International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 05, May-2020

Study on Decision Tree and KNN Algorithm for Intrusion Detection System

Ashwini Pathak
MPS in Analytics
Northeastern University
Boston, USA

Sakshi Pathak
B.E. Information Technology
SGSITS
Indore 452001, India

***Abstract*— With the increase in use of network technology and internet in today's world, cyber attacks and corruption of network protocols have become an inevitable part of the system. In order to tackle this, an efficient Intrusion Detection System (IDS) is required. IDS is a system that detects malicious activity by monitoring a system or a network. This paper focuses on implementing machine learning techniques - Decision Tree and KNN on IDS and evaluates the performance of both the techniques based on their accuracy. The performance is calculated after applying the Univariate feature selection technique with ANOVA(Analysis of Variance) and algorithms are executed on the NSL-KDD dataset. The performance of algorithms is calculated by metrics like accuracy, recall, precision and F-score. The two algorithms are compared on the basis of these performance metrics.**

Keyword— ANOVA, Decision Tree, IDS, KNN, NSL-KDD Dataset

I. INTRODUCTION

The main objective of IDS is to protect the network from malicious events. Considering the variety of cyber attacks, the traditional firewall will not be able to protect the network. Therefore, IDS is important to prevent novel attacks that make the system more vulnerable.

The IDS are of two types based on employed detection method:[1-2]. First is Signature based- as the name suggests, it identifies signatures or specific patterns that can be recognized as a malicious information. It works best for only those type of attacks that are known and has no false alarm. Second is anomaly based which implements machine learning techniques and compares current real time traffic to its previously recorded malicious free real time traffic. It is effective in recognizing unknown attacks but it gives rise to higher false negatives. Therefore, it is most widely used.

The IDS are of three types based on where the attack is taking place:[1-2]

a. Host based IDS- It functions on a device. It checks for any malicious activity by comparing a file in its present state with its past state and informs the owner or administrator in case any change is found.

b. Network based IDS- It is placed in the network from where it analyzes all devices. It compares the current traffic on subnet to the known attacks. If any anomaly is detected, it informs it to the administrator.

c. Hybrid based IDS- When both HIDS and NIDS come into play it is called Hybrid based IDS.

II. RELATED WORK

There have been several research work on implementing machine learning algorithm on KDD CUP 99 dataset and NSLKDD dataset. Some work on feature selection for intrusion detection system are done in [3-4]. Reference [3] termed feature selection as an important factor for better accuracy. They found that seven features were not important in the determination of attacks. Reference [5] compared Naïve Bayes and Decision tree algorithm and they found out that decision tree performs better when compared to Naïve Bayes. A method of TCM-KNN is proposed for network anomaly detection in [6] on KDD Cup 99 dataset. KNN algorithm is studied in [7] while a study on Random forest and SVM is done in [8]. Aegean Wi-Fi Intrusion Dataset (AWID) with different machine learning techniques implementation is proposed in [9] with information gain and chi square metrics.

The previous research papers show various machine learning algorithms evaluated on KDD CUP 99, NSL KDD dataset. Various feature selection technique have been applied in research papers. This paper will study Decision tree and KNN and will apply Univariate feature selection with ANOVA. We will compare the algorithms by calculating their performance metrics.

III. DATASET

Dataset that provide the best understanding of various intrusion attacks are KDD dataset. The popular KDDCUP'99 dataset was one of the dataset which was available for network IDS but it had a major problem. The problem is the redundancy of data. By analysis, it is found that 78% of data in train dataset is duplicated and 75% of test dataset is duplicated. The newer KDD CUP '99 is NSL KDD. It has selective records from the KDD CUP '99 and does not have redundant data. It has reasonable number of records in both test and train data set which makes it easier to analyze and eliminate the need of choosing some records from it [10]. NSL KDD dataset description is given in Table 1.[11]

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Table 1. Dataset Description

Feature	Description
Duration	duration of connection(in seconds)
protocol_type	type of protocol
Service	Network type
Flag	Flag status
Src_bytes	Number of bytes transferred from source to destination
Dst_bytes	Number of bytes transferred from destination to source
Land	If connection is to same host land=1, else 0
Wrong_fragment	Number of wrong fragments
Urgent	Number of urgent packets
Hot	Number of “hot” indicators
Num_failed_logins	Number of failed logins
Logged_in	If logged in logged_in=1, else 0
num_compromised	Number of compromised conditions
root_shell	If root shell is obtained root_shell=1, else 0
su_attempted	If “su root” accesses, su_attempted=1, else 0
num_root	Number of accessed roots
num_file_creations	Number of file creations
num_shells	Number of shell prompt
num_access_files	Number of operations on access files
num_outbound_cmds	Number of outbound commands
is_host_login	If login is hot is_host_login=1, else 0
is_guest_login	If login is guest is_guest_login=1, else 0
Count No.	Number of connections to the same host in last 2 seconds
srv_count	Number of connections to the same service in last 2 seconds
serror_rate	Percentage of connection with syn error
srv_serror_rate	Percentage of connection with syn error
rerror_rate	Percentage of connection with rej error
srv_rerror_rate	Percentage of connection with rej error
same_srv_rate	Percentage of connection of same service
diff_srv_rate	Percentage of connection of different service
srv_diff_host_rate	Percentage of connection of different hosts
dst_host_count	Number of connections of same destination host
dst_host_srv_count	Number of connections of same destination host and service
dst_host_same_srv_rate	Percentage of connections having same destination host and service
dst_host_diff_srv_rate	Percentage of connections having different service on current host
dst_host_same_src_port_rate	Percentage of connections of current host having same src port
dst_host_srv_diff_host_rate	Percentage of connection of same service and different hosts
dst_host_serror_rate	Percentage of connections of current host having S0 error
dst_host_srv_serror_rate	Percentage of connections of current host of a service having S0 error
dst_host_rerror_rate	Percentage of connections of current host that have rst error
dst_host_srv_rerror_rate	Percentage of connections of current host of service that have rst error
xAttack	Type of attack

Table 2. presents the attacks types classified in 4 types of attacks namely: Dos, U2R, R2L, Probe.

Table 2. Distribution of attacks

Class	Subclass
DoS	Neptune, back, land, pod, smurf, teardrop, mailbomb, apache2, processtable, udpstorm, worm
U2R	buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, httptunnel
Probe	ipsweep, nmap, portsweep, satan, mscan, saint

There are 4 basic type of class:[10]

1. DoS(Denial of Service)- It is one of the most harmful attacks. These type of attacks restrict the user from using certain services. The attacker tries to overload the system or keep the resources busy in the network and does not allow the user to access services.
2. U2R- In this kind of attack, the attacker tries to gain access to the system as a root user. The attacker tries to gain access to all data of the system and have full control on the server.
3. R2L- In this attack, the attacker tries to gain access to a system by sending some message to the server and gaining access to system from a remote machine. The attacker makes some changes to the server to get access to resources. One of the examples being guessing passwords.
4. Probe attacks- This attack aims to analysing the network, gather information. This attack is generally performed to be able to attack through some other methods later.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Fig 1. shows the number of instances of each attack type in the train dataset while Fig 2. presents data from test dataset. The analysis shows that DoS class has maximum instances in both datasets.

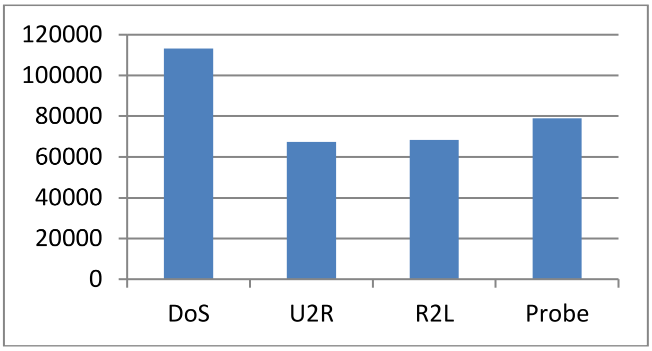

Fig. 1. Number of instances of each type of attack in train dataset

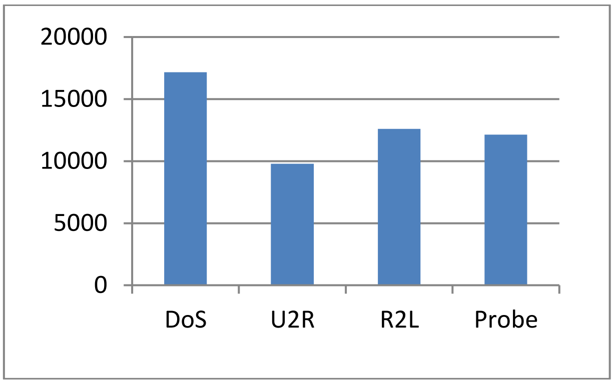

Fig. 2. Number of instances of each type of attack in test dataset

IV. DATA PRE-PROCESSING

Data Pre-processing is done to convert raw data into another format for preparing data for analysing the data further.

1. Handling missing data
2. One Hot Encoding: Since any feature with ‘object/string’ type values will not be valuable to analyse the data, we have to convert categorical data to quantitative variables by using One Hot Encoding. The features having object datatype are: protocol_type, service, flag. We convert these columns to ‘category’ data type and then apply label encoding to convert it into numerical values. These column values are then converted to binary vectors for further analysis. After performing One Hot Encoding the training and testing dataset is divided into 4 parts depending on the type of attack: DoS, U2R, R2L or Probe attack.
3. Data Normalization: It is performed in order to scale the data so that any feature which is comparatively high in values might not be given more weightage. While it is necessary for KNN, it is not useful for Decision Tree algorithm.

V. FEATURE SELECTION

It is one of the most important processes in data preprocessing in machine learning. It keeps only the relevant attributes and removes all the redundant features. Any attribute that does not contribute towards predicting the target value are not important and are removed.

The feature selection technique comprises of 4 steps: First step is the feature selection that selects the required features for a particular problem. Second step is the function assessment that evaluates the set of features selected. Third step is the stopping criteria that identifies if the features will help in stopping the search. Fourth step is the validation procedure that assesses the features and identifies the quality of features. They are divided in two groups: Filter and Wrapper. Filter finds the similarity between attributes and the class. Wrapper evaluates the concerned attributes [12]. The Univariate Feature selection with ANOVA F-test is performed to find the appropriate features for a label by determining the relationship between features and label. The univariate approach chooses the attributes which are related largely with target variable. The class SelectPercentile selects the features which are best associated with the target variable. The metrics chosen is f_classif for finding the best features.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

VI. CLASSIFIERS

Classifiers are given training data, it constructs a model. Then it is supplied testing data and the accuracy of model is calculated. The classifiers used in this paper are :

A. Decision Tree:

Decision Tree algorithm is one of the well known classification method. Decision tree is a tree like graph. It assigns classification based on the rules applied from the root to the leaf of the tree. The internal nodes are test, branch corresponds to the result of test and leaf nodes assigns a classification. The data having least impurity is chosen. This impurity is measured by entropy. Higher the entropy means high impurity.[13]

Decision Tree algorithm:

1. Choose any attribute from the data.
2. Calculate the significance of attribute while splitting the data.
3. Split the data on value of best attribute.
4. Again go to step 1

Gini impurity: This measure is used for classification of trees. It is a variation of entropy calculated for decision tree. If an item is classified according to the distribution of labels, gini impurity is the likelihood of incorrect classification of that item. Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset [14]. The Gini index is defined as [15]:

$$\text{Gini}\ (y, S) = 1 - \sum_{c_j \in \text{dom}(y)} \left(\frac{\left|\sigma_{y=c_j} S\right|}{|S|} \right)^2$$

The criteria for defining a_i is defined as [15]:

$$\text{Gini Gain}\ (a_i, S) = \text{Gini}\ (y, S) - \sum_{v_{i,j} \in \text{dom}(a_i)} \frac{\left|\sigma_{a_i=v_{i,j}} S\right|}{|S|} \cdot \text{Gini}\ \left(y, \sigma_{a_i=v_{i,j}} S\right)$$

Advantages of Decision tree are that it does not require parameter setting. Also, it evaluates all possible outcomes and makes the best possible decision. Disadvantages of Decision tree are that small change in dataset will lead to bigger changes in the decision tree. [16]

B. KNN(K-nearest neighbour): It is a method for classifying similar cases. Also called the lazy learner because it does not have any learning phase. It produces results only when they are requested.

Algorithm for KNN-

1. Choose a value of k
2. Calculate the Euclidean distance of all cases from unknown case.
   The Euclidean distance (also called the least distance) between sample x and y is : [17]

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

where x_i is the i^{th} element of the instance x, y_i is the i^{th} element of the instance y and n is the total number of features in the data set.

3. Choose k number of observations near to the unknown case.
4. The unknown data will belong to the majority cases in k nearest neighbour

Advantages of KNN are that it is one of the simplest algorithms as it has to compute the value of k and the Euclidean distance only. It is faster than majority of other algorithms because of its lazy learning feature. Also, it is very efficient for multiclass problem. Disadvantages of KNN are that the algorithm may not generalize well as it does not go through the learning phase. It is slower for a large dataset as it will have to calculate sort all the distances from the unknown item. KNN algorithm also requires feature scaling for best result.[18-19]

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

VII. RESULTS AND DISCUSSIONS

After training the model, the algorithms: Decision Tree and KNN is tested on a testing dataset. It is represented by a confusion matrix. The confusion matrix presents the value of True positives, true negatives, false positives and false negatives.

		Predicted	
		No	Yes
Actual	**No**	**TN**	**FP**
	Yes	**FN**	**TP**

1. True Negative(TN): It is the number of correct prediction that an instance does not belong to a particular class.
2. False Positive(FP): It is the number of incorrect prediction that an instance belongs to same class when it belongs to some other class.
3. False Negative(FN): It is the number of incorrect prediction that an instance belongs to some other class when it belongs to the same class.
4. True Positive(TP): It is the number of correct prediction that an instance belongs to same class.

The following performance metrics are calculated:

Accuracy = (TP+TN) / (TP+TN+FN+FP)
Recall = TP/ (TP+FN)
Precision = TP/ (TP+FP)
F-Score = 2*(precision*recall) / (precision + recall)

Table 3 shows the result of the implementation of both the algorithms and displays the value of Accuracy, Precision, Recall and F-score for each kind of attack.

Table 3. Experimental Result

Classifier	**Class**	**Accuracy**	**Precision**	**Recall**	**F-score**
Decision Tree	DoS	99.6	99.5	99.6	99.5
	U2R	99.6	86.4	91.6	88.6
	R2L	97.9	97.1	96.9	97
	Probe	99.5	99.3	99.2	99.3
KNN	DoS	99.7	99.6	99.6	99.6
	U2R	99.7	93.2	84.8	87.7
	R2L	96.7	95.3	95.4	95.3
	Probe	99	98.6	98.5	98.5

Fig. 3 compares the accuracy of both the algorithms for all types of attacks. Fig. 4 compares the average value of accuracy, precision, recall and f-score for both algorithms.

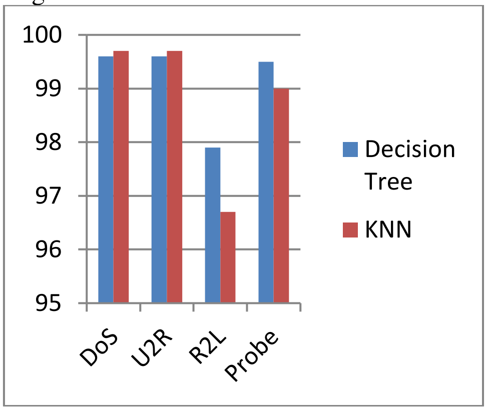

Fig. 3. Comparison of accuracy acquired by the two algorithms: Decision tree and KNN

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

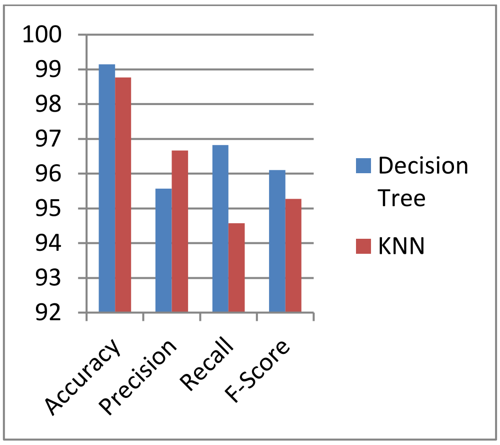

Fig. 4. Comparison of average accuracy, precision, recall and f-score of the two algorithms: Decision tree and KNN

The observations made from the performance are:

1. The average accuracy of Decision tree algorithm is better than KNN.
2. The accuracy for correctly identifying DoS, U2R is slightly better for KNN while the accuracy for correctly identifying R2L and Probe attacks is better for Decision tree.
3. The KNN algorithm reaches high precision.
4. The precision rate for DoS, U2R is higher for KNN while it is higher in case of R2L and Probe attacks for Decision tree algorithm.
5. The recall value and F score is higher for Decision tree algorithm.
6. Decision tree has a fairly low time demand as compared to KNN.

VIII. CONCLUSION

We have met the objective of this paper of studying two algorithms: Decision tree and KNN applied on the NSLKDD dataset. We performed data pre-processing on the NSL KDD dataset by handling missing values, applying one hot encoding and normalizing data. We have successfully applied ANOVA f-test on the dataset and selected the best features for each type of attack. We have acquired the performance of both algorithms on classification of 4 attacks: DoS, U2R, R2L and Probe. We have found that the Decision tree algorithm gives a better result with an accuracy of 99.15%. Also, we found that the time taken to build the Decision tree algorithm was far less than the KNN algorithm. Therefore, the results show that Decision tree algorithm gives an overall better result as compared to KNN.

IX. REFERENCES

[1] Ashoor A S, Gore S. Importance of Intrusion Detection System (IDS). IJSER, 2011, 2(1).
[2] Saxena A K, Sinha S, Shukla P. General study of intrusion detection system and survey of agent based intrusion detection system. ICCCA, 2017, pp. 471-421.
[3] Olusola A A, Oladele A S, Abosede D O. Analysis of KDD ’99 Intrusion Detection Dataset for Selection of Relevance Features. Proceedings of the World Congress on Engineering and Computer Science WCECS, San Francisco, USA, 2010, Vol. 1.
[4] Sung A H,Mukkamala S. Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. Proceedings of the 2003 Symposium on Applications and the Internet, 2003.
[5] Amor N B, Benferhat S, Elouedi Z. Naive Bayes vs decision trees in intrusion detection systems, Proceedings of the ACM Symposium on Applied Computing, 2004, 1: pp. 420-424.
[6] Li Y, Fang B, Guo L, Chen Y. Network Anomaly Detection Based on TCM-KNN Algorithm. Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, 2007, 7, pp. 13-19
[7] Liao Y, Vemuri R V. Use of K-Nearest Neighbor classifier for intrusion detection. Computers & Security, 2002, 21(10): 439-448.
[8] Patgiri R, Varshney U, Akutota T, Kunde R. An Investigation on Intrusion Detection System Using Machine Learning, 2019
[9] Thanthrige U S K P M, Samarabandu J, Wang X. Machine learning techniques for intrusion detection on public dataset . IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2016, pp. 1-4.
[10] Tavallaee M, Bagheri E, Lu W, Ghorbani A A. A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications CISDA, 2009.
[11] Jha J, Ragha L. Intrusion Detection System using Support Vector Machine. International Journal of Applied Information Systems (IJAIS), 2013
[12] Tribak H, Delgado B, Rojas P, Valenzuela O, Pomares H, Rojas I. Statistical analysis of different artificial intelligent techniques applied to Intrusion Detection System. Proceedings of 2012 International Conference on Multimedia Computing and Systems, ICMCS, 2012, pp. 434-440.
[13] Rai K, Devi M, Devi, Guleria A. Decision Tree Based Algorithm for Intrusion Detection. IJANA, 2016, 07:2828-2834
[14] Kumar A, Nidhi A, Michael R. Implementing an Intrusion Detection System using a Decision Tree, 2014.
[15] Rokach L, Maimon O. Top-down induction of decision trees classifiers - a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2005, 35(4), pp. 476-487.
[16] Sahu S K, Sarangi S, Jena S K. A detail analysis on intrusion detection datasets. IEEE International Advance Computing Conference (IACC), 2014, pp. 1348-1353.
[17] Brao B, Swathi K. Fast kNN Classifiers for Network Intrusion Detection System. Indian Journal of Science and Technology, 2017, 10, pp. 1-10.
[18] Ajayi, A, Idowu S A. Comparative study of selected data mining algorithms used for intrusion detection, 2013, 7.
[19] Kim J, Kim B S, Savarese S. Comparing Image Classification Methods: K-Nearest-Neighbor and Support-Vector-Machines. Applied Mathematics in Electrical and Computer Engineering, 2019, 6, pp. 133-138.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)