Analysis of Intrusion Detection using Machine Learning for Computer Network System

Sonam Jayaswal[1], Gholam Mursalin Ansari[2], Upendra Kumar[3]
[1]*Scholar, Computer Science & I T, YBN University, Ranchi, India*
[2]*Assoc. Professor, Computer Science & IT, YBN University, Ranchi, India*
[3]*Assist. Professor, Computer Science & Engineering, BIT Patna, India*

Abstract- In today's generation internet has become a very essential component for almost everything. According to an estimate it is suggested that at least 2.5 quintillion bytes of data is generated by a human being on a daily basis. This large dumps of data increases the risk of network attacks to an alarming rate jeopardizing the integrity and confidentiality of the users. These penetration in the network is increasing day by day and becoming more sophisticated and complexed. This is where the IDS (Intrusion detection system) comes into picture, providing a protecting layer over the infrastructure with its continuous adaptation. Through this paper we tested various machine learning classifiers on KDD and UNSW-NB15 dataset to strengthen the detection ratio in IDS. The main focus was on the false negative and false positive performance matrix for more accurate detection in the intrusion detection system. Lastly, we tried to test the classifiers with the important features of the dataset along with the use of ensemble methods to enhance the performance of the IDS.

Keywords: Intrusion detection, KDD dataset, UNSW-NB15, SVM, Random forest, F1 Score, Accuracy.

1. INTRODUCTION

IDS innovation has progress essentially since it initially began in the '80s along with various endeavors. On account of the ever-progressing cyber dangers, crime, and bullying technology's evolution will never rest. However, most of these developments are made on the basis of previous problems making us more reactive, in today's world of technology we need more human involvement in this area in understanding the pattern of attack as predicting the attacks before its occurrence. Our technology needs to be faster than these threats [1]. The goal of IDS is pretty simple to detect an intrusion but quite difficult on its own. The fact of the matter is IDS systems only find the evidence of an intrusion while they are in progress or after a while of the occurrence hence they needed to be continuously monitored. The attacker's manifestations that are the data collected as evidence of the attack needed to be insightful and trustworthy, then only the system can detect an intrusion [2]. In this world where the reach of the Internet is increasing every day, there is a threat of increasing Intrusion and a variety of attacks too. Specifically, these dangers keep on continuing because of the prerequisites, restricted accuracy, and absence of adaptability in our current IDS systems[1,4,5]. Nowadays we have systems that can detect the change in the normal pattern as well as characteristics of the intrusion as well but they both have their restrictions on their own [3]. With all these things pushing, our old and traditional IDS system cannot keep up with the complexity of attacks and their types. we need a much more effective way to deal with this problem and machine learning provides an outlet as the current intrusion detection systems require more intelligent mechanisms and better insights so that their efficiency can increase incredibly[4,11]. With more and more mysterious and sophisticated attacks occurring in Intrusion detection many theoretical and practical approaches had been made to detect them using various Machine learning models and other techniques. Various algorithms and data mining techniques are applied to make the IDS more flexible and prone to withstand the attack both in the field of AIDS (anomaly-based) and SIDS (signature-based). [12,13]Nonetheless, most of them produce either a very low accuracy rate or a very high false alarm ratio. Moreover, these tests are done on the KDD99 that have a high range of attacks but lack newer and advanced attacks making it difficult for today's world.

The most significant and drawn-out interaction of beginning with machine learning models is getting dependable and reliable information. We use KDD Cup 1999 Data and the unsw-nb15 datasets to assemble prescient models equipped for recognizing

interruptions and attacks, and important associated connections.

1.1 The working on KDD DATASET:

The KDD dataset which contains standard set of information comprising of 4898431 instances with 41 attributes. Each connection is labelled as either normal or as an attack, with exactly one specific attack type. Each connection record consists of about 100 bytes, with a total of 21 attack types.

Table1: Different attacks and Instances

Categories Of Attack	Attack name	Number Of Instances
DOS	SMURF	2807886
	NEPTUNE	1072017
	Back	2203
	POD	264
	Teardrop	979
U2R	Buffer Overflow	30
	Load Module	9
	PERL	3
	Rootkit	10
R2L	FTP Write	8
	Guess Password	53
	IMAP	12
	MultiHop	7
	PHF	4
	SPY	2
	Warez client	1020
	Warez Master	20
PROBE	IPSWEEP	12481
	NMAP	2316
	PORTSWEEP	10413
	SATAN	15892
normal		972781

The KDD dataset is exceptionally useful for assessing and testing different machine learning algorithms. The classifier determination model proposed by the authors comprised of extracted 49596 instances of KDD dataset to be implemented on several machine learning algorithms.

In following sections, the accompanying machine learning classifiers (J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network) were implemented, executed, tried and assessed dependent on KDD dataset. The interest is in the most important performance parameters e.g. false negative and false positive to evaluate the selected classifiers[8-10]. As a result of the implemented experiments the focus will be on selecting the effectiveness of the machine learning classifier which achieved the accepted accuracy rate with the minimum false negative value.

The upcoming table describes the conveyance of attack types inside KDD dataset. The 21 types of attack was distributed upon four groups with difference in occurrences and instance. The whole dataset appeared to be a bit unbalanced but provided largest number of packet attributes, compromising of 79% Dos attack with normal packet containing 19% of the dataset and the rest was other attack types.

1.2 DATA PRESPROCESSING AND CLEANING:

Before the implementation of the machine learning algorithm data present in the dataset need to be processed meaning the important features must be selected for evaluation[8-11].n. Using all the features of a dataset has a higher chance to increase the computational cost as well as the error rate of the system, hence using all the data of the dataset may not provide the best required results. This is on the

grounds that a few highlights are repetitive or are not helpful for making a qualification between different classes

The different classifiers used in the dataset includes Gaussian Naive Bayes, Decision Tree, Random Forest, Support Vector Machine, And Logistic Regression apart from these implemented classifiers other algorithms such as K-means, Bayes network, Multi-Layer Perceptron (MLP), j48 are also discussed. The outcome of continuous development of the technology increases the demand of the of machine learning algorithm to analyse and extract information from the processed dataset. In view of 148753 cases of records we were able to create the preparation models for all the chose machine learning classifiers. Every one of the examined models are arranged and looked at for a thorough study of classifiers efficiency[14-16].

Data Correlation is an approach to comprehend the connection between different factors and characteristics in your dataset. The following heat map shows the highly correlated values in the dataset and ignoring them from the analysis. Feature mapping for different features were applied and irrelevant features were removed before the modelling process.

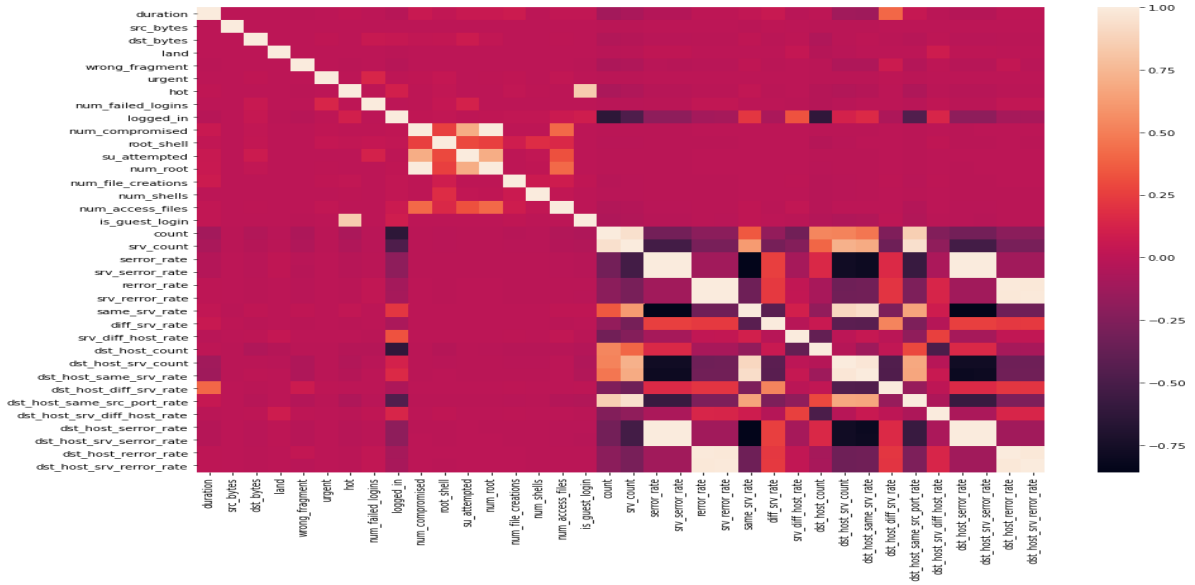
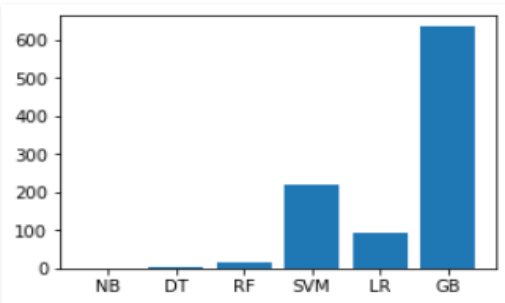


Fig.1 cleaning of the data set

2. MACHINE LEARNING CLASSIFIERS PERFORMANCE THEIR RESULTS AND DISSCUSSION

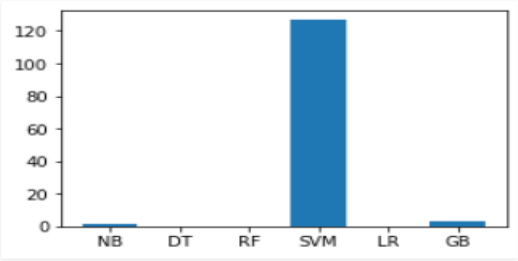
The different machine learning models were created and testing phase was implemented on the models. The classifiers provided different testing and training time from the instances which is shown in the plots. The testing time score for different models.

Output:



The testing time score for different models.

Output:



The training and testing accuracy for the different models

Output:

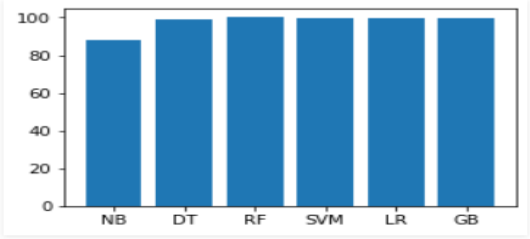


Fig.3: Accuracy of the models

There are many evaluation metrics for the classification algorithm, hence to carry out a reasonable testing stage randomized data was extracted including all the attack types within 21 types of attack present in KDD dataset. Confusion matrix was generated for the classifiers incorporating significant data about existing and anticipated output classes. The performance matrix was computed on the basis of the True positive (a result where the model accurately predicts the positive class) showing the attack packet as an attack, True negative (a result where the model accurately predicts the negative class) representing normal packet as normal, False positive (showing an existence of a condition which do not exist) hence showing an incorrect classification when an attack packet is considered as normal , False negative (do not show an existence of a condition which exist) hence showing an incorrect classification when an normal packet is considered as attack. All the given criteria has their significant role in the computation time and hence the effect the precision which is one of a primary performance indicator representing the ratio of the correctly identified attack by total number of attacks.

$$\text{Precision}=TP/(TP+FP)$$

In general both the TP and precision are both important criteria for the intrusion detection process, but the FP and FN rates should also be kept in mind for the performance parameters. The model tends to reduce both the parameters as much as possible.

Table 2:The precision score and the true positive rate for different model were recorded:

Machine Learning Classifiers	TP Rate	Precision
J48	0.931	0.989
Random forest	0.938	0.991
Random tree	0.906	0.992
Decision table	0.924	0.944
MLP	0.919	0.978
Naive Bayes	0.912	0.988
Bayes Network	0.907	0.992

The result shows the fact that the Decision tree classifier has the lowest precision score indicating that a large number of normal packets were classified as attack packets showing an increase in FP rates. Apart from that the random tree classifier lowest TP rate and the random forest classifier come with the highest TP of 93%. The precision score of both the random forest and random tree classifier are quite similar and debatable.

The false positive and the false negative rates of the different classifiers are also plotted and can be concluded that the random tree classifier has the highest FN rate of 0.093 showing a large number of attack classified as normal also decision tree classifier shows the lowest FN rate combining with a highest FP rate concluding to a fact that a large number of normal packets were classified as attack packets.

Fig.4: FP and FN rates

Finally the average accuracy score was calculated for different classifiers for the KDD dataset. The formula used for the evaluation is given by
AVERAGE ACCURACY RATE = (TP + TN) / (TP + FP + TN + FN).

Table3: The table shows the accuracy of the different machine learning classifiers:

Machine Learning Classifiers	Accuracy Rate
J48	93.10%
Random forest	93.77%
Random tree	90.57%
Decision table	92.44%
MLP	91.90%
Naive Bayes	91.23%
Bayes Network	90.73%

It is evident from the table that the Random tree classifier has the lowest accuracy score and the random forest model comes up with the highest of 93.7% followed by the decision tree classifier. The MLP and Naïve Bayes classifier presented with the similar accuracy score.

THE UNSW-NB15 dataset:
Preceding to the development of the UNSW-NB15, there were a few dataset already present in the field of ID. But these dataset were not able to generate a real world environment of the network traffic and also there was a lot of redundancy in the dataset as well as a lot of data was also missing[17,18].

These factors created a lot of challenges for cyber security group in Australian Centre for Cyber Security (ACCS) and around the globe which lead to the configuration and generation of UNSW-NB15 dataset. Hence with the help of the IXIA Perfect Storm tool the new dataset was created which extricate a hybrid of modern as well as normal attacks.
The generated dataset consisted of 2, 540,044 records of data divided into four CSV files and further a section of data was also divided into training and testing data. The dataset was further elaborated to have nine types of attack which includes:-

Table 4: Attacks descriptions

Type	No. Records	Description
Normal	2,218,761	Natural transaction data.
Fuzzers	24,246	Attempting to cause a program or n-etwork suspended by feeding it the randomly generated data.
Analysis	2,677	It contains different attacks of port scan, spam and html files penetrations.
Backdoors	2,329	A technique in which a system security mechanism is bypassed stealthily to access a computer or its data.
DoS	16,353	A malicious attempt to make a server or a network resource unavailable to users, usually by temporarily interrupting or suspending the services of a host connected to the Internet.
Exploits	44,525	The attacker knows of a security problem within an operating system or a piece of software and leverages that knowledge by exploiting the vulnerability.
Generic	215,481	A technique works against all block-ciphers (with a given block and key size), without consideration about the structure of the block-cipher.
Reconnaissa-nce	13,987	Contains all Strikes that can simulate attacks that gather information.
Shellcode	1,511	A small piece of code used as the payload in the exploitation of software vulnerability.
Worms	174	Attacker replicates itself in order to spread to other computers. Often, it uses a computer network to spread itself, relying on security failures on the target computer to access it.

3.DATA PREPROCESSING

Before the application of different machine learning classifiers the records in the dataset should be processed in order to remove highly correlated values from the dataset. The UNSW-NB15 dataset was divided into two sets namely training and testing data for the analytic purposes. The goal is to determine the complexity of different machine learning classifiers on the training and testing set[20].

The first step was to read and clean the dataset. As the dataset contains 2, 540,044 records of data divided into four CSV files, a single dataframe was created for the application of the data in the record. Then the dataset was divided into training and testing set so that the cleaning pre-processing and other kind of operation can be performed on the dataset. The goal is to create a pipeline so that the raw data can be directly cleaned and send into the classifiers.

The first step in the data pre-processing was to do an exploratory data analysis of the dataset to evaluate the correlation of features and generate the highly correlated values. The following heat map shows the highly correlated values in the dataset.

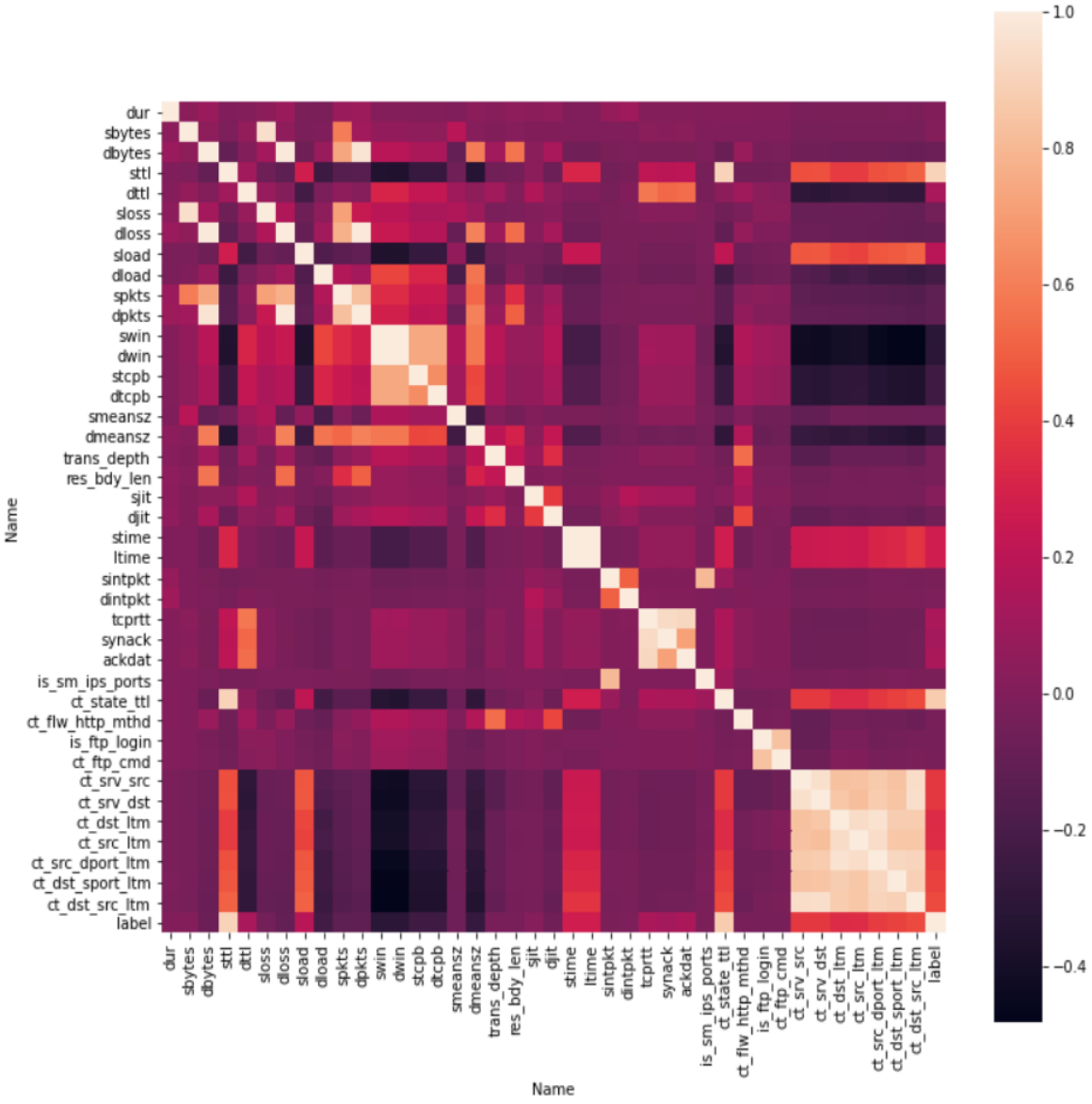


Fig.5 Cleaning of the Dataset UNSW-NB15

The next step was to remove the highly correlated values from the dataset. The features having more than the desired correlation value were dropped as machine learning classifiers work more efficiently with low correlation value with each other and high correlation value with the target labels.

3.1 MACHINE LEARNING MODELS APPLICATION AND THEIR RESULTS:

After the pre-processing of data, various machine learning classifiers were used for the training and testing of data. The generated data from the dataset is highly imbalanced hence accuracy score cannot became a valid performance matrix. The validation of the performance of various model with the help of AUC and f1 score along with the FAR (false alarm rate) can be justified.

The AUC (area under curve) score is used determines the ability of classifier to judge between correct and false class labels and its value lies between [0, 1]. A higher AUC score determines better performance and a vice versa.

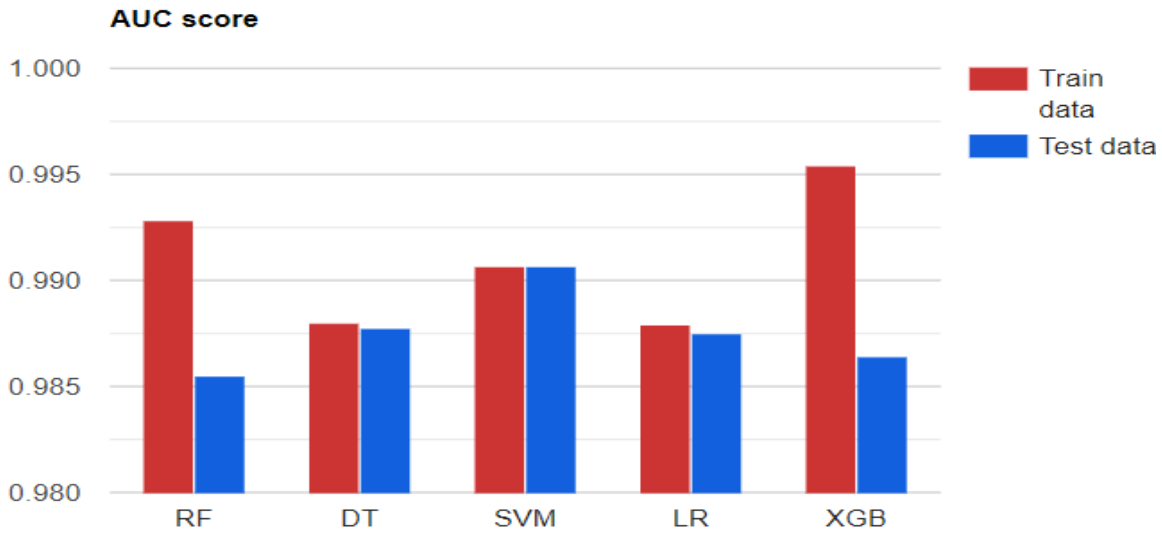
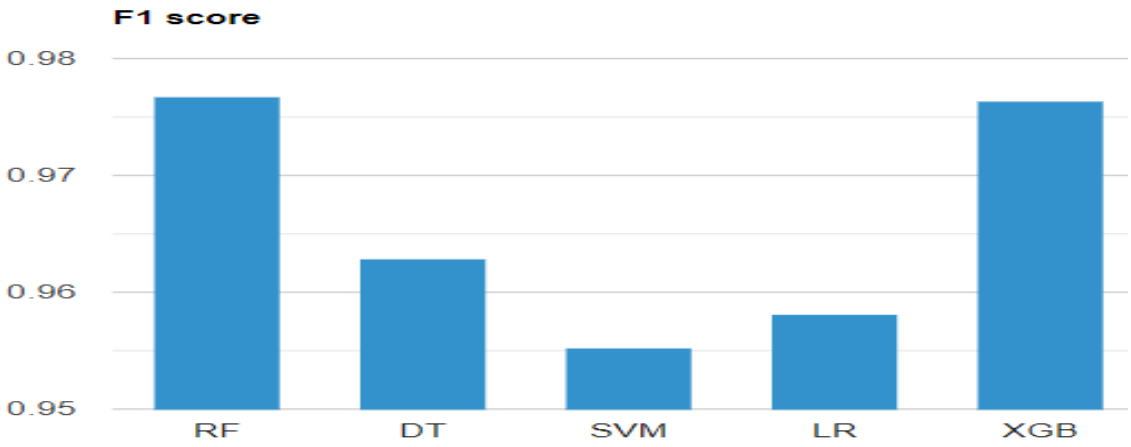


Fig 6: AUC score

The above data shows that the there is no visible overfitting between the data for the models but as compared to the other machine learning classifiers in Random forest and XGB (Gradient boosted decision tree) there is a gap between the train and test score but not significant enough for overfitting.

Similarly, F1 score is used when a middle ground or balance is needed between the recall and precision.

F1 = 2*((precision * recall)/ (precision + recall))



The above graphs shows the F1 score of the various machine learning classifiers, the data shows that Random forest performed better score than the rest of the models. The decision tree classifier produced better results than LR and SVM classifiers.

The following table shows the AUC score, the F1 score and the False alarm rate for the different machine learning classifier.

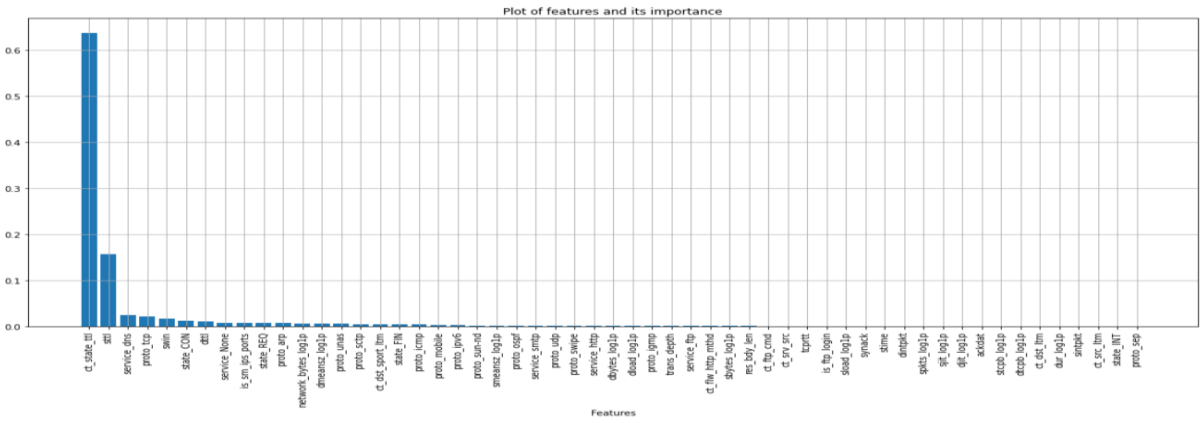
Table5. Details descriptions of Classifiers:

Classifiers	AUC SCORE	F1 SCORE	False Alarm Rate
Random forest	0.98547	0.97675	0.01452
Decision Tree	0.98774	0.96258	0.01225
SVM	0.99068	0.95520	0.00933
Logistic Regression	0.98725	0.95810	0.01247
GBDT	0.98640	0.97636	0.013591

Further observation showed that in random forest classifier the FP is reduced but there is increase in FN rates. The FAR score of SVM is very impressive, FP in comparison to the Logistic regression classifier has also increased, but the FN rate is very low. The decision tree classifier also show promising results but there is no reduction in the FP rate.

3.2 WORKING WITH IMPORTANT PARAMETRES:

For the purpose of achieving a higher F1 score and low False alarm rate, the various machine learning classifiers were tested with only the important features of the dataset. From 197 features of the dataset only the important features are sorted and the classifiers are evaluated to check how it effects their performance.

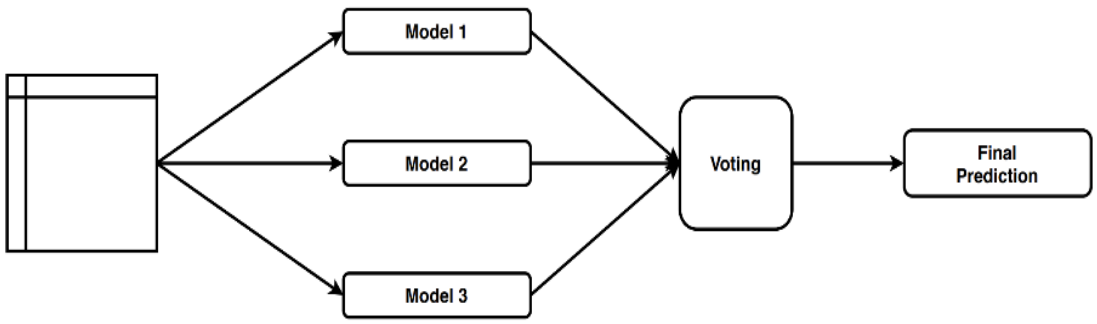


This graph shows the important features of the dataset. The result of the performance of the classifiers are shown in the table. Although for Decision tree classifier, it did not affect the values significantly but reduced its value by a bit. The performance of Random forest classifier was boosted by a little amount and this is giving the highest score among all the trained model.

Classifiers (with imp parameters)	AUC SCORE	F1 SCORE	False Alarm Rate
Random forest	0.98622	0.97805	0.01377
Decision Tree	0.98732	0.96241	0.01267

4. USING ENSEMBLE METHODS

Voting classifiers is one of the ensemble training model which trains on the numerous machine learning classifiers by combining their predictions. Its working is similar to that of the Random forest classifier but with the exception in the base estimator, unlike the other techniques in voting ensemble they can be heterogeneous in nature. Hence it gives us the freedom to use various machine learning classifiers to be used as a base and then combine their results using voting ensemble.



The voting ensemble is done using two types of voting one in which final model is selected on the basis of majority vote and another in which average voting is done by calculating the average probability of all the base models.

The voting ensemble technique is a convenient technique and sometimes give accuracy and overall results better than the best base classifier as it assembles the accuracy of other classifiers and predicts the results hence covering the weak points of the individual models. The main key for better results is the use of a diversity in base classifiers for voting.

*"*Voting Classifier supports two types of voting:

Hard Voting: - Predict the class with the largest sum of votes from models. Suppose three classifiers predicted the output class (A, A, B), so here the majority predicted A as output. Hence A will be the final prediction.

Soft Voting: - Predict the class with the largest summed probability from models. Suppose given some input to three models, the prediction probability for class A = (0.30, 0.47, 0.53) and B = (0.20, 0.32, 0.40). So, the average for class A is 0.4333 and B is 0.3067, the winner is clearly class A because it had the highest probability averaged by each classifier.

A voting ensemble may be considered as a meta-model, a model of models. As a meta-model, it could be used with any collection of existing trained machine learning models and the existing models do not need to be aware that they are being used in the ensemble. This means you could explore using a voting ensemble on any set or subset of fit models for your predictive modelling task."

The best results for the dataset were produced by the Random forest, Decision tree and Gradient Boosted Decision tree classifiers, hence the dataset was evaluated on the voting classifier model of the combined three models which produced an AUC score of 0.98698 along with that the F1 score achieved was 0.97793 and the FAR score of 0.01301.

Table6: comparative analysis of different classifiers:

Classifiers	AUC SCORE	F1 SCORE	False Alarm Rate
Random forest	0.98547	0.97675	0.01452
Decision Tree	0.98774	0.96258	0.01225
SVM	0.99068	0.95520	0.00933
Logistic Regression	0.98725	0.95810	0.01247
GBDT	0.98640	0.97636	0.01359
Classifiers (with imp parameters)	AUC SCORE	F1 SCORE	False Alarm Rate
Random forest	0.98622	0.97805	0.01377
Decision Tree	0.98732	0.96241	0.01267
Voting	0.98698	0.97793	0.01301

5. CONCLUSION

In the above report the various machine learning models J48, Random Forest, Random Tree, Decision Table, MLP, Naive Bayes, and Bayes Network were executed to judge their efficiency and precision in the intrusion detection of KDD dataset. Around 148000 instances were tested as training model and 60000 random testing data was implemented for different machine learning model. The process has shown that all the machine learning classifiers are able to create a model for the detection of attacks. Although they perform differently in different environment. The Random forest classifier comes up with the highest accuracy score and was very promising and acceptable for the performance parameter except for the false negative rates. The decision tree classifier although did not come up with the highest accuracy rate but it achieved the lowest FN value. Finally to save the accessibility and the confidentiality of the network all the performance parameters including the FN and FP rates, combing with the testing and training time for all the classifiers must be taken in account.

Different machine learning classifiers were trained and evaluated and their score is validated. The table shows the validation of all the tried models long with their score. Along with all the classifier Random forest classifier and the Decision tree classifier were again tested with only important parameters. To even extend the working the dataset was trained on the voting ensemble classifier which produced the highest AUC score among all the classifier tested but there was a significant difference in between the test and train F1 and FAR score. The training FAR

was reduced to a very low but in test FAR some of the FN and FP present which were equal in number.

If the classifiers were judged on the basis of their F1 score then the Random forest classifier with the important parameters resulted at the peak followed by the voting classifier. The FAR score of Random forest was also only 1.3% hence it is best suited for the classification task.

REFERENCE

[1] Mohamed, Ashara Banu, Norbik Bashah Idris, and Bharanidharan Shanmugum. " A brief introduction to intrusion detection system." International Conference on Intelligent Robotics, Automation, and Manufacturing. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[2] Kemmerer, R. A., & Vigna, G. (2002). Intrusion detection: a brief history and overview. Computer, 35(4), supl27–supl30.

[3] Intrusion Detection Systems. (2008). Advances in Information Security. doi:10.1007/978-0-387-77265-3

[4] Hui, L., & Yonghui, C. (2010). Research Intrusion Detection Techniques from the Perspective of Machine Learning. 2010 Second International Conference on Multimedia and Information Technology. doi:10.1109/mmit.2010.161

[5] Sanjay K, Ari V, Timo H (2017) Machine learning classification model for network based intrusion detection system. In: 11th international conference for internet technology and secured transactions (ICITST).

[6] Sravani, K., and P. Srinivasu. "Comparative study of machine learning algorithm for intrusion detection system." In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, pp. 189-196. Cham: Springer International Publishing, 2013.

[7] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., & Lin, W.-Y. (2009). Intrusion detection by machine learning: A review. Expert Systems with Applications, 36(10), 11994–12000.

[8] Pieprzyk J., Hardjono T., Seberry J. (2003) Intrusion Detection. In: Fundamentals of Computer Security. Springer, Berlin, Heidelberg.

[9] Mehrotra, L., Saxena, P. S., & Doohan, N. V. (2017). A Data Classification Model: For Effective Classification of Intrusion in an Intrusion Detection System Based on Decision Tree Learning Algorithm. In Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2016, Volume 1 (pp. 61-66). Singapore: Springer Singapore.

[10] A survey on intrusion detection system using machine learning algorithms, S, V Gulghane Shingate, S Bondgulwar, G Awari, P Sagar International Conference on Innovative Data Communication Technologies and 2019 Springer

[11] Kim, B.-J., & Kim, I. K. (2005). Machine Learning Approach to Realtime Intrusion Detection System. Lecture Notes in Computer Science, 153–163.

[12] Chie-Hong L, Yann-Yean S, Yu-Chun L, Shie-Jue L (2017) Machine learning based network intrusion detection. In: IEEE 2nd international conference on computational intelligence and applications, pp 79–83

[13] A. Abbasi, J. Wetzels, W. Bokslag, E. Zambon, and S. Etalle, "On emulation-based network intrusion detection systems," in Research in attacks, intrusions and defenses: 17th international symposium, RAID 2014, Gothenburg, Sweden, September 17–19, 2014. Proceedings, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham: Springer International Publishing, 2014, pp. 384–404

[14] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges.

[15] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," J Netw Comput Appl, vol. 60, pp. 19–31, 1// 2016

[16] Heenan, R.; Moradpoor, N. A Survey of Intrusion Detection System Technologies. In Proceedings of the 1st Post Graduate Cyber Security (PGCS) Symposium, Edinburgh, UK, 10 May 2016.

[17] Moustafa, N.; Slay, J. UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, ACT, Australia, 10–12 November 2015; pp. 1–6.

[18] Al-Daweri, M. S., Zainol Ariffin, K. A., Abdullah, S., & Md. Senan, M. F. E. (2020). An

Analysis of the KDD99 and UNSW-NB15 Datasets for the Intrusion Detection System. Symmetry, 12(10), 1666.

[19] Rajaallah E.M., Chamkar S.A., Ain El Hayat S. (2019) Intrusion Detection Systems: To an Optimal Hybrid Intrusion Detection System. In: Khoukhi F., Bahaj M., Ezziyyani M. (eds) Smart Data and Computational Intelligence. AIT2S 2018. Lecture Notes in Networks and Systems, vol 66. Springer, Cham.

[20] Borhade, Vipul, Aparna Nayak, and R. Dakshayani. "Intrusion detection: A machine learning approach." In Advanced Computing Technologies and Applications: Proceedings of 2nd International Conference on Advanced Computing Technologies and Applications—ICACTA 2020, pp. 555-561. Singapore: Springer Singapore, 2020.