines the results from many weak learners into one high-quality ensemble model. This has three model types namely RUSBoosted Tree, Boosted Tree, Bagged Tree

RUSBoosted Tree - This is called hybrid boosting algorithm used for random under sampling, it is also accepted as an advance data sampling method which improves the performance of the week classifiers.
Boosted Tree - Boosting is one of the classical methods for creating an ensemble, it refers to a group of algorithms that utilize weighted averages to make weak learners into stronger learners. Boosting builds a series of different trees in a stepwise manner and then find the optimal tree using an arbitrary differentiable loss function.
Bagged Tree -To reduce the variance of a decision tree a Bagging tree is used. It creates several data subsets by selecting data randomly from the training sample. After that, each subset is used to train their decision tree and ensemble of different models are produced. The average of all the prediction is taken as a single decision tree.

III. INSIGHT OF DATASET USED

KDD Cup 1999 Data Dataset is available in machine learning repository. It is having multivariate data with categorical and integer values. The total number of 4000000 instances has 42 number of attributes[15].

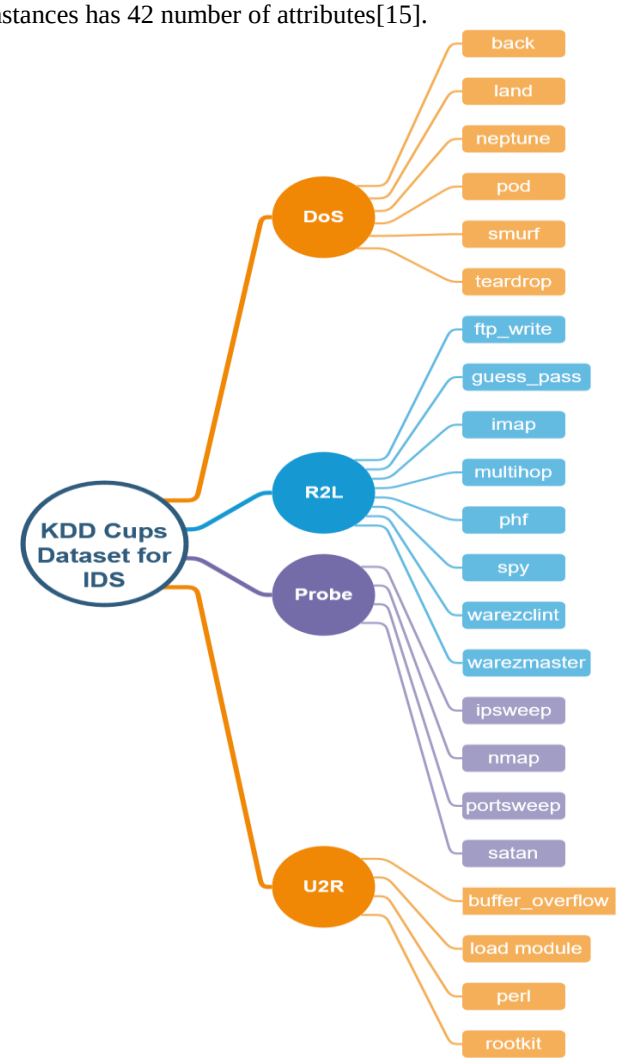


Fig 2 KDD Dataset attack category mapping

The dataset contain total 22 training attack types(i.e back dos, buffer overflow, ftp write, guess passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster)with an additional 14 types in the test data only.

These 22 attack types are mapped into four classes that are DoS, R2L, U2R, Probe. Fig 2 shows that the total 6/22 attack types are mapped as DoS attack, 8/22 attack types are mapped as R2L (Remote to Local), 4/22 attack types are mapped as Probe, 4/22 attack types are mapped as U2R (User to Remote).

KDD Dataset features are divided into three categories i.e basic features, content features and traffic features[16]. Basic features are the type of protocols, number of byte flow, etc. content features a number of failed logins/successful logins, number of guest login, etc. traffic features like number of connections to the identical node as the present connection in the previous two seconds, etc.

IV. EXPERIMENTAL ANALYSIS

The laptop used for the experiment is Core i5 Intel processor having 4GB RAM installed with Matlab2017b on Windows Operating system. The KDD CUP Dataset was downloaded from http://kdd.ics.uci.edu/databases/kddcup99/kddcup99. html[17]. For the experimental purpose, the dataset size is reduced to 25244 instances, having all the possible attack type available in the dataset.

The dataset is imported into the Matlab2017b for preprocessing. After prepossessing Principal Component Analysis (PCA) is used for feature extraction. By using PCA the dataset dimensionality is reduced and selected the feature subsets of the original dimensions X1, X2, X3…Xd. From these subsets of features the good class “predictors” are selected.

PCA will construct a new set of dimensions E1, E2, E3……Em with the combination of original X1, X2, X3…..Xd. It helps defining two things, firstly direction of the greatest variability in the data and secondly perpendicular to first, the highest inconsistency of whatever data is left until the original dimensionality d.

$$m<<d$$

here, d components become m.

In this experiment, 20 extracted features using PCA is rooted in the classifiers i.e SVM, Decision Tree, and Ensemble for traffic classification.

A. Model Types and Results

Table I Accuracy of SVM Classifiers

Classifier Model type	Kernel Function	Training time (in sec)	Prediction time (obs/sec)	Accuracy
Linear SVM	Linear	396.86	**5800**	97.8
Quadratic SVM	Quadratic	337.52	5000	99.1
Cubic SVM	Cubic	6327.2	3600	**99.3**
Fine Gaussian SVM	Gaussian	306.6	2800	99.1
Coarse Gaussian SVM	Gaussian	**300.31**	3100	99.1
Medium Gaussian SVM	Gaussian	318.34	2100	99.1

Table I shows model types and its results obtained using different SVM machine learning classifiers. The best results with respect to training time, prediction time, and acurracy is highlighted.

Using the SVM classifiers model, the Cubic SVM is found better with respect to the rate of accuracy as shown as a graph in Fig 5. As far as the training time is concern Fig4 shown that the Coarse Gaussian SVM is taking less time to train the model i.e 300.31 sec as compared to other classifiers as shown in Fig 4. Fig 3 shows that the prediction time of Linear SVM is 5800 obs/sec which is better than other SVM classifiers.

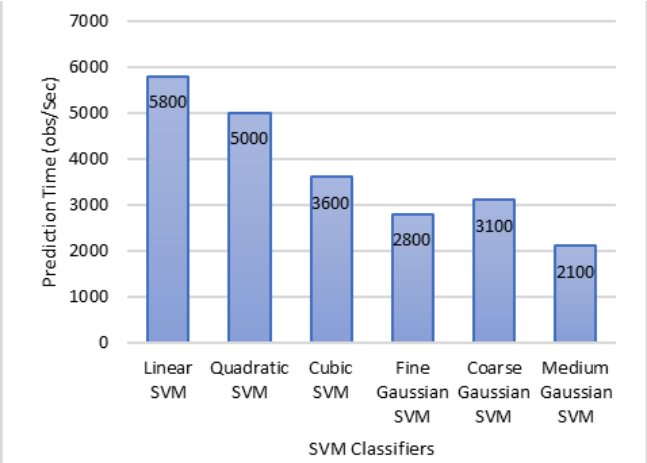


Fig. 3 Prediction time of SVM Classifier Models

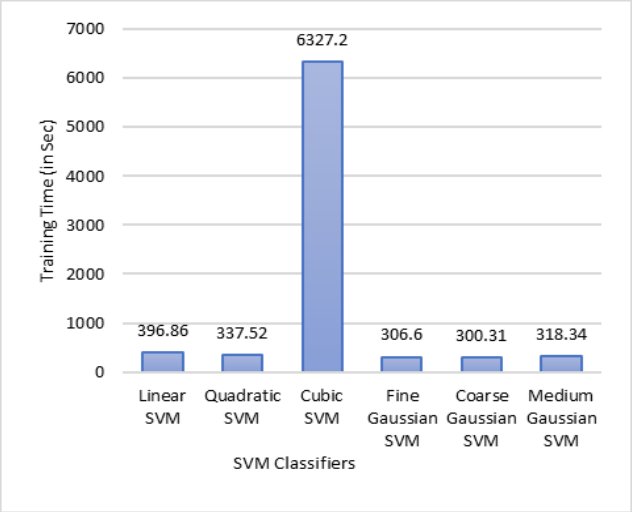


Fig. 4 Training time of SVM Classifier Models

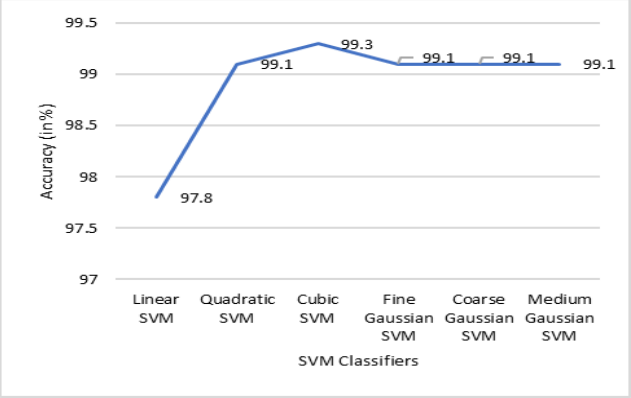


Fig 5 Accuracy of SVM Classifier Models

Table II Accuracy of Decision Tree Classifiers

Classifier Model Type	Maximum No. of Splits	Training time (in sec)	Prediction time (obs/sec)	Accuracy in %
Simple Tree	4	**19.642**	**19000**	94
Medium Tree	20	25.052	15000	98
Complex Tree	100	83.262	12000	**99.4**

Table II shows the Machine Learning technique for classification used is the Decision Tree. In decision tree splitting of tree criteria are important. The tree splitting criterion used in this model is Gini's diversity index. For simple tree maximum 4 splits were taken, for medium 20 splits and for complex tree total 100 splits were carried out.

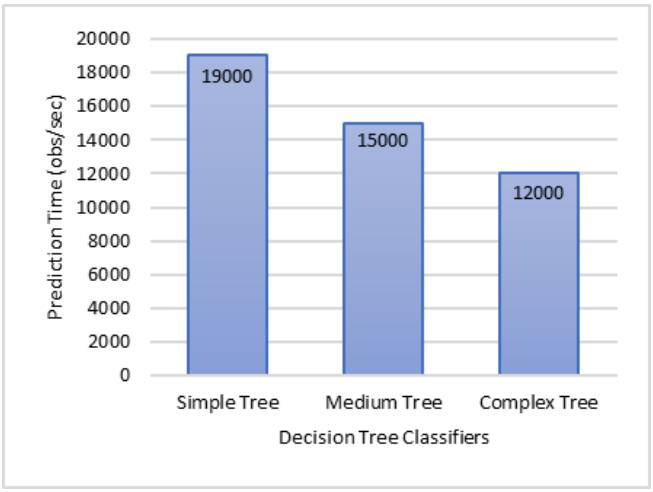


Fig 6 Prediction time of Decision Tree Classifier Models

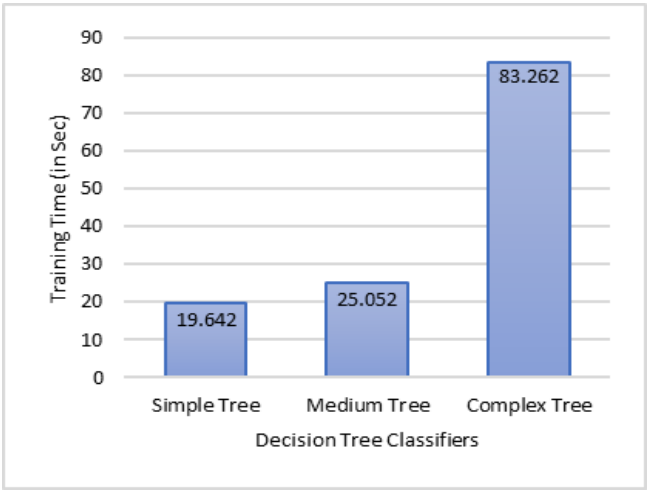


Fig 7 Training time of Decision Tree Classifier Models

Fig 8 Accuracy of Decision Tree Classifier Models

Fig 6 shows the prediction time of complex tree is better than the other two models i.e 19000 obs/sec. The training speed of simple tree is faster i.e 19.642 sec is taken to train the model as shown in Fig7. Using these three classifiers it is found that the Complex Tree accuracy is 99.4% as shown in Fig 8 which is a better result in terms of accuracy percentage w.r.t other two classifiers i.e Simple Tree and Medium Tree having 94% and 98% respectively. Complex Tree takes more time to train the model and number of observations predicted per second is also less than the other two models.

Table III Accuracy of Ensemble Classifiers

Classifier Model Type	Ensemble method	Training time (in sec)	Prediction time (obs/sec)	Accuracy in %
Bagged Tree	Bag	169.85	5300	**99.7**
Boosted Tree	AdaBoost	**54.293**	**6200**	98
RUSBoosted Tree	RUSBoosted	320.81	**6200**	98.9

Table3 shows an Ensemble classifier model types i.e Bagged, Boosted and RUSBoosted tree. It uses a Decision tree as a learning type. The maximum number of splits, Number of Learners and Learning rate is common to all the three ensemble classifiers i.e 20, 30 and 0.1 respectively. After training the model it is found that the accuracy of Bagged Tree is 99.7% as shown in Fig11 and Fig9 shows the perdition time of Boosted and RUSBoosted tree is same i.e 6200 obs/sec whereas the Fig10 shows that the training time of Boosted Tree is faster i.e 54.293sec taken to tain the model which is better than other two models under ensemble models. The accuracy of the Bagged tree is better but prediction time and training speed are slower than Boosted and RUSBoosted Tree.

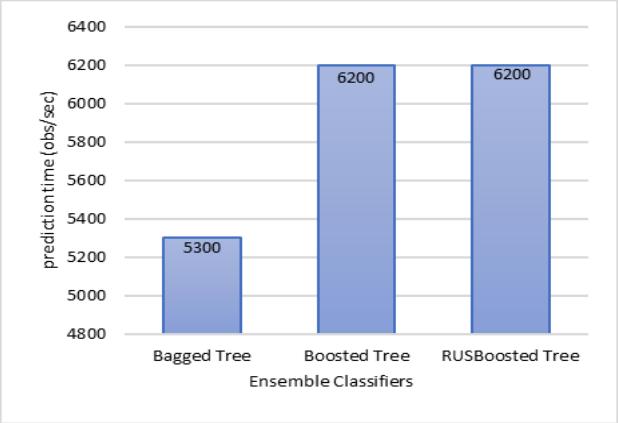


Fig 9 Prediction time of Ensemble Classifier Models

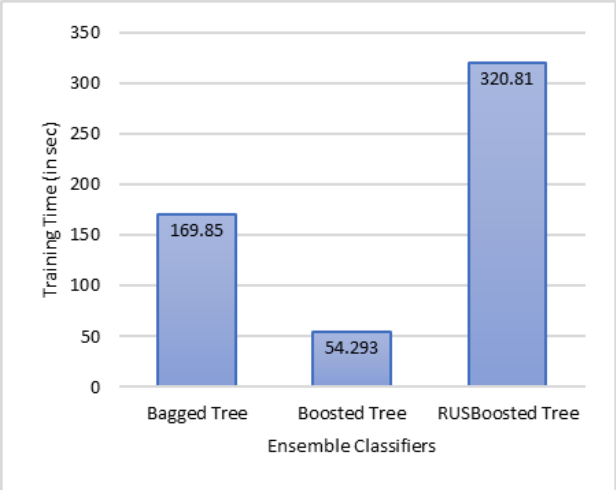


Fig 10 Training time of Ensemble Classifier Models

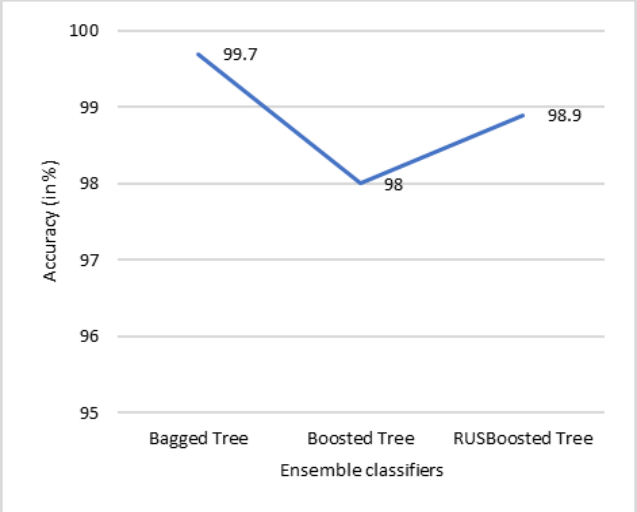


Fig 11 Accuracy of Ensemble Classifier Models

B. Performance Measures

To evaluate the performance in the area of Machine Learning the confusion matrix is used[18]. It is a table used to evaluate the performance of the classifier w.r.t dataset. It also helps to identify the rate of misclassification happened while training the model. In this work we have evaluated the performance based on four performance measures i.e - Sensitivity is also as known True Positive Rate (TPR), Precision also known as Positive Predictive Value, (PPV), False Discovery Rate and False Negative Rate.

Sensitivity – It is a measure of the proportion of actual positives that are correctly identified

TPR = TP / (TP + FN)

Precision – When it predicts actual positive, how often is it correct? Is checked using PPV

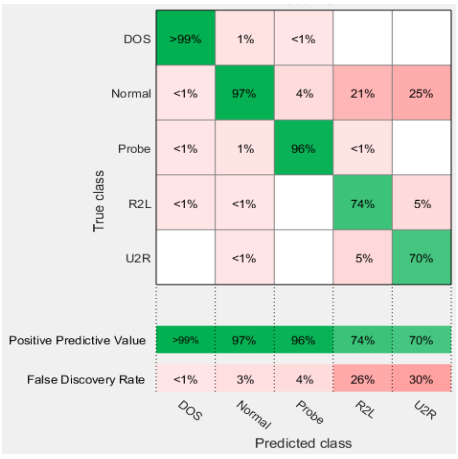
PPV = TP / (TP + FP)

False Discovery Rate – It means to reduce the number of tests to be wrong out of all tests we detect

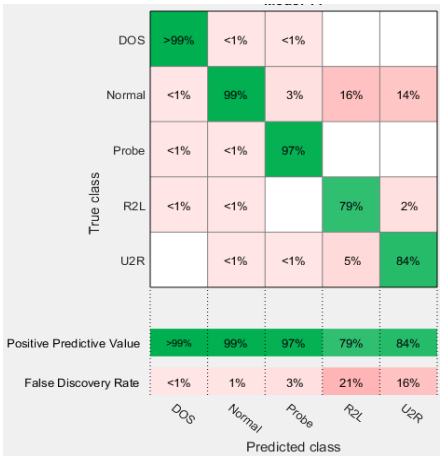
FDR = FP / (FP + TP)

False Negative Rate – It is a test result that indicates that a condition does not hold, while in fact, it does.

FNR = FN / (FN + TP)

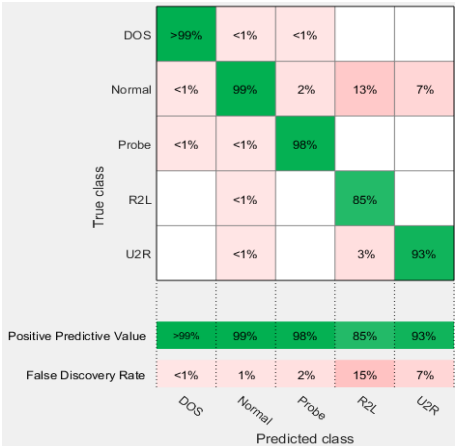


Linear SVM

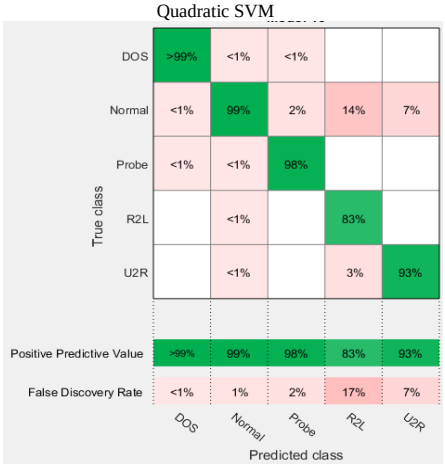


Quadratic SVM

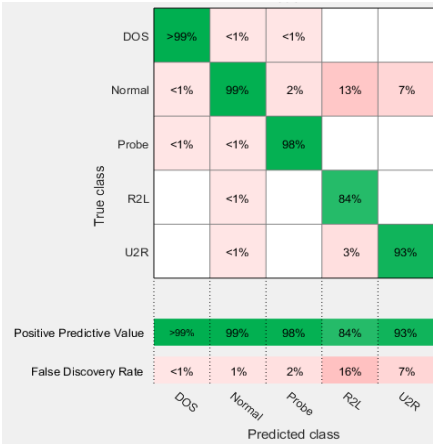
Cubic SVM



Coarse Gaussian SVM

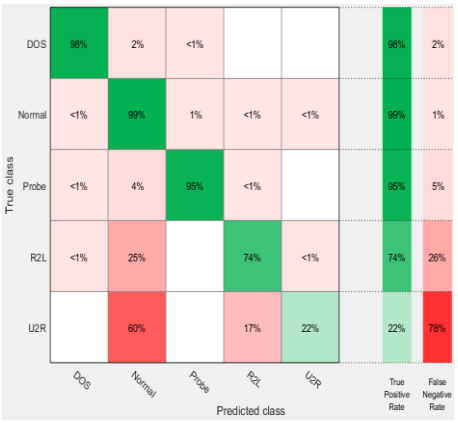


Medium Gaussian SVM

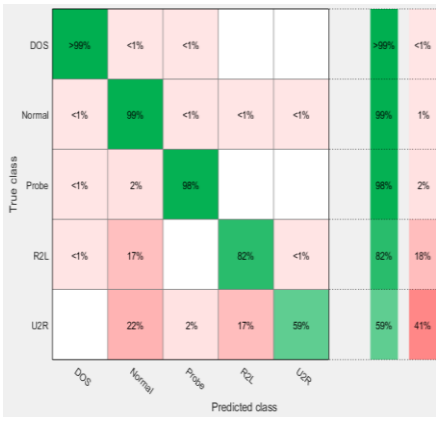


Fine Gaussian SVM

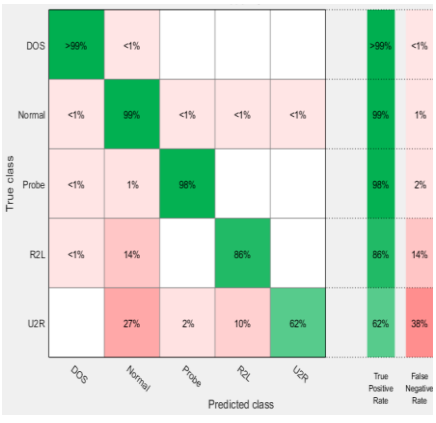
Fig 12 Positive Predictive Value vs False Discovery Rate of different SVM Classifiers



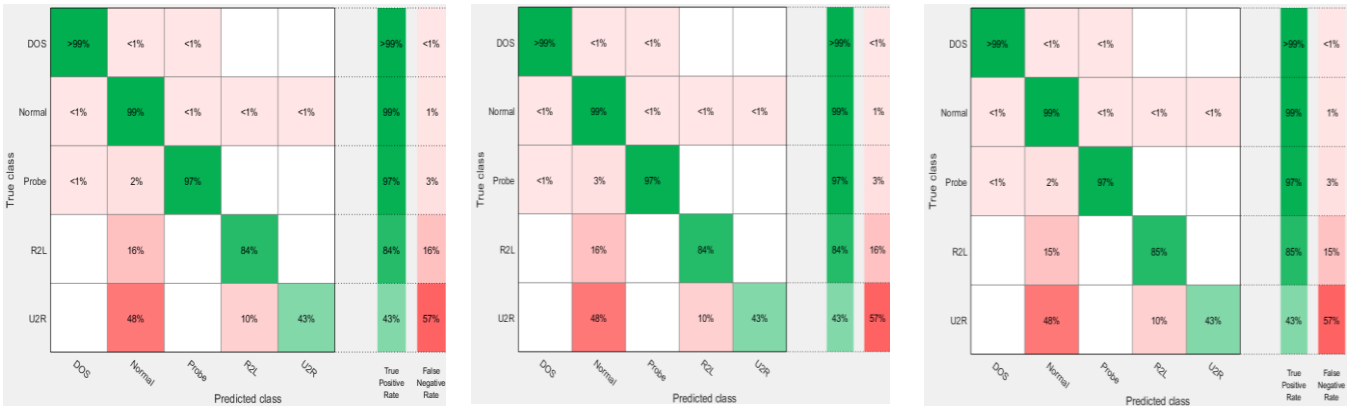
Linear SVM



Quadratic SVM

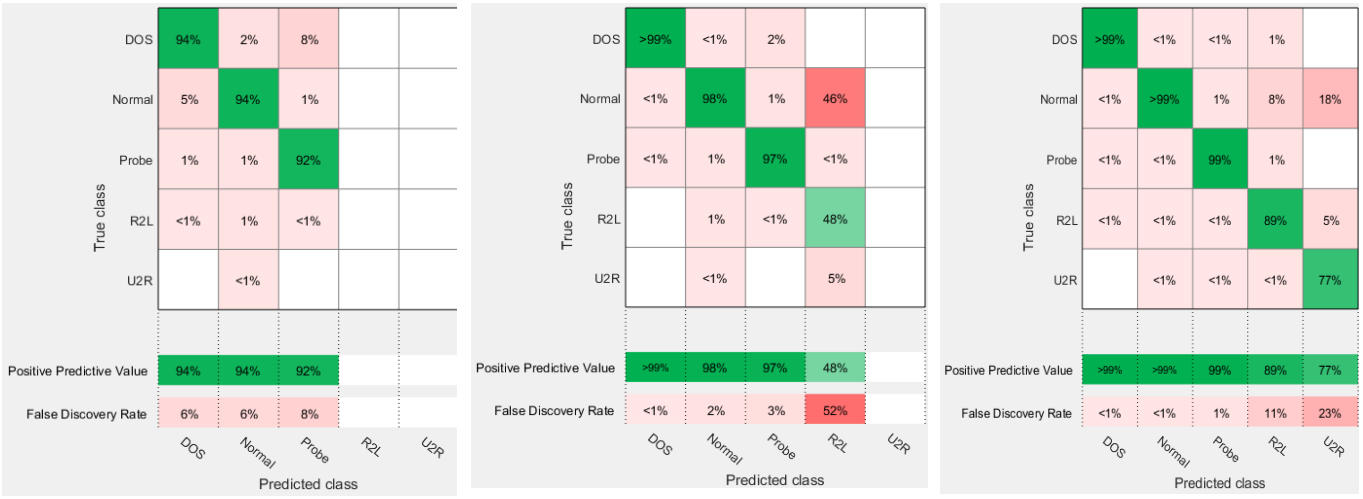


Cubic SVM



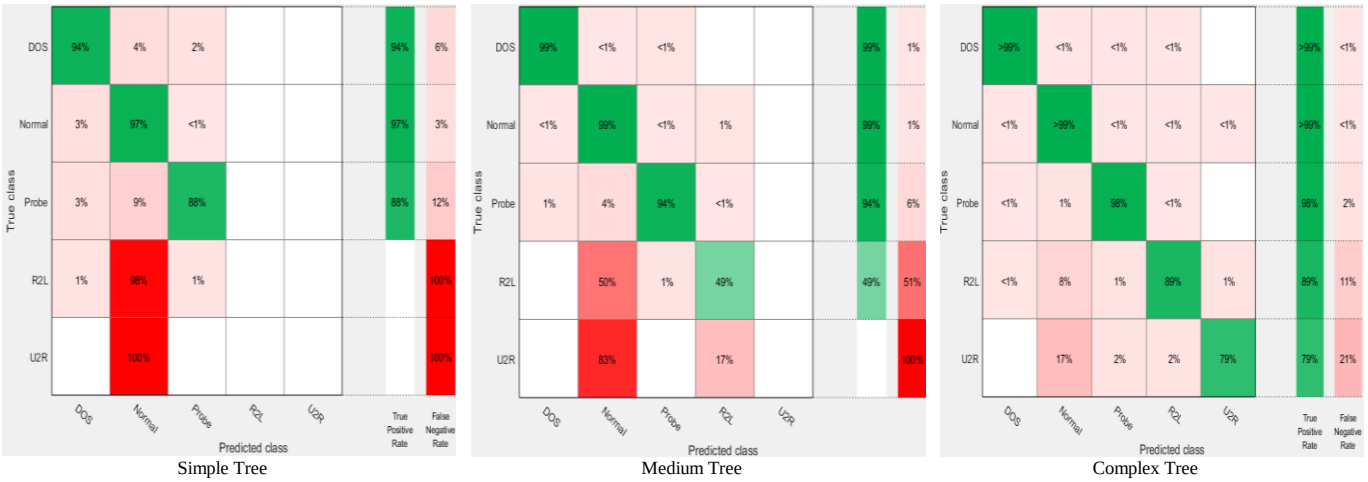
Coarse Gaussian SVM | Medium Gaussian SVM | Fine Gaussian SVM

Fig 13 True Positive Rate vs False Negative Rate of different SVM Classifiers



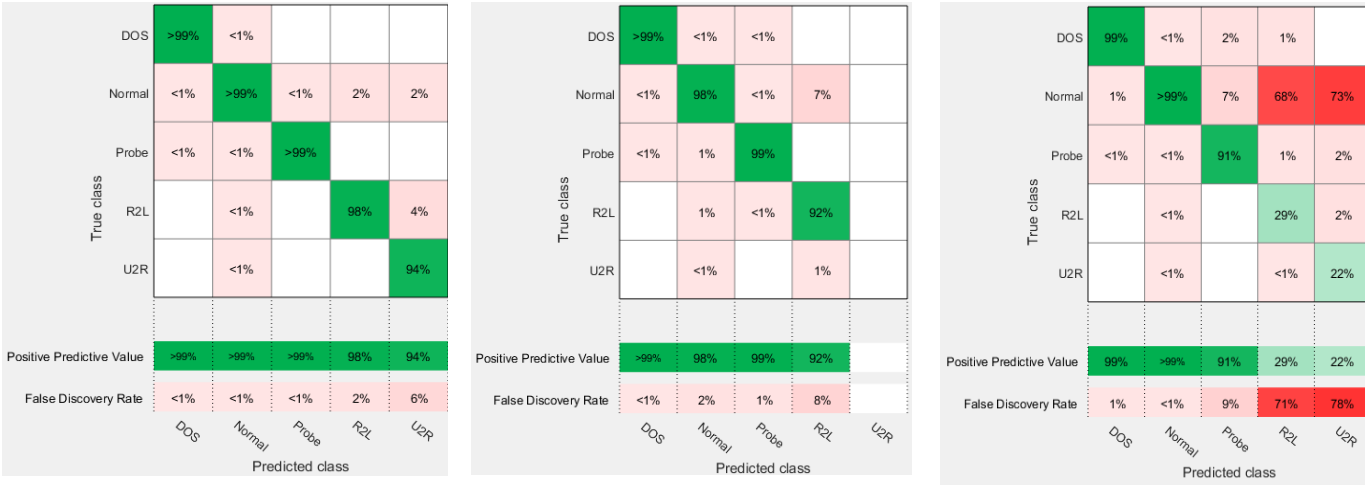
Simple Tree | Medium Tree | Complex Tree

Fig 14 Positive Predictive Value vs False Discovery Rate of different Decision tree Classifiers



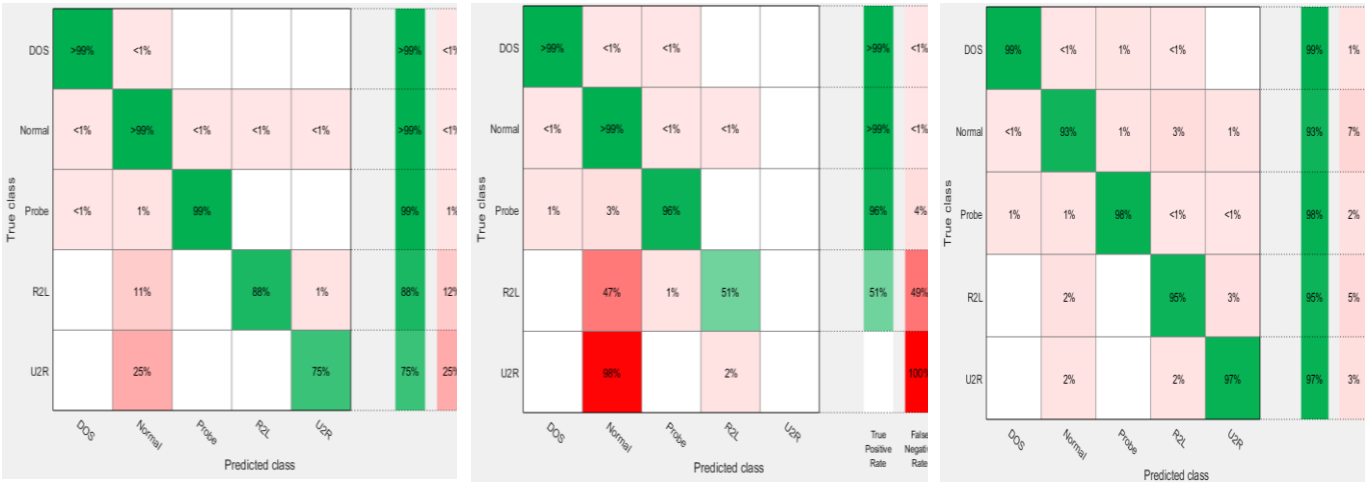
Simple Tree | Medium Tree | Complex Tree

Fig 15 True Positive Rate vs False Negative Rate of different Decision tree Classifiers



Bagged Boosted RUSBoosted

Fig 16 Positive Predictive Value vs False Discovery Rate of different Ensemble Classifiers



Bagged Boosted RUSBoosted

Fig 17 True Positive Rate vs False Negative Rate of different Ensemble Classifiers

SVM Classifiers -Fig 12 represents the attack traffic classification of four types of attack class and a normal class using the performance measure i.e Positive Predictive Value and False Discovery Rate. From this Matrix, it is observed that for the detection of DoS class all the SVM variants are accurate i.e greater than 99% PPV is recorded, only less than 1% of DoS attack class is misclassified. For normal traffic class, except Linear SVM all other SVM classifiers had shown 99% detection rate whereas the Linear SVM shows 97%, the misclassified rate (FDR) in Linear SVM is 3% where it wrongly classified the 3% of Normal traffic as attack traffic. For Probe class detection rate (PPV) using Linear SVM, it is 96%, Quadratic SVM it is 97%, rest all i.e Cubic SVM, Coarse, Medium, and Fine Gaussian is 98% each, where 4% of Probe traffic is miss-classified as Normal by Linear SVM. Quadratic SVM miss-classified 3% Probe traffic as Normal, other classifiers under SVM miss-classified 2% Probe traffic as Normal. For R2L traffic, the PPV using Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian is 74%, 79%, 83%, 85%, 83%, and 84% respectively. Miss classification rate(FDR) of R2L using Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian SVM is 26%, 21%, 17%, 15%, 17%, and 16% respectively. For correct U2R traffic class detection (PPV) using Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian SVM is 70%, 84%, 95%, 93%, 93% and 93% respectively. Miss classification (FDR) of U2R in Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian SVM is 30%, 16%, 5%, 7%, 7% and 7% respectively.

Fig 13 shows the true traffic class values actually detected as true i.e True Positive Rate (TPR) and False Negative Rates (FNR). While detecting different classes of attack traffic using SVM classifiers the following observations are recorded. Firstly, Classifying DoS class except Linear SVM all other classifiers show greater than 99% TPR and Linear SVM it is 98%. It means 2% of DoS traffic is misclassified by Linear SVM. In the case of Normal Traffic class all the classifiers are able to classify perfectly. For Probe class detection using Linear SVM, the results show 95% TPV, where 5% of traffic is misclassified. TPR of Quadratic and Cubic SVM is 98%, here 2% of traffic is misclassified. TPR of all the Gaussian SVM is 99% each. For R2L traffic class Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian SVM TPR is 74%, 82%, 86%, 84%, 84% and 85% respectively. FNR for R2L traffic class using Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian SVM is 26%, 18%, 14%, 16%, 16%, and 15% respectively.

For U2R traffic class TPR using Linear, Quadratic,

Cubic, Coarse, Medium and Fine Gaussian SVM is 22%, 59%, 62%, 43%, 43%, and 43% respectively. Whereas, the FNR for U2R traffic class detection using Linear, Quadratic, Cubic, Coarse, Medium and Fine Gaussian SVM is 78%, 41%, 38%, 57%, 57%, and 57% respectively.

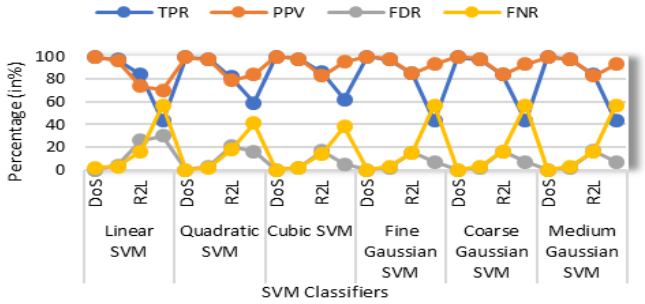


Fig 18 Performance analysis of SVM Classifiers

The Fig18 shows that the Cubic SVM Miss-classification rate (w.r.t False Discovery Rate and False Negative Rate performance measures) is less than other SVM classifiers. Also, the Positive Predictive Value and True Positive Rate of Cubic SVM is higher than other SVM classifiers with respect to all the prediction classes recorded in table4. Using table4 and Fig18 it is observed that Cubic SVM classifier attack detection rate higher and misclassification of traffic is minimized as compared to other SVM classifiers.

Table IV Various performance measures of SVM Classifiers

Classifier Model Type	Pred. Class	TPR %	PPV %	FDR %	FNR %
Linear SVM	DoS	99.46	99.58	<1	2
	Probe	97	96	4	3
	R2L	84	74	26	16
	U2R	43	70	30	57
Quadratic SVM	DoS	99.34	99.47	<1	<1
	Probe	98	97	3	2
	R2L	82	79	21	18
	U2R	59	84	16	41
Cubic SVM	DoS	**99.57**	**99.65**	**<1**	**<1**
	Probe	**98**	**98**	**2**	**2**
	R2L	**86**	**83**	**17**	**14**
	U2R	**62**	**95**	**5**	**38**
Fine Gaussian SVM	DoS	99.71	99.76	<1	<1
	Probe	97	98	2	3
	R2L	85	85	15	15
	U2R	43	93	7	57
Coarse Gaussian SVM	DoS	99.53	99.61	<1	<1
	Probe	97	98	2	3
	R2L	84	84	16	16
	U2R	43	93	7	57
Medium Gaussian SVM	DoS	99.62	99.46	<1	<1
	Probe	97	98	2	3
	R2L	84	83	17	16
	U2R	43	93	7	57

Decision Tree Classifiers – Fig14 shows the PPV vs FDR of Decision Tree Classifier variants. Here, DoS attack traffic class detection using Simple Tree classifier PPV is 94%, whereas the other two i.e Medium and Complex Tree shows greater than 99% each. FDR for DoS attack class detection using Simple Tree is 6%, which is misclassified as Normal, Probe and R2L class. For Normal Class, the Complex Tree gives better PPV i.e greater than 99% whereas Simple Tree and Medium Tree gives 94% and 98% respectively. Using Simple tree classifier misclassification rate of Normal traffic class is shown 6% and using Medium Tree classifier it is 2%. For Probe class attack detection using Simple Tree classifier shows 92% PPV, using Medium Tree classifier it is 97% and Complex Tree classifier gives 99%. Here, 8% of Probe traffic is misclassified by Simple Tree classifier, 3% misclassified by Medium Tree classifier whereas Complex Tree misclassification rate is 1%. Further, it is observed that the Simple Tree is not able to find the R2L and U2R attack traffic classes, whereas using Medium Tree classifier PPV for R2L is 48%, means FNR is 52% in R2L traffic classification which is very high. U2R traffic class Medium Tree classifier is failed to classify. Using Complex Tree classifier PPV for R2L is 89% and U2R is 77% and FDR for R2L and U2R is 11% and 23% respectively.

Table V Various performance measures of Decision Tree Classifiers

Classifier Model Type	Pred. Class	TPR %	PPV %	FDR %	FNR %
Simple	DoS	94	94	6	6
	Probe	98	92	8	12
	R2L	0	0	0	100
	U2R	0	0	0	100
Medium	DoS	99	99.41	<1	1
	Probe	94	97	3	6
	R2L	49	48	52	51
	U2R	0	0	0	100
Complex	DoS	**99.51**	**99.48**	**<1**	**<1**
	Probe	**96**	**99**	**1**	**2**
	R2L	**89**	**89**	**11**	**11**
	U2R	**79**	**77**	**23**	**21**

The TPR and FNR of Decision Tree classifier variants is shown in Fig 15. We can see here for DoS traffic class TPR of Simple, Medium and Complex Tree is 94%, 99%, and greater than 99% respectively. The misclassification rate for DoS using Simple, Medium and Complex Tree is 6%, 1%, less than 1% respectively. The Normal Traffic is fairly classified by Simple, Medium and Complex Tree i.e 97%, 99% and greater than 99% respectively. For Probe Class TPR using Simple, Medium and Complex Tree is 88%, 94% and greater than 98% respectively. The misclassification rate for DoS traffic class using Simple, Medium and Complex Tree is 12%, 6%, and less than 2% respectively. For R2L and U2R traffic class the Simple Tree classifier FDR is 100% it means the simple tree failed to identify the R2L and U2R class available in the dataset. Further using Medium Tree classifier also, the U2R traffic class is not identified. Whereas R2L TPV is 49% and FNR is 51% identified. Using Complex Tree classifier TPR and FDR for R2L, U2R traffic class is 89% and 11%, 79% and 21% respectively.

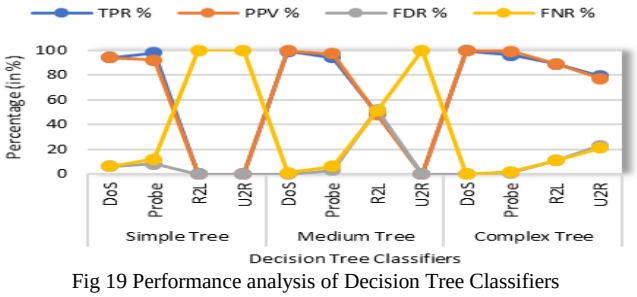


Fig 19 Performance analysis of Decision Tree Classifiers

Fig 19 shows that using Complex tree the PPV and TPR is respectable to classify all the four classes i.e DoS, Probe, R2L and U2R, which is better than other two classifiers, whereas the Simple and Medium Trees are not able to predict the R2L and U2R traffic class and the FDR and FNR are also higher than Complex Tree.

Fig19 shows Complex Tree traffic classification results are better than other two classifiers. All the Decision Tree classifier variants results are recorded in table5.

Ensemble Classifiers – Fig16 shows PPV and FDR of True classes vs Predicted classes. It is observed that the Bagged Tree Model is giving better performance than other to ensemble classifiers. The PPV using Bagged Tree classifier to identify the DoS and Probe traffic is greater than 99% each, for R2L and U2R it is 98% and 94% respectively. FDR of Bagged Tee is less than 1% for the Dos and Probe traffic class. It is observed that 2% R2L and 6% U2R traffic classes are wrongly discovered by Bagged Tree Classifier. In the case of Boosted Tree Classifier DoS and Probe traffic classes PPV is greater than 99% each. For R2L traffic 92% PPV and 8%, FDR is observed. Whereas using RUSBoosted Tree PPV and FDR of DoS, Probe, R2L and U2R traffic is 99%, 91%, 29%, 22% and 1%, 9%, 71%, 78% respectively.

TPR and FNR to classify the DoS traffic using all the ensemble tree classifiers are found fair enough (i.e greater than 99%). For Normal class traffic except RUSBoosted (i.e 93% TPR and 7% FNR) other two are fair. Bagged, Boosted and RUSBoosted classifier TPR for Probe traffic is 99%, 96% and 98% respectively; for R2L it is 88%, 51% and 95% respectively; and for U2R traffic class it shows 75%, 0%, 97% respectively. FNR for Probe traffic class using Bagged Boosted and RUSBoosted classifier is 1%, 4%, and 2% respectively. FNR for R2L traffic classification using Bagged, Boosted and RUSBoosted classifier is 12%, 49%, and 5%. FNR for U2R class using Bagged Classifier is 25%, using Boosted classifier it is 100% and using RUSBoosted classifier it shows 3%.

Table VI Various performance measures of Ensemble Classifiers

Classifier Model Type	Pred. Class	TPR %	PPV %	FDR %	FNR %
Bagged	DoS	**99.72**	**99.61**	**<1**	**<1**
	Probe	**99**	**99.52**	**<1**	**1**
	R2L	**88**	**98**	**2**	**12**
	U2R	**75**	**94**	**6**	**25**
Boosted	DoS	99.51	99.66	<1	<1
	Probe	96	99	1	4
	R2L	51	92	8	49
	U2R	0	0	0	100
RUSBoosted	DoS	99	99	1	1
	Probe	98	91	9	2
	R2L	95	29	71	5
	U2R	97	22	78	3

Bagged Tree is giving better TPR and PPV for the classification of all the prediction classes (ie. Dos, Probe, R2L, and U2R) available in the dataset shown in the Fig 20.

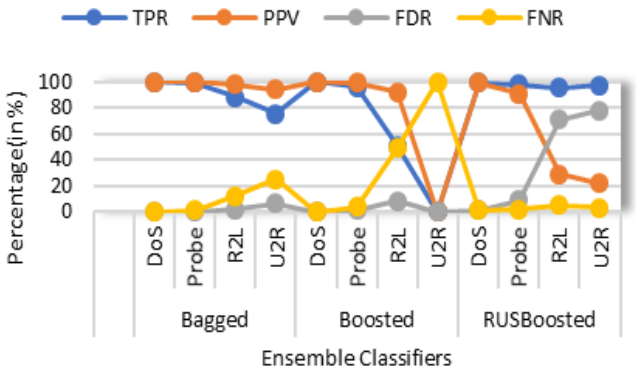


Fig 20 Performance analysis of Decision Tree Classifiers

It is also observed that the Boosted Classifier is not able to classify the traffic labeled as U2R. Thus, the FNR is 100% and the PPV and TPR are dropped to zero. In the case of RUSBoosted classifier, the graph shows that the R2L and U2R traffic classes are wrongly predicted. Bagged Tree classifier is giving better performance as compared to the other two classifiers under Ensemble classifier methods as recorded in table6.

C. Comparative Analysis

The experimental results and performance measures of all the variants of SVM, Decision Tree, and Ensemble classifier concludes that the Bagged Classifier is the best classifier among the variants of all the Machine Learning Classifiers i.e Cubic SVM, Complex Tree and Bagged Ensemble classifier as shown in table 7 and Table 8.

Table VII Best variants of SVM, Decision Tree and Ensemble.

Classifier Type	Pred. Class	TPR %	PPV %	FDR %	FNR %
Cubic SVM	DoS	99.57	99.65	<1	<1
	Probe	98	98	2	2
	R2L	86	83	17	14
	U2R	62	95	5	38
Complex	DoS	99.51	99.48	<1	<1
	Probe	96	99	1	2
	R2L	89	89	11	11
	U2R	79	77	23	21
Bagged	DoS	**99.72**	**99.61**	**<1**	**<1**
	Probe	**99**	**99.52**	**<1**	**1**
	R2L	**88**	**98**	**2**	**12**
	U2R	**75**	**94**	**6**	**25**

Fig 21 shows the TPR and PPV of all the classes using Cubic SVM, Complex Tree, and Bagged Ensemble classifier. Here, the PPV of Bagged Ensemble for classification of DoS, Probe, R2L, and U2R attack traffic is 99.61%, 99.52%, 98%, and 94% respectively, which is better than Cubic SVM and Complex Tree classifiers.

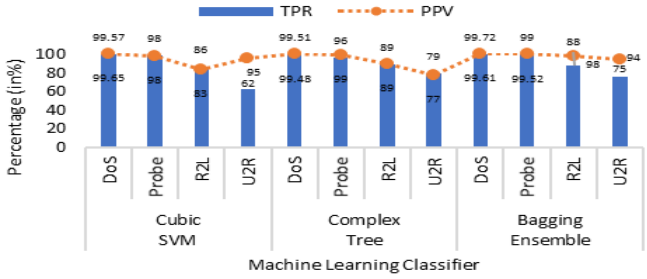


Fig 21 Evaluation of best classifiers of SVM, Complex Tree, and Ensemble in terms of True class

For the classification of all the attack traffic classes, i.e DoS, Probe, R2L and U2R in the dataset using Bagged Ensemble classifier shows the TPR of 99.72%, 99%, 88%, and 75% respectively. DoS and Probe classification using Bagged Ensemble is giving better performance. R2L the complex tree gives 89% TPR which is 1% better than Bagged Ensemble classifier and for R2L Complex tree classifier is giving 79%. Here, it is 4% higher than Bagged Ensemble Classifier.

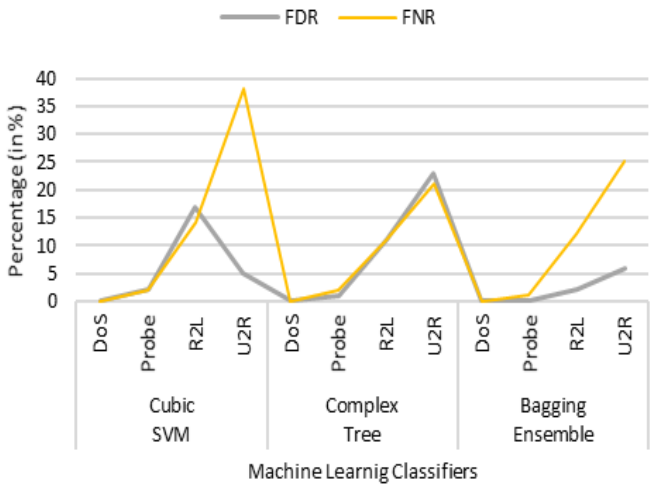


Fig 22 Evaluation of best classifiers of SVM, Complex Tree, and Ensemble in terms of False class

The wrong prediction rate of Cubic SVM, Complex Tree and Ensemble Classifiers is shown in Fig 22. It is observed that the FDR to classify the R2L is more i.e 17% using the Cubic SVM,11% using Complex Tree and 2% using Bagged Ensemble classifier. FDR of U2R is more using Complex tree classifier i.e 23% than Cubic and Bagged classifier. FNR of Cubic SVM to classify the U2R class is very high i.e 38%, in Complex Tree, it is 21% and using Bagged Ensemble it gives 25%. After evaluating overall performance measurement and results of three best variants (i.e Cubic SVM, Complex Tree, and Bagged Ensemble) it is found that the performance of Bagged Ensemble classifier is better to classify DoS, Probe, R2L and U2R classes as the FDR and FNR among all the class is minimal or one place it is close to other classifiers.

Table VIII Accuracy of Cubic SVM, Complex Tree, and Ensemble

Classifier Model Type	Accuracy in %
Cubic SVM	99.3
Complex Tree	99.4
Bagged Tree	**99.7**

Fig 23 Accuracy rate of best classifiers of SVM, Complex Tree and Ensemble

Fig 23 Shows the accuracy rate of Cubic SVM, Complex Tree, and Bagged Ensemble classifier, where the accuracy rate of Bagged Ensemble is 99.7% which is higher than Cubic SVM and Complex Tree Classifier.

II. CONCLUSION AND FUTURE DIRECTION

In this paper, different types of Machine Learning classifiers are used to evaluate Accuracy and Performance measures i.e PPV, TPR, FDR and FNR using the KDD Cup Dataset. We have taken approximately 25244 instances having all the possible traffic class instances for the evaluation of the classifiers. Based on the results and performance matrix the Cubic SVM is found best variant among SVM classifiers i.e Linear SVM, Quadratic SVM, Cubic SVM, Coarse Gaussian SVM, Medium Gaussian SVM and Fine Gaussian SVM having an accuracy of 97.8%, 99.1%, 99.3%, 99.1%,99.1% and 99.1% respectively. Among all the decision trees used, Complex Tree classifier outperforms all other trees by having an accuracy of 99.4%. Simple tree and Medium tree are having an accuracy of 94% and 98% respectively. Further, the Bagged Ensemble classifier is found better among other Ensemble Classifiers. Bagged Ensemble, Boosted Ensemble, and RUSBoosted Ensemble show the accuracy of 99.7%, 98%, and 98.9% respectively. Based on the attack traffic class prediction rate and accuracy, the Bagged Tree is giving better results w.r.t the rate of accuracy i.e 99.7%. It is capable of correctly classifying the four attack classes i.e DoS, Probe, R2L and U2R as compared to other evaluated Machine Learning Classifiers.

In Future, these Machine Learning classifiers can be used for the classification and validation of other repositories of Intrusion Detection System (IDS) dataset.

REFERENCES

1. "Intrusion Detection System – A," vol. 4, no. 1, pp. 31–44, 2015.
2. A. Sivaprasad and S. Hodge, "Machine Learning based Traffic Classification using Statistical Analysis," no. March, pp. 187–191, 2018.
3. I. Alsmadi and D. Xu, "Security of Software Defined Networks: A survey," *Comput. Secur.*, vol. 53, pp. 79–106, 2015.
4. H. Sedjelmaci, S. Mohamed Senouci, and T. Taleb, "An Accurate Security Game for Low-Resource IoT Devices."
5. T. Nadu, "A survey on network traffic classification techniques," vol. 117, no. 22, pp. 107–111, 2017.
6. Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of Denial-of-Service Attacks Based on Computer Vision Techniques," *IEEE Trans. Comput.*, 2015.
7. S. A., "Defense Mechanism for Ddos Attack Through Machine Learning," *Int. J. Res. Eng. Technol.*, vol. 03, no. 10, pp. 291–294, 2015.
8. C. Zhang *et al.*, "Neural Network Based Protection of Software Defined Network Controller against Distributed Denial of Service Attacks Neural Network Based Protection of Software Defined Network Controller against Distributed Denial of Service Attacks," no. November 2017, pp. 2557–2575, 2017.
9. R. T. Kokila, S. Thamarai Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," *6th Int. Conf. Adv. Comput. ICoAC 2014*, pp. 205–210, 2015.
10. S. Hao, J. Hu, S. Liu, and T. Song, "Improved SVM Method for Internet Traffic Classification Based on Feature Weight Learning," pp. 102–106, 2015.
11. N. Williams, S. Zander, and G. Armitage, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification," vol. 36, no. 5, pp. 7–15, 2006.
12. J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, and Y. Liu, "A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges," *IEEE Communications Surveys and Tutorials*, 2018.
13. A. Jamuna and V. E. S. E, "Efficient Flow based Network Traffic Classification using Machine Learning," vol. 3, no. 2, pp. 1324–1328, 2013.
14. O. Y. Al-Jarrah, O. Alhussein, P. D. Yoo, S. Muhaidat, K. Taha, and K. Kim, *Data Randomization and Cluster-Based Partitioning for Botnet Intrusion Detection*, vol. 46, no. 8. 2016, pp. 1796–1806.
15. S. D. Hettich, S. and Bay, "KDD-CUP-99 Task Description." pp. 1–4, 1999.

16. P. Aggarwal and S. K. Sharma, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection," *Procedia Comput. Sci.*, vol. 57, pp. 842–851, 2015.
17. UCI Machine Learning Repository, "KDD Cup 1999 Data." p. 92697, 2015.
18. M. R. Joshi, "A Review of Network Traffic Analysis and Prediction Techniques."

AUTHORS PROFILE

Sukhvinder Singh received the M.E degree in Computer Science and Engineering from Thapar University, Patiala, India in 2011. He is pursuing the doctoral degree in the Department of Computer Science, Pondicherry Central University, India. His research interest includes Network Security, Software Defined Network and Cyber Security. Currently, he is a Lecturer in the Department of Computer Science and Engineering, Dr.B.R Ambedkar Institute of Technology, Port Blair, India.

S.K.V.Jayakumar received the PhD degree in Computer Science and Engineering, specialized in Services Computing from Pondicherry Central University, India. He is a graduate of Electrical Engineering and Masters in Computer Science and Engineering. He is an Assistant Professor of Computer Science in School of Engineering and Technology at Pondicherry Central University. He has been in Teaching and Research for 22 years. His main research area is Data Science and Services Computing and has number of research publications as output to his research in the said specializations. He has good experience of handling funded projects in his specializations and organizing National and International Conferences.
.