iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35644

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

A Novel Machine Learning based Hybrid Model for Webshell Detection

Dr. Chandrika J[1], Megha D[2], Pooja K P[3], Roja Mallik S P[4], Syeda Shafain Fathima[5]
[1]*Professor, Department of Computer Science, Malnad College of Engineering, Hassan*
[2, 3, 4, 5]*Student, Department of Computer Science, Malnad College of Engineering, Hassan*

Abstract: ***Webshell attack has become a greater cause of concern while major episodes are shifting online. Today different forms of webshell attacks and attack inducing tools are available to hamper the security of computer systems. These attacks strongly escalate the requisite for Machine Learning based detection. In this work, we are going to obtain behavioral-pattern that may be achieved through static or dynamic analysis, afterward we can apply dissimilar ML techniques to identify whether it's web shell or not. Behavioral based Detection methods will be discussed to take advantage from ML algorithms so as to frame social-based web shell recognition and classification model.***
Keywords: Machine Learning, web shell detection, KNN, decision tree, Hybrid model.

I. INTRODUCTION

Webshells have posed threat and challenge to the security of websites and applications hosted in the internet. The weekly safety report of National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC) in 2019, has crucially stated that the backdoors for the websites are growing in a rapid phase[1]. Vulnerabilities in web-based applications are strategically exploited through backdoors. Malicious file inclusion through SQL injection is achieved where loop hole in server security configuration is made use of. Webshells with ransomware capabilities can induce brute-force attacks, botnet command and control for denial of service(DDoS) attacks and many other dangerous activities. Obfuscation is often used to bypass endpoint security software and antivirus software deployed.

Web shells are installed through vulnerabilities in web application or weak server security configuration, include SQL injection, vulnerabilities in applications and services (e.g. web server software such as NGINX or content management system applications such as WordPress)[2], remote file inclusion (RFI) and local file inclusion (LFI) vulnerabilities; file processing and uploading vulnerabilities, vulnerable plugg-ins in web servers. Webshells for maintaining persistent access after initial attack vector is into the system(PAS webshell) is also established. Different detection methods are widely employed to mitigate webshell attacks[3][4]. This model detects based on behaviors related to network using TCP dump data.

The paper presents hybrid model after including feature extraction, data pre-processing, experimentation and comparison of different machine learning methods. The proposed hybrid model utilizes the behavioral based detection methods to take advantage from Machine Learning algorithms so as to frame social-based webshell attack recognition and classification model.

The rest of the paper is organized as follows :Section 2 reviews the related work, proposed approach is discussed in section 3. Section 4 and 5 discusses empirical analysis and section 6 concludes the paper.

II. LITERATURE REVIEW

[5] Félix Iglesias et al., (2014) have addressed the feature selection problem for network traffic based anomaly detection. A multi-stage feature selection method using filters and stepwise regression wrappers are proposed. This paper has shown the elimination of expensive 13 features out of 41 features, significantly reducing the computational cost and effort enhancing the experimentation and feature generation from live traffic observations at network nodes.

[6] Firdausi et al., (2010) studied the behavior of every malware on an emulated (sandbox) environment. This environment will automatically analyze and generate behavior reports. Preprocessing of the reports into sparse vector models is employed for subsequent machine learning(classification). KNN, Naïve Bayes, J48 Decision Tree, Support Vector Machine (SVM), and Multilayer Perceptron Neural Network (MlP) are the classifiers experimented in this research. The overall best performance was achieved by J48 decision tree. It can be concluded by the proof -of-concept achieved through behavior-based malware analysis using machine learning techniques is effective and efficient for malware detection.

[7] Zhuang Ai et al., (2020) proposed a deep super learner for attack detection. First, the collected data are deduplicated to prevent the influence of duplicate data on the result. Second, to detect the results of the algorithm, static and dynamic feature are taken as the feature of the algorithm to construct a comprehensive feature set. Word2Vec algorithm is used to vectorize the features. During this period, to prevent the outbreak of the number of features, genetic algorithm is used to extract the validity of the feature dimension. Finally, deep super learner is used to detect Web Shell. The experimental results show that this algorithm can effectively detect Web Shell, and its accuracy and recall are greatly improved.

[8] Yixin Wu et al., (2019) demonstrated detection of web shell communication through an intelligent and innovative framework that employs precise sessions derived from the weblogs. Features were extracted from the raw sequence data in weblogs. Session identification is achieved specifically by the time interval-based statistical method. The paper incorporates long short-term memory and hidden Markov model to constitute the framework, respectively. The framework was then evaluated with real-time data. The experiment shows that the LSTM based model can achieve a higher accuracy rate. The experiment results in higher efficiency of the proposed approach in terms of the quick detection without source code, especially when it only considers detecting for a certain period, as it takes 98.5% less time than the cited related approach to get the result.

[9] Muataz Salam Al-Daweri et al., (2020) study presents an extensive analysis of the relevance of individual features in the KDD99 and UNSW-NB15 datasets. The major 3 methods employed consist of rough-set theory (RST), a back-propagation neural network (BPNN), and a discrete variant of the cuttlefish algorithm (D-CFA). Initially, the dependency ratio was calculated between the features and the classes, using the RST. In the next step, each feature in the datasets is fed to the BPNN as input, inorder to measure the ability for a classification task concerning each class. Later, a feature-selection process was undertaken iteratively, to indicate the frequency of the selection of each feature. The results indicate that certain features in the KDD99 dataset can be used to achieve a classification accuracy above 84%. High contributing features were obtained which are a combination of features that yield higher accuracy. This study is expected to fuel cybersecurity enthusiasts for the creation of a lightweight and accurate IDS model with apt features.

III. PROPOSED SYSTEM

The objective of the proposed hybrid model is to detect attack according to the behaviors related to networks, including socket monitoring behavior like TCP/UDP/HTTP request sending. The intrusion detector learning task is to build a predictive model (i.e. a classifier) capable of distinguishing between “bad” connections called attacks/anomalies and “good” normal connections. It is composed of several components, as illustrated in Figure 1.

A. Dataset

The dataset used is NSL-KDD. NSL-KDD is an updated alternative to KDD’99 dataset. MIT Lincoln Labs had directed the 1998 DARPA Intrusion Detection Evaluation Program with an objective to survey and evaluate research in intrusion detection. A U.S Air Force LAN in a well-set-up environment was stimulated with multiple attacks to acquire raw TCP dump data. NSL KDD solves ingrained issues in KDD, it does not include redundant records in the train set, so the classifiers will not be biased towards more frequent records. Test sets of NSL is also devoid of redundant records; therefore, the performance of the learners is seldom biased by the methods which have better detection rates on the frequent records.

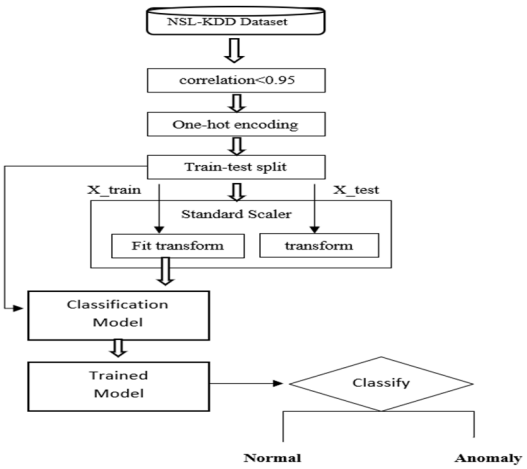


Figure 1: Architecture of the proposed model

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

B. Data Preparation

The dataset synthesized for this project contains approximately 1lakh rows and 42 columns including label. As the project aims at detecting if the connection is normal or attack all the types of attacks are labelled as 'anomaly' and good connections are labelled as 'normal'.

C. Feature Selection

Logistic Out of 41 independent variables, not relevant feature 'service' which states the service at the destination is removed. Correlation between the columns are checked Pearson correlation is applied.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

r = correlation coefficient
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i= values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

Columns with correlation greater than 0.95 are removed as higher correlation makes the model more sensitive. This reduces the dataset to 35 features.

D. Encoding

Furthermore, one hot encoding is applied to categorical variables. (protocol type and flag). The input to this transformer is an array-like integers or strings, denoting the values taken on by categorical (discrete) features. The features are encoded using a one-hot frequently called as 'dummy' encoding scheme. This creates a binary column for each category and returns a sparse matrix.

E. Train-test-split

After processing our dataset to a certain considerable level, the next step is to specify the input and target variables present. Our input will be every column except the 'label' column, since that's what we're attempting to predict—hence, it's our target variable. The data is then split into training and test sets, and a random seed of 42 is specified for the purpose of reproducing the results.

F. Standardization

Standard Scaler function performs Standard Normal Distribution (SND).

$z = (x - u) / s$

where u is the mean of the training samples and s is the standard deviation of the training samples

IV. EXPERIMENTATION AND EVALUATION

Different classification model were experimented and evaluated for accuracy to get the basic idea of well performing models. The selected models include Naïve Bayes Bernoulli, KNN, Decision Tree, Naïve Bayes Gaussian , Logistic Regression.

A. Hyperparameter Tuning

The selected models upon experimentation are subjected to tuning their respective parameters. RandomizedSearchCV considers a set of possible hyperparameters as an input. It exploits random search over the random combinations of hyperparameters supplied and the best score yielding combination of parameters for the targeted model is found. RandomizedSearchCV implements a "fit" and a "score" method. The parameters are optimized using cross-validated search over parameter settings.

The best performing model were decision tree and KNN.

B. KNN

K-Nearest Neighbor is a data mining supervised classifier. The output of the target variable is predicted by finding the k closest neighbor, by calculating the Euclidean Distance. It is a non- parametric classification technique which does not make any assumptions about underlying data.

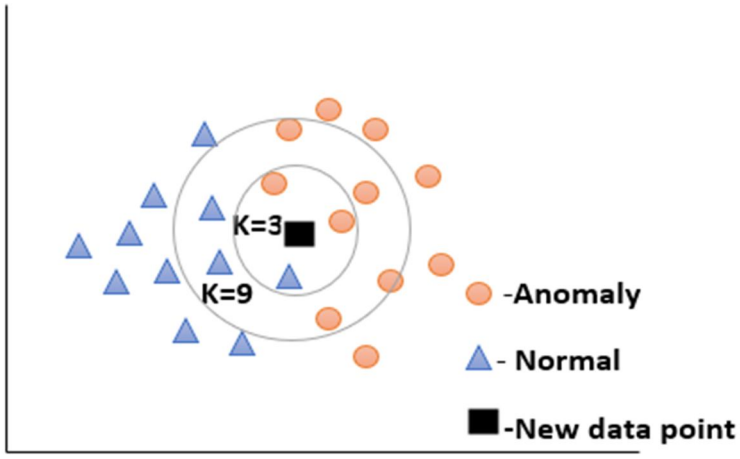


Figure 3 : KNN model classification

C. Decision Tree

Decision trees split the node based on predictor variables or through information gained. One of the attributes at the root node is selected to split and classify. The main approach is to select the attributes, which best divides the data items into their classes. Based on the values of these attributes the data items are partitioned(figure 2). This process is recursively applied to every partitioned subset of the data items. Purity (homogeneous) of the data is tried to achieve in every splits. The process terminates when all the data items in current subset belongs homogenous class. A node of a decision tree specifies the test or condition of split and branches represent the outcome.

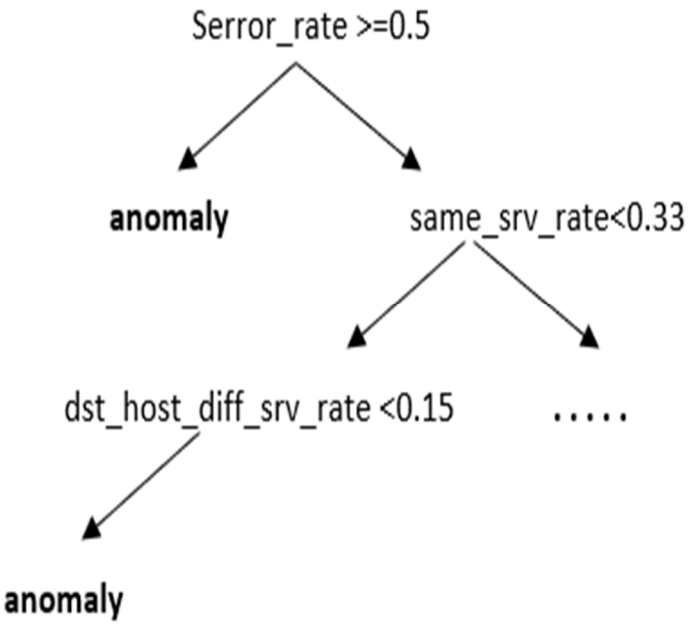


Figure 2: Sample decision tree

D. Hybrid Model

KNN and Decision Tree models are selected to ensemble for hybrid models. We have ensembled five KNN models with best parameters obtained from different iterations of random search as shown in figure(4) and one decision tree and two KNN models are ensembled for another hybrid model. Figure(5).

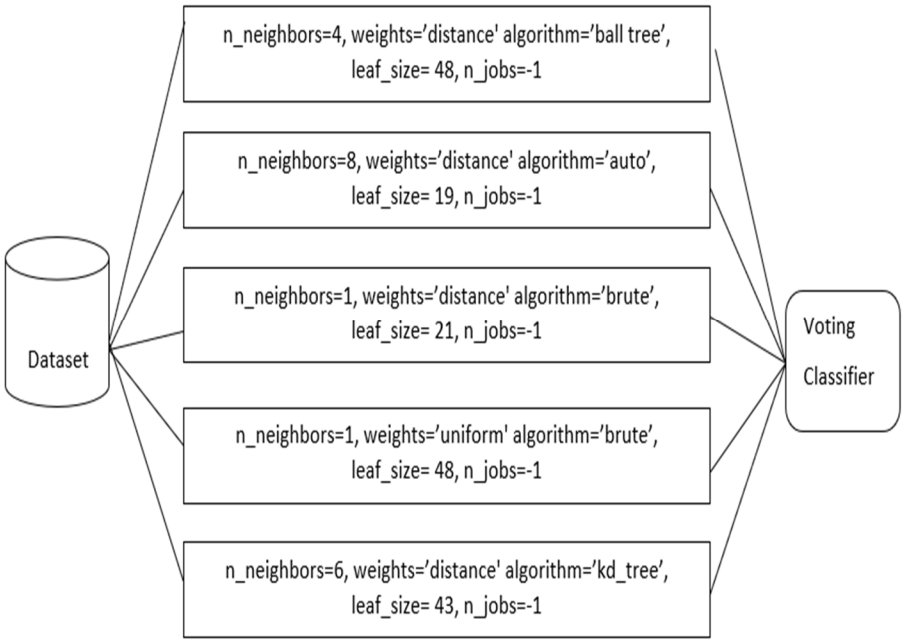


Figure 4: Hybrid model of 5 KNN with different parameters.

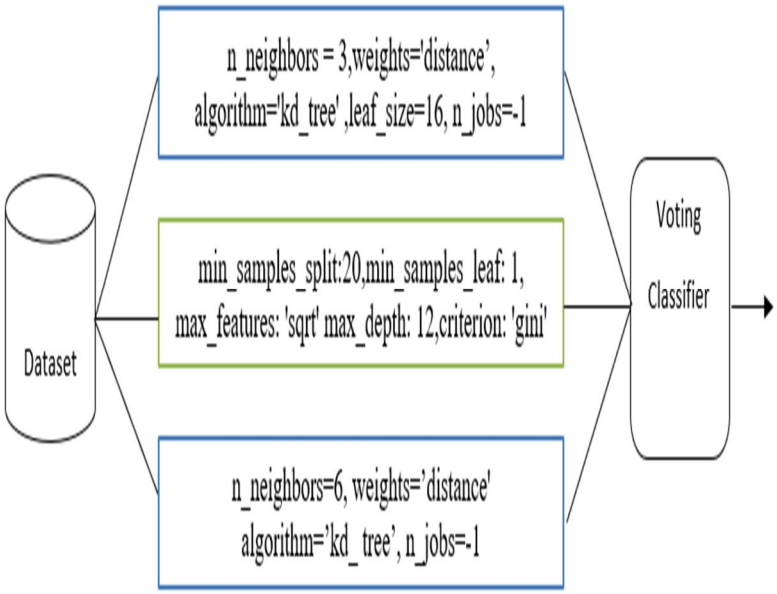


Figure 5: Hybrid model involving 1 Decision tree and 2 KNN

E. Voting Classifier

Voting classifier is applied with hard voting for output. It simply aggregates the findings of each classifier passed into Voting Classifier and predicts the output class based on the highest majority of voting. Hard voting is applied. Suppose three classifiers predicted the output class(anomaly, anomaly, normal), so here the majority predicted is 'anomaly' . Hence 'anomaly' will be the final prediction. This is how hard voting functions.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

V. RESULTS

A. *Accuracy Score*

The accuracy score of different models are plotted in the figure 6. We observe that ensemble methods have yielded better accuracy than individual models. Amongst the hybrid models , the model involving decision tree has performed with higher accuracy rate.

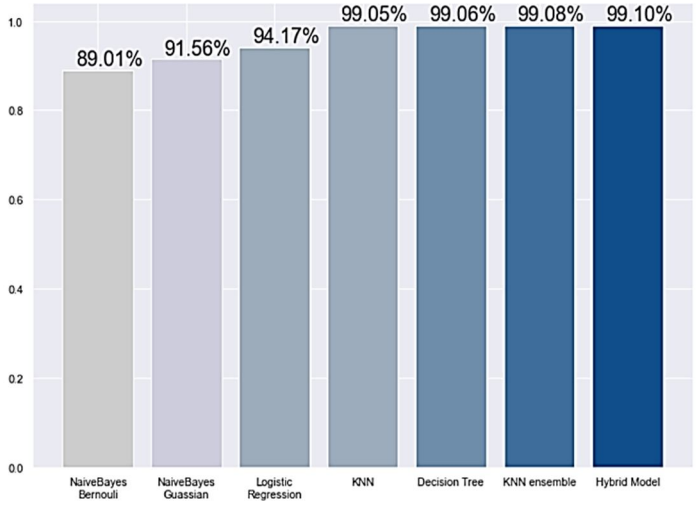


Figure 6: Plot of scores of different models

B. *Classification Report*

The dataset used for the experiment is unbalanced, so evaluation metrics such as accuracy, recall, precision, *F*1 score ,the receiver operating characteristics (ROC) curves, and area under curve (AUC) measure are used to evaluate the proposed method. Visualization of the relation between TPR and FPR of a classifier is depicted by AUC.. These indicators can be expressed as

$$\text{Accuracy} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{TN}_i + \text{FP}_i + \text{FN}_i},$$

$$\text{Recall} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i},$$

$$\text{Precision} = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i},$$

$$F1\ \text{score} = 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

$$\text{AUC} = \frac{\sum_{i \in \text{positiveclass}} \text{rank}_i - (M(1+M)/2)}{M \times N}.$$

where *M* is the number of positive samples and *N* is the number of negative samples. The probability of test sample belonging to positive class is indicated by score, , while rank is a positive sample set sorted in the descending order based on score.

Classification Report of highest performing hybrid model(2 KNN and 1 Decision Tree)

	precision	recall	F1-score	support
anomaly	0.9899	0.9911	0.9905	10465
normal	0.9921	0.9910	0.9915	11735
accuracy			0.9910	22200
Macro avg	0.9910	0.9910	0.9910	22200
Weighted avg	0.9910	0.9910	0.9910	22200

Table 1: Classification Report of Hybrid Model

10-folder cross validation is also applied to calculate accuracy. The dataset is equally divided into 10 subsets, 9 folds are held for training and one set is retained for testing the trained model. The cross validation is repeated 10 times, one subset out of 10 is chosen each time as a test set. The average cross validation recognition accuracy rate is found to be 99.36486 %.

The plot of minimum sample split of the decision tree versus accuracy score is plotted in figure(7). The plot gives interesting results. It shows a gradual decrease in the accuracy as the splits increases. But there is frequent spike in the graph along the minimum split. It is to be noted that other factors are kept constant

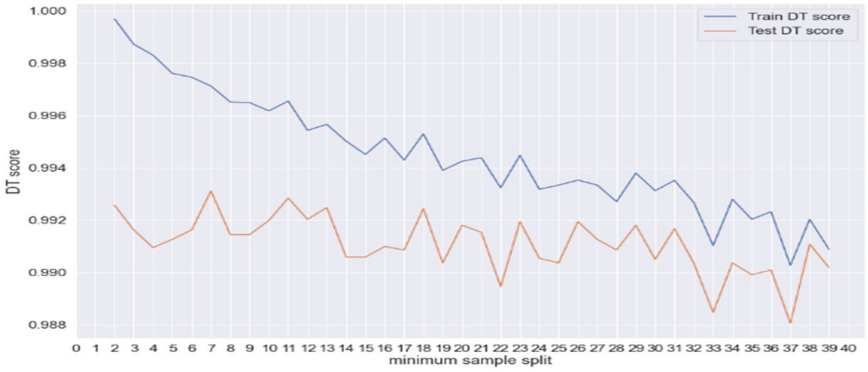


Figure 7: Characteristics of minimum sample split with respect to Decision tree

ROC curve: ROC curve of KNN and Decision tree are plotted in figure(8) and figure(9).

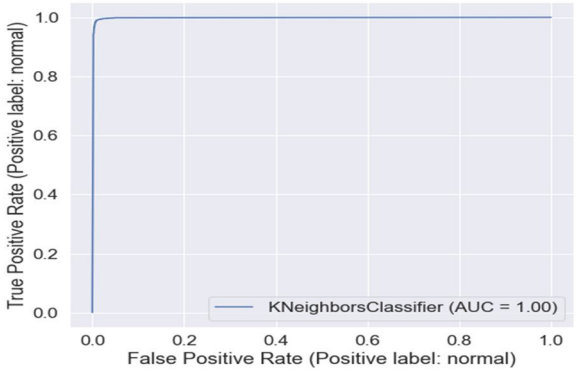


Figure 8 : KNN ROC curve

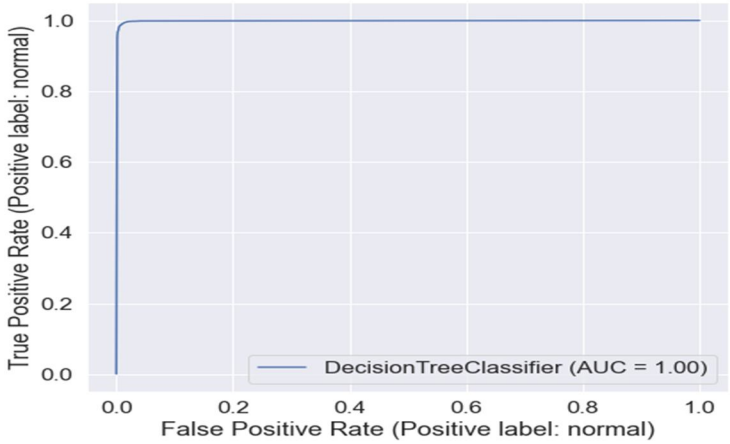


Figure 9 : KNN ROC curve

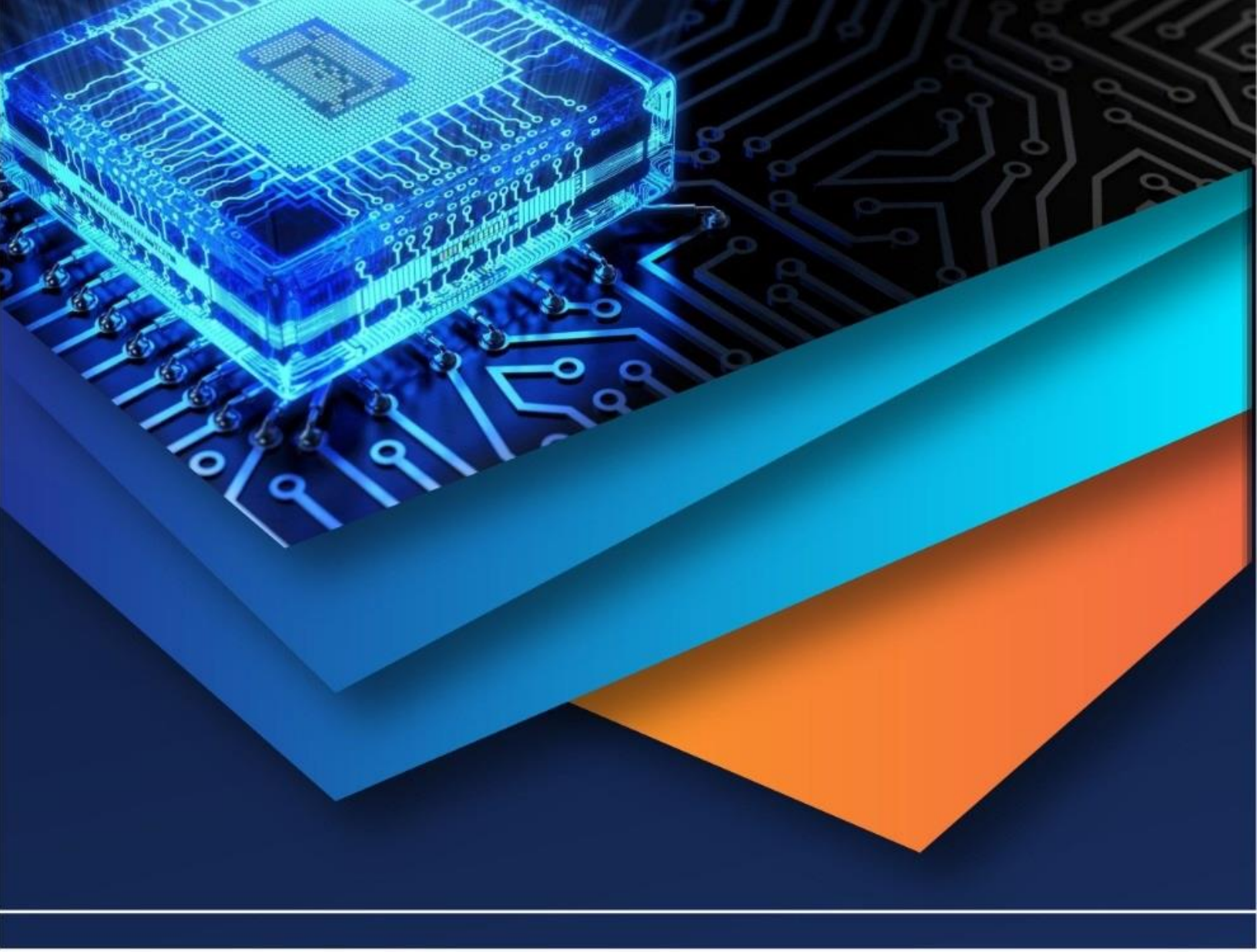
Area under curve is almost equal to 1 due to higher performance.

VI. CONCLUSION

In this paper, we applied different algorithms to NSL-KDD data set to classify normal and anomaly connections. Proposed method is validated by 10 fold cross validation for the classification. The experimental analysis conveys that in comparison with different classification methods, proposed hybrid model have exhibited higher the accuracy, precision, recall and f-measure values. In future, we plan to apply optimization techniques for the classification of the dataset.

REFERENCES

[1] Weekly Report of CNCERT
[2] Article, Dave McMillen(2016) in Security Intelligence "Got WordPress? PHP C99 Webshell Attacks Increasing"
[3] You Guo, Hector Marco-Gisbert and Paul Keir(2020) Mitigating Webshell Attacks through Machine Learning Techniques
[4] Cybersecurity Information (2020). CSI- Detect and Prevent Web Shell Malware
[5] Iglesias, Félix; Zseby, Tanja (2015). Analysis of network traffic features for anomaly detection. Machine Learning, 101(1-3), 59–84. doi:10.1007/s10994-014-5473-9
[6] Firdausi, Ivan; lim, Charles; Erwin, Alva; Nugroho, Anto Satriyo (2010). [IEEE 2010 Second International Conference on Advances in Computing, Control and Telecommunication Technologies (ACT) - Jakarta, Indonesia (2010.12.2-2010.12.3)], Second International Conference on Advances in Computing, Control, and Telecommunication Technologies - Analysis of Machine learning Techniques Used in Behavior-Based Malware Detection. , (), 201–203. doi:10.1109/ACT.2010.33
[7] Zhuang Ai , Nurbol Luktarhan , AiJun Zhou and Dan Lv (2020). WebShell Attack Detection Based on a Deep Super Learner
[8] Yixin Wu, Yuqiang Sun, Cheng Huang , Peng Jia, and Luping Liu (2019) Session-Based Webshell Detection Using Machine Learning in Web Logs
[9] Muataz Salam Al-Daweri, Khairul Akram Zainol Ariffin, Salwani Abdullah, Mohamad Firham Efendy Md. Senan (2020). An Analysis of the KDD99 and UNSW-NB15 Datasets for the Intrusion Detection System



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY