Performance Analysis of Network Based IDS using Machine Learning Algorithms

Divya Nehra
Punjab University, Chandigarh

Abstract- In today's scenario, a Network based IDS is one of the best security providing solution. Though, no system is perfect some loopholes are always present from where security gets breached. In context to it, we have performed a performance analysis of network based IDS using machine learning algorithms. The benchmark dataset KDD'99 is used. We trained nine classifiers and observed their performance to find out the best classifier among them in the given environment. Parameters used are Accuracy, Precision, Recall, Cohen's Kappa, Jaccard Similarity Coefficient, Hamming Loss and Zero-One loss. In the proposed work binary classification has been performed as attack or normal. The results concluded are very interesting and the analysis is very helpful in choosing a better classifier.

Keywords- Intrusion Detection System, Network IDS, KDD'99 Dataset, Machine Learning,

I. INTRODUCTION

Intrusion Detection System (IDS) are the software that locate and identify the malicious activity by analysing network traffic in real time. The IDS which works to analyse the network traffic and network logs, among hosts, is termed as Network based intrusion detection system. Network IDS looks for the patterns or signatures of nefarious behaviour. It uses the concept of baselining i.e. a method of observing computer network performance by comparing current performance to a typical metric[1]. Network IDS analyses the network traffic which is being transmitted, forwarded or received over a network link and detects the intrusive actions. The effectiveness of NIDS revolves around its capability in detecting the intrusion efficiently without false alarms[2]. To summarize, this paper presents the introduction to Network based IDS and the related work in the same domain. Then the dataset used which is KDD'99 is described and on this basis the purpose of this paper is to design a approach for designing aprecise model out of other nine trained models in the terms of highest accuracy, True positive rate, precision, F1-Score, Cohen's Kappa, Jaccard_score and lowest hamming loss , Zero-one loss.

II. RELATED WORK

Previous studies have proposed a number of techniques based on conventional machine learning. . The Network IDS model proposed by authors in [3]is able to detect malicious traffic with accuracy of 97.5%. The model was also able to detect false positives with True Positive Rate of around 99.6%. The model was efficient to detect unseen traffic.The performance analysis done by the authors in [4] is to determine the best and optimal parameters for the classifiers without using the default values for the tool used. The computed results say that there is no best algorithm that outperforms others. The work done in [5] have created an evaluation metrics made up of combination of accuracy, detection rate and false alarm rate in a way that helps in selecting the best classifier among them. The approach used in [6] is performance bound ensemble of classifier to build a strong classifier that stands out of all the weak classifier. The model proposed in [7] isa Network based IDS to study the performance of classifiers using machine learning.

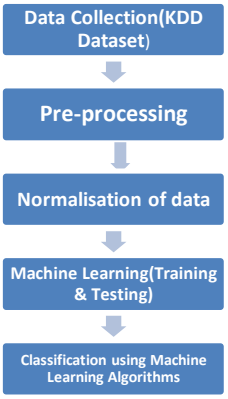

Fig.1: Functioning of Network IDS

III. DATASET USED

KDD'99 DATASET: KDD'99 is a freely accessible benchmark dataset. KDD'99 consists of data captured in DARPA'98 IDS assessment program. KDD dataset uses the information at TCP/IP level and entrenched among domain particular heuristics, to identify intrusions on network stage. The major drawback in KDD dataset is,it contains large amount of extraneous and irredundant data[8]. The simulated attacks has four categories of attacks: Probe, Denial of service (DoS), User-to-Root (U2R) and Remote-to-local(R2L) attacks[9]. The attacks fall in four categories generally:

1) Probing Attack: In probe attacks, the attacker scans the system to find vulnerabilities. These attacks are performed using various amenities to find the computer systems which are actually responding actively to the network. The vulnerability found may later be exploited to deteriorate services for legitimate users. Some probing attacks are Portsweep, Nmap, Satan and IPsweep.

2) Denial of Service Attack (DoS): These attacks are basically network level attacks in which hacker sends malicious traffic to a web server to disrupt the availability of resources to authenticate users hence denying users access to

machine. Moreover, when attacks are made from a single system it is called a Denial of Service attack and when the attacks are made from multiple systems to overflow network traffic, it is known as DDoS. Examples of DoS are: - ARP, UDP Storm, Teardrop Attack etc.

3) Remote to Local Attack (R2L): Remote to local attack. In this attack the intruder has no account on host and tries to gain local access over network. To perform R2L type of attack the buffer flow attacks are performed.

4) User to Root Attack (U2R): These attacks are the exploitations of flaws in OS. A local user gains the super user privileges to access the confidential files. Example: -Xterm, perl, Ntfs DOS, SQLAttack.

There are 4900000 single connection records in the dataset which contains 41 features and labelled as either attack or normal with specific type of attack.The KDD dataset features are divided into three broad categories:

1) Basic Features(B): Those features which are extracted from a TCP/IP link and leads to hidden delay in detection. Feature No.1 to feature No.10 is basic features[10].

2) Content Features(C): Those characteristics which are used for recognising the attacks having very low frequency. These features are able to find out malicious actions in the statistics section like gaining the root access during session. Feature No.11 to feature No.22 is content features[10].

3) Traffic Features (T): Those features which are computed with respect to a gap interval. Feature No.23 to feature No.41 is traffic features[10]

A. Dataset Transformation:

The dataset used consists of 490000 instances where each instance has 42 features with binary classification as "normal" or "attack". These instances require transformation of nominal or symbolic features to the numerical form. For example, ICMP protocol type is mapped to 0, TCP is mapped as 1 and UDP is mapped as 2. Table 2 can be used for more clarification regarding the same. Similarly all the features are transformed.

Feature	Type	Conversion
Protocol	ICMP	0
	TCP	1
	UDP	2
Flag	SF	9
	S0	5
	REJ	1
	RSTR	4
	RST0	2
	SH	10
	S1	6
	S2	7
	RSTOS0	3
	S3	8
	OTH	0

Table 1:Transformation Table for Protocol and Flag Values

B. **Dataset normalization**

To enhance the performance of IDS over big dataset, the fundamental process, i.e. normalization is performed. This step is very important part of pre-processing and ensures that all values of the attribute fall purely in the range of 0 to 1. The normalisation is done only on continuous values.In the proposed modelMin-Max normalization has been used, using the following equation no. 9:

$$x_i = \frac{Vi - \min(Vi)}{\max(Vi) - \min(Vi)} \qquad \text{Eq.9}$$

Where $x_{i=normalised\ value}$; $x_{i=0,\ when\ max=min}$

v_i $= concerned\ value\ of\ the\ attribute\ which\ is\ to\ be\ normalised$

Original KDD Dataset Record
0 tcp http SF 181 5450 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 8 8 0 0 0 0 1 0 0 9 9 1 0 0.11 0 0 0 0 0 normal
Transformed and Normalised KDD Dataset Record
0 1 22 9 .28 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 11 0 0 0 0 .14 0 10 0.3 19 0 2 0 0 0 0 0 1

Fig.2: KDD Dataset Records

IV. PERFORMANCE EVALUATION

a. Random Forest Classifier: A Random Forest Classifier consists of a collection of various tree structured classifier and it is an ensemble method used for classification. The independent identically distributed random decision trees are grown by classifier and each tree casts a unit vote for majority to predict class label[11]. Selected parameters for the model are:

max_depth= None (It is the maximum depth of a tree)

random_state=None (It is the seed used by the random number generator)

n_jobs= 1 (These are the number of jobs to run in parallel for both fit and predict)

n_estimators=10 , Criterion= "gini" (It is the function to measure the quality of a split)

Verbose=0 (It controls the verbosity of the building process)

class_weight= None(It is the weight associated with classes)

Confusion Matrix is an n*n matrix which is generated after a classifier is trained.

The confusion matrix can be formed using (Eq.1)

$$\text{Confusion Matrix} = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

Eq.1

- **True Negative (TN):** Those instances which are correctly predicted as normal or non-intrusive action.
- **True Positive (TP):** Those instances which are correctly predicted as an attack or intrusive action.
- **False Negative (FN):** Those instances which are falsely predicted as normal or non-intrusive action.

- **False Positive (FP):** Those instances which are falsely predicted as an attack or intrusive action.

The confusion matrix for Random Forest Classifieris:-

$$\begin{bmatrix} 79300 & 17 \\ 6 & 19482 \end{bmatrix}$$

b. K-Nearest Neighbour: A k-NN classifier is a type of instance based learning or lazy learning classifier. An entity is classified by a bulk vote of its neighbours, with the object being assigned to the class most common amongst its k-NN where k is a positive integer, generally small[12] For eg: when k=1, then the entity is assigned to the class of its nearest neighbour. Selected parameters for the model are:n_neighbours=5; weights='uniform'; algorithm='auto'; leaf_size=30; metric='minkowski'; metric_params=None; n_jobs=1

The confusion matrix for k-Nearest Neighbour is:-

$$\begin{bmatrix} 79297 & 20 \\ 22 & 19466 \end{bmatrix}$$

c. Gaussian Naive Bayes: This classifier is a part of Naive Bayes family. These aresimple probabilistic classifiers and use a very popular baselining method of text classification[13]. They are highly scalable and require number of parameters to work:

To fit X,Y according to Gaussian Naive Bayes we used:fit (self, X, Y, sample_weight =None) where
X is an array and Training Vectors are:
n_samples(These are the number of samples)
n_features(These are the number of features)

The confusion matrix for GaussianNaive Bayes :

$$\begin{bmatrix} 78981 & 336 \\ 5355 & 14133 \end{bmatrix}$$

d. Perceptron: These classifiers make use of very simple learning algorithm which is suitable for large scale learning. Perceptron is little faster to train than Stochastic gradient descent classifier and updates its model only on mistakes. The parameters used for perceptron are:

verbose=0; n_jobs=1; random_state=0; class_weight=none; warm_start=False; fit_intercept=True; learning_rate='constant'

The confusion matrix for Perceptron is:

$$\begin{bmatrix} 78563 & 754 \\ 56 & 19432 \end{bmatrix}$$

e. Linear SVC: It is similar to Support Vector Classifier but has more flexibility is terms of penalties and loss function. It accepts the data in numerical form only and if given categorical input conversion to binary dummy values is required.

LinearSVC has been implemented using liblinear. Shrinking= True ; probability=False ; cache_size= 200; class_weight=None ; verbose=False ; max_iter = -1, decision_function_shape= 'ovr'; random_state=None.

The confusion matrix for LinearSVC is:

$$\begin{bmatrix} 79206 & 111 \\ 76 & 19412 \end{bmatrix}$$

f. Support Vector Classifier: This classifier has the best learning algorithm for binary classification. Support Vector Classifier has become one of the most popular classifier for anomaly detection. The parameters used are exactly same as of LinearSVC .

The confusion matrix for Support Vector Classifier is:

$$\begin{bmatrix} 79138 & 179 \\ 248 & 19240 \end{bmatrix}$$

g. Stochastic Gradient Descent: It uses an optimization technique for minimizing multidimensional smooth convex objective functions. Stochastic Gradient Descentfastens the optimization process significantly to obtain results comparable to state-of-the-art to other techniques[13].

Parameters used are: loss='hinge'; penalty=L2; verbose=0; n_jobs=1; random_state=None; epsilon=DEFAULT_EPSILON; n_iter=None.

The confusion matrix for Stochastic Gradient Descent is:

$$\begin{bmatrix} 79120 & 197 \\ 143 & 19345 \end{bmatrix}$$

h. Decision Tree Classifier: It works well with big data set. The high performance among big data set makes it more useful in real time intrusion detection system. They are also efficient in working with rule-based models where requirement of processing is less[14].

The parameters required are: splitter="best"; min_samples_split=2; min_samples_leaf=1; min_weight_fraction_leaf=0; max_features= None; max_leaf_nodes= None; presort= False;

The computed confusion matrix for Decision Tree is:

$$\begin{bmatrix} 79301 & 16 \\ 19 & 19469 \end{bmatrix}$$

i. Logistic Regression: It is a linear model for classification before regression. Logistic Regression is also called as maximum-likelihood estimation, maximum-entropy classification (MaxEnt) or the log-linear classifier[15]. The parameters required are:

Loss="hinge'; penalty=L2; L1 ratio= 0.15; learning_rate='optimal'; warm start= False;

The confusion matrix for Logistic Regression is:-

$$\begin{bmatrix} 79223 & 94 \\ 61 & 19427 \end{bmatrix}$$

V. RESULTS AND DISCUSSIONS

1.**Accuracy**:It is the ratio of correctly identified records versus the total number of records. It can be calculated using (Eq.2)

$$\mathbf{AC=}\frac{TP+TN}{TP+FP+TN+FN} \quad \mathbf{Eq.2}$$

2. Recall: It is the percentage of intrusive-actions which are correctly predicted. It is the True Positive rate. It is also known as sensitivity and can be calculated using (Eq.3)

$$\mathbf{TPR}=\frac{TP}{(TP+FN)} \quad \textbf{Eq.3}$$

3. Precision: It is the probability of positive prediction being correct. It is also called as positive predictive value and can be calculated using (Eq.4)

$$\mathbf{PPV}=\frac{TP}{(TP+FP)} \quad \textbf{Eq.4}$$

4. F1-Score: It is the harmonic mean of Recall and Precision. Eq.5 can be used to calculate the value for F1-Score.

$$\mathbf{F1\text{-}Score}=\frac{2TP}{(2TP+FP+FN)} \quad \textbf{Eq.5}$$

5. Cohen's Kappa(K): It is the measure of the agreement between two setstoclassify the N instances into X exclusive classes.The higher value is desirable for K. It can be computed using (Eq.6)

$$\mathbf{K}=\frac{P_{0-P_c}}{1-P_c} \text{ where } P_0=\frac{TP+TN}{TP+TN+FP+FN}$$

$$P_c=P_{yes}+P_{no}$$

$$P_{yes}=\frac{TP+TN}{TP+TN+FP+FN}*\frac{TP+FP}{TP+TN+FP+FN}$$

$$P_{no}=\frac{FP+TN}{TP+TN+FP+FN}*\frac{FN+TN}{TP+TN+FP+FN} \quad \textbf{Eq.6}$$

6. Jaccard Similarity Coefficient Score: It is the measure of similarity for data of two sets. It has a rangeof 0%to 100%. The higher value is desirable.It can be computed using (Eq.7)

$$J=\frac{TP}{FP+FN+TP} \quad \textbf{Eq.7}$$

7. Hamming Loss: It is the measure of wrongly predicted label to the total number of labels. It is similar to the hamming distance. The lower value is desirable.

8. Zero-One loss: It is also known as error rate. It gives the value for the output of the data for test dataset which are not same as the output of the data of training dataset with different features. The lower value is desirable. The result for Zero-one loss can be computed using (Eq. 8)

$$\textbf{Zero-one loss= (1- Accuracy)} \quad \textbf{Eq.8}$$

Table 1: Performance Evaluation

Classifiers	AC	PPV	TPR	F1-Score	Cohen's Kappa	Jaccard Index	Hamming Loss	Zero-One Loss
RFC	**0.999817**	**1.00**	**1.00**	**1.00**	**0.999269**	**0.999757**	**0.000242**	**0.000242**
KNC	0.999545	1.00	1.00	1.00	0.998631	0.999545	0.000455	0.000455
GNB	0.896989	0.97	0.90	0.93	0.717723	0.896989	0.103010	0.103010
PCP	0.991751	0.99	0.99	0.99	0.975369	0.991751	0.008248	0.008248
LSVC	0.997651	1.00	1.00	1.00	0.992937	0.997651	0.002348	0.002348
SVC	0.995203	1.00	1.00	1.00	0.985536	0.995203	0.004797	0.004797
SGDC	0.996285	1.00	1.00	1.00	0.988816	0.996285	0.003714	0.003714
DTC	0.999645	1.00	1.00	1.00	0.998934	0.999645	0.000354	0.000354
LR	0.998390	1.00	1.00	1.00	0.995160	0.998390	0.001609	0.001609

VI. CONCLUSION AND FUTURE WORK

In this study performance analysis of network based IDS using machine learning algorithms is performed.This study proves that Random Forest Classifier provides better accuracy among the nine classifiers. To enhance the results Cohen's Kappa, Jaccard_Score, Hamming Loss and Zero-One Loss values are also applied. The benchmark dataset KDD'99 is used and data transformation and normalization on original data is performed to ensure that data falls in a particular range. To evaluate the performance all the parameters used by each and every classifier has been mentioned.In this proposed work, it is not made conventional that RFC is the best classifier but it performed better than the other classifiers in the given circumstances and scenarios. Though there have been major advances in technology but still a gap exists where less-capable techniques are being used. Future work may focus on deeper analysis using more granular datasets.

VII. REFERENCES

[1]. "No Title." [Online]. Available: www.youtube.com/asianvault. [Accessed: 17-Apr-2018].

[2]. G. Kumar, "Evaluation Metrics for Intrusion Detection Systems - A Study," Eval. Metrics Intrusion Detect. Syst. - A Study, vol. 2, no. 11, pp. 11–17, 2014.

[3]. S. Kumar, A. Viinikainen, and T. Hamalainen, "Machine learning classification model for Network based Intrusion Detection System," Internet Technol. Secur. Trans. (ICITST), 2016 11th Int. Conf., pp. 242–249, 2016.

[4]. M. Anbar, R. Abdullah, I. H. Hasbullah, Y.-W. Chong, and O. E. Elejla, "Comparative performance analysis of classification algorithms for intrusion detection system," 2016 14th Annu. Conf. Privacy, Secur. Trust, pp. 282–288, 2016.

[5]. T. Hamed, R. Dara, and S. C. Kremer, "Network intrusion detection system based on recursive feature addition and bigram technique," Comput. Secur., vol. 73, pp. 137–155, 2018.

[6]. N. N. P. Mkuzangwe and F. Nelwamondo, "Ensemble of Classifiers Based Network Intrusion Detection System Performance Bound," no. Icsai, pp. 970–974, 2017.

[7]. V. Das, V. Pathak, S. Sharma, and M. Srikanth, "NETWORK INTRUSION DETECTION SYSTEM BASED ON MACHINE LEARNING," vol. 2, no. 6, pp. 138–151, 2010.

[8]. G. Meena and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA," 2017 Int. Conf. Comput. Commun. Electron. COMPTELIX 2017, pp. 553–558, 2017.

[9]. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009, no. Cisda, pp. 1–6, 2009.

[10]. Y. Chuan-long, Z. Yue-fei, F. Jin-long, and H. Xin-zheng, "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks," IEEE Access, vol. 5, pp. 1–1, 2017.

[11]. "Random Forest Classifier." [Online]. Available: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf. [Accessed: 19-Apr-2018].

[12]. K. Law, "IDS false alarm filtering using KNN classifier," 5th Int. Work. WISA, Revis. Sel. Pap., pp. 114–121, 2005.

[13]. E. Alpaydın, "Introduction to machine learning," Methods Mol. Biol., vol. 1107, pp. 105–128, 2014.

[14]. M. Kumar, M. Hanumanthappa, and T. V. S. Kumar, "Intrusion detection system using decision tree algorithm," Commun. Technol. (ICCT), 2012 IEEE 14th Int. Conf., pp. 629--634, 2012.

[15]. D. Y. Mahmood, "Classification Trees with Logistic Regression Functions for Network Based Intrusion Detection System," IOSR J. Comput. Eng., vol. 19, no. 3, pp. 48–52, 2017.