Comprehensive Performance Evaluation of Network Intrusion System Using Machine Learning Approach

Muhammad Shahzad Haroon, Dr Syed Sajjad Hussain

***Abstract*— Over the last three decades, network devices are increasing due to technology like the Internet of Things (IoT) and Bring Your Own Device (BYOD). These rapidly increasing devices open many venues for network attacks whereas modern attacks are more sophisticated and complex to detect. To detect these attacks efficiently, we have used recently available dataset UNSW-NB15. UNSW-NB15 is developed according to the modern flow of network traffic with 49 features including 9 types of network attacks. To analyze the traffic pattern for the intrusion detection system(IDS), we have used multiple classifiers to test the accuracy. From the dataset UNSWNB15, we have used medium and strong correlated features. All the results from different classifiers are compared. Prominent results are achieved by ensemble bagged tree which classifies normal and individual attacks with an accuracy of 79%.**

Keywords: Network Intrusion Detection System(NIDS), Attacks, Machine Learning

1. INTRODUCTION

Intrusion detection becomes a highly important aspect of any network infrastructure due to rapid growth in the number of users. The number of users is increased due to technology like IoT and other network-based application. Performance of the intrusion detection system remains a concern in order to detect, identify and track the attacker footprints. To detect footprints with high accuracy researchers are continuously working and contributing to the network society[1]. IDS are divided into two types; misuse and anomaly. Misuse IDS are based on the databases of attacks signature whereas Anomaly-based IDS can classify the normal and abnormal activity by monitoring the network traffic. The advantage of anomaly-based IDS is it can detect more type of unknown attacks [2]. Misuse IDS have a very low rate of attack detection whereas anomaly IDS are difficult and time-consuming in order to analyze a large amount of data[2][3]. Machine learning techniques can be useful to detect and prevent network infrastructure from an intruder.

Muhammad Shahzad Haroon is with the department of computer science, Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi Pakistan. E-mail: Shahzad.haroon@szabist.edu.pk

Dr Syed Sajjad Hussain is with the department of computer science, Hamdard University, Karachi, Pakistan. Email: sshussainr@gmail.com

In this paper, we will perform exhaustive testing on UNSW-NB15 dataset using various machine learning algorithms with their multiple variants.

2. LITERATURE REVIEW

The machine learning techniques are used extensively these days to analyzed data pattern. The machine learning technique further divided into three categories; supervised, unsupervised and reinforcement [2]. The supervised machine learning algorithm used to classify different type of traffic which are already labelled. Whereas, Unsupervised machine learning algorithm is used to group the data which are similar in type with respect to their properties. Reinforcement learning algorithm in which agent, learns by interacting with its environment. The agent receives rewards by performing correctly and penalties for performing incorrectly.

Researcher are using machine learning technique as a tool to analyze the network traffic pattern[11]. To distinguish the normal and abnormal traffic in a large network, accuracy is the key point to defeat the attacker[4]. Since real-time network traffic cannot be used due to the privacy concerns of users, Multiple testbed environments have been created to design a dataset in which these machine learning algorithms can be tested[5][6].

Many researchers have exhaustively used benchmark dataset KDDCUP99. KDDCUP99 included 22 attack types in training dataset and testing data contained 15 attack types [4][5]. The major reason for making KDDCUP99 as a benchmarking is public availability of the dataset. The research community also highlighted the disadvantages or disappearance in KDDCUP99 [6] [7] [8].

NSLKDD[17], which is basically an upgrade version of KDDCUP99 was introduced as keeping three goals to improve. First is to remove all the duplication of records. Second, selecting a variety of records from different parts of the original KDDCUP99 dataset to achieve reliable results from classifier systems. Third, eliminating the unbalancing problem among the number of records[2]. NSLKDD still lack those scenarios which contain low footprints in modern attacks[3].

The unavailability of a comprehensive network-

based dataset which fulfils all the parameters of modern traffic leads to another dataset UNSWNB15. Researchers have used UNSWNB15 dataset to detect attacks in multiple ways. Multiple works have been reported since the dataset publicly available[2][9]. Classification of the incoming network traffic into DoS traffic or normal traffic [4]. Comparison between NSL-KDD and UNSWNB15 dataset by using different machine learning algorithm[5]. Detection of known and unknown web attacks using logit boost algorithm[10]. Random Forest performs better while including all the 42 features which in result identify traffic in normal or abnormal[7]. Feature selection has been predicted by using different techniques[8].

Authors of UNSWNB15 dataset, correlate all the 42 features and ranked them into three categories; highest correlated, middle correlated, and lowest correlated[13]. To the best of my knowledge, classification by taken highest and middle features have not reported yet.

In this paper, we have used the highest and middle features of UNSWNB15 dataset to classify normal and individual attacks. We have used multiple machine learning algorithms with their variants such as Decision Tree, Support Vector Machine(SVM), K-Nearest-Neighbor(KNN) and Ensemble to performed exhaustive testing.

The rest of this paper is organized as follows: In Section 3, we have discussed UNSW-N15 Dataset. In Section 4, Present the three types of correlated Features. In Section 5, machine learning classification algorithms are discussed. In Section 6, Performance parameters are noted which includes training and testing accuracy, true positive ratio, prediction speed and training time. In Section 7. Concludes our work and finally in Section 8, future work.

3. DATASET

The motivation of using UNSW-NB15 dataset, it contains newly attacks which appear in modern traffic with many records. The UNSWNB15 dataset has 49 features with a total of records is two million and 540,044. The dataset is further broken into the training and testing file with 175,341 and 82332 records respectively. The dataset contains nine types of attacks with the normal traffic pattern. The 49 features of the dataset are groups into five categories namely Flow features, Basic features, Content features, Time features and Additional generated features.

4. FEATURE SELECTION

The UNSWNB15 contains 49 features which contain 9 different attack types with normal traffic. We are classifying all the attacks instead of classifying traffic with normal or abnormal type. Correlation of all the features classified into three ranks lowest refer table 1, middle refer table 2 and highest refer table 3 [12]. In the paper, we have used middle and highest ranks features to classify all the attacks. After the selection of middle and highest correlated features, the number of features is reduced to 24.

#	Name	Description
1	srcip	Source IP address
2	sport	Source port number
3	dstip	Destination IP address
7	dur	Record total duration
12	sloss	Source packets retransmitted or dropped
13	dloss	Destination packets retransmitted or dropped
16	dload	Destination bits per second
17	spkts	Source to destination packet count
27	sjit	Source jitter (mSec)
28	djit	Destination jitter (mSec)
29	stime	record start time
34	synack	The time between the SYN and the SYN_ACK packets of the TCP
35	ackdat	The time between the SYN_ACK and the ACK packets
36	is_sm_ips_ports	If source equals to destination IP addresses and port numbers, this variable takes value 1 else 0
37	ct_state_ttl	No. of flows of source/destination time to live
38	ct_flw_http_mthd	No. of flows that has methods such as Get and Post in http service
39	is_ftp_login	If the ftp session is accessed by user and password then 1 else 0

Table 1- Lowest Ranked Feature

#	Name	Description
4	dsport	Destination port number
5	proto	Transaction protocol
6	state	The state and its dependent protocol, e.g. ACC, CLO, else (-)
10	sttl	Source to destination time to live
11	dttl	Destination to source time to live

15	sload	Source bits per second
18	Dpkts	Destination to source packet count
19	Swin	Source TCP window advertisement
20	dwin	Destination TCP window advertisement
21	stcpb	Source TCP sequence number
22	dtcpb	Destination TCP sequence number
23	smeansz	Mean of the flow packet size transmitted by the src
24	dmeansz	Mean of the flow packet size transmitted by the dst
25	trans_dept	the depth into the connection of http request/response transaction
26	res_bdy_len	The content size of the data transferred from the server's http service
31	sintpkt	Source inter-packet arrival time (mSec)
32	dintpkt	Destination inter-packet arrival time (mSec)
40	ct_ftp_cmd	No of flows that has a command in ftp session.

Table 2-Middle Ranked Feature

#	Name	Description
8	sbytes	Source to destination bytes
9	dbytes	Destination to source bytes
30	ltime	record last tim
33	tcprtt	The sum of 'synack' and 'ackdat' of the TCP
41	ct_srv_src	No. of connections that contain the same service source based
42	ct_srv_dst	No. of connections that contain the same service destination base

Table 3-Highest Ranked Feature

5. CLASSIFICATION TECHNIQUES

Decision Tree

A decision tree is a tree where each root is connected to its branch via the link. The root is represented as the strongest feature. The impact of the features is higher at the root and gradually decreases as the tree grows through its branches. Links are the rules on which root attribute is connected to its branches attributes[14]. To decide the root attribute information gained is calculated using.

$$entropy(D) = -\sum_{i=1}^{|C|} Pr(c_i)\, log_2 \quad Pr(c_i)$$

(1)

$$\sum_{i=1}^{|C|} Pr(c_i) = 1,$$

(2)

Where Pr(ci) is the probability of class ci in a data set

$$entropy_{B_j}(D) = \sum_{i=1}^{v} \frac{|D_i|}{|D|} \times entropy(D_i)$$

(3)

$$gain(D, B_j) = entropy(D) - entropy_{B_j}(D)$$

(4)

Support Vector Machine

Support vector machine algorithm use kernel to take data as an input and transform it into the required method[14]. The SVM kernel is a set of mathematical functions. The different variant of SVM uses different types of the kernel. The variant of SVM is, for example, Linear SVM, Non-Linear SVM, polynomial SVM, radial basis function and sigmoid. The inner product between two-point is return by the kernel in each feature space. SVM has an ability to work with large dataset with little computational cost

K-Nearest-Neighbor

The K-NN is a supervised machine learning algorithm that can be used for classification and regression problems.

A case is classified by a majority vote of its neighbours, with the case being assigned to the class most common amongst its K nearest neighbours measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbour. Below equation number 5 and 6 are valid for continues variables otherwise hamming distance can be used[15].

$$d = \sqrt{\sum_{a=i}^{k} (x_a - y_a)^2}$$

(5)

$$d = \sum_{a=1}^{k} |x_a - y_a|$$

(6)

Ensemble

The ensemble is basically a technique in which multiple machine learning algorithm are combined to produce better results[16]. Ensemble method is further divided into sequential and parallel ensemble methods. Both methods are used to exploit the base learners.

6. METHODOLOGY

From the dataset UNSWNB15, we have selected middle and highest correlated features as mentioned in section 4. Dataset was initially labelled as a binary class, 0 for normal traffic and 1 for abnormal. Out of ten, nine classes belonged to different attack types which were labelled as class 1. For individual detection of each attack type, each attack is labelled with a different class. After analyzing each class, we are left with six classes as four classes were imbalanced refer table 4.

Type	Class #	Training records	Testing Records
Normal	1	55999	36999
DoS	2	12264	4089
Exploits	3	33393	11132
Fuzzers	4	18184	6062
Generic	5	40000	40000
Reconnaissance	6	10491	3496

Table 4- Dataset Distribution

Table 5 shows the removed classes from the dataset due to class imbalanced problem.

Type	Class #	Training records	Testing Records
Analysis	7	2000	677
backdoor	8	1743	583
Shellcode	9	1133	378
Worm	10	130	44

Table 5- Class Imbalanced

Multiple machine learning algorisms are applied to remaining of six classes using Matlab.

7. RESULT

Confusion matrix for dominate classifiers is noted as follow. Table 6 is the confusion matrix of Fine KNN. After analyzing the confusion matrix, we can easily identify the problem of low accuracy in class 2 and class 3 as they were not trained as other classes were trained. In Fine KNN class 1, 4 and 5 are classified more appropriately then the rest of them. Whereas, class 6 classified as moderately.

Confusion Matrix		True Class					
KNN Fine		1	2	3	4	5	6
	6		8	4	3		85
	5					99	1
	4	3	4	2	90	1	
	3		18	71	6		5
	2		57	20	12		11
	1	99			1		

Table 6-Confusion Matrix Fine KNN

Confusion matrix for weighted KNN is noted in table 7. The poorly classified class is 2 which eventually degrade the overall performance of the KNN weighted test accuracy, Rest of the classes are classified more appropriately apart from class 6 which performed moderately.

Confusion Matrix		True Class					
KNN Weighted		1	2	3	4	5	6
	6		6	11	1		82
	5			1		99	
	4		3	6	90		
	3		12	88			
	2		49	51			
	1	99			1		

Table 7-Confusion Matrix Weighted KNN

Confusion matrix of Ensemble bagged tree is noted in table 8. All the classes are classified more accurately except Class 2 which degrading the overall performance of ensemble bagged tree.

Confusion Matrix		True Class					
Ensemble Bagged Tree		1	2	3	4	5	6
	6			17			83
	5				1	99	
	4		1	9	89		1
	3		1	99			
	2		20	79			
	1	97			3		

Table 8-Confusion Matrix Ensemble Bagged Tree

Confusion matrix of all the dominate classifiers points out the class 2 as a culprit. The reason behind the bad performance of class 2 is the low number of records in both train and test dataset. More the number of records better the algorithm will train.

Performance parameters are also noted in table 9 of various machine learning algorithms, which includes True positive rate, Training percentage, Testing percentage, prediction speed and training time.

With respect to the testing accuracy, Ensemble bagged tree performed well among the rest of the algorithms with the accuracy of 79%. Few variants of KNN also performed notably near to the ensemble bagged tree which includes Weighted KNN and Fine KNN with the testing accuracy of 71.54% and 71.15% respectively.

The true positive rate of Weighted KNN is better among others while the nearest are Ensemble bagged tree and Fine KNN.

Fine Gaussian SVM has performed in the highest prediction speed but low in other performance parameters. Fine gaussian SVM providing a tradeoff between the prediction speed and others parameter.

Similarly, another tradeoff is provided by Tree variants in which training time is less but the other performance parameters are affecting its overall performance.

In comparison with Prediction speed and Test accuracy, we have observed another tradeoff where SVM Gaussian providing better prediction speed as compare to others but lack in the test accuracy. Similarly, KNN variants also providing better prediction speed as compare to Ensemble bagged tree but they all lack in test accuracy.

In our approach, if test accuracy is crucial parameter then Ensemble bagged tree is clearly a choice. Whereas, Prediction speed of Ensemble bagged tree is among the lowest. The training time of ensemble is also very competitive among others.

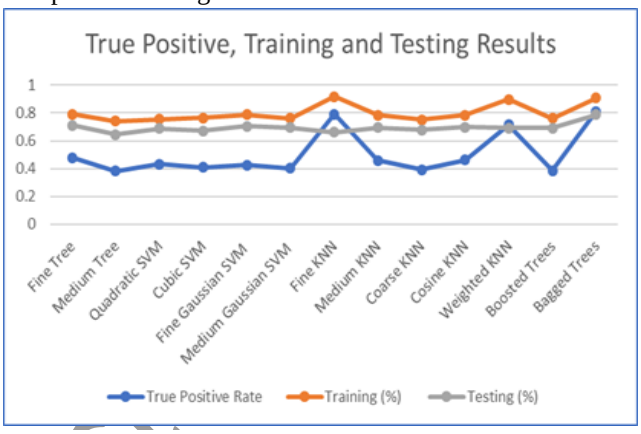

Figure 1-Perforamce Parameters

Comparison of performance parameter displayed in figure 1. Three dominate candidates are Ensemble tree, Fine KNN and Weighted KNN. Performance measurements are very close but in overall a test accuracy is the most required parameter in which Ensemble bagged tree has the accuracy of 79%. Results of ensemble bagged tree are better as ensemble combine different prediction trees to produce one strong which is more powerful than a single tree.

8. Conclusion

To cater to the problem of modern security attacks we have used multiple machine learning algorithms. We have classified 6 types of traffic classes out of 10 due to class imbalanced issue in dataset UNSW-NB15. We have classified all the individual attacks instead of classifying normal and abnormal traffic by using middle and highest correlated features. Ensemble bagged trees providing better test accuracy in order to classify all the classes but not performed well in other performance parameters.

Similarly, few classifiers like Fine KNN performs better in the true positive rate but degrade in other performance parameters.

9. FUTURE WORK

To encounter the security problems in recent day attacks, the result of our approach needs further improvements. Our future work includes better feature selection, better train algorithm and selection of better machine learning algorithm in order to achieve optimal results.

Predicted Class		True Positive Rate	Training (%)	Testing (%)	Prediction Speed(K-Obs/sec)	Training Time (hours)
Tree	Fine Tree	0.48	0.79	0.71	1.50	0.12
	Medium Tree	0.38	0.74	0.64	1.10	0.12
SVM	Quadratic SVM	0.43	0.75	0.69	0.52	5.78
	Cubic SVM	0.41	0.77	0.67	0.24	3.11
	Fine Gaussian SVM	0.43	0.79	0.70	0.11	1.37
	Medium Gaussian SVM	0.40	0.76	0.69	0.23	2.09
KNN	Fine KNN	0.79	0.92	0.66	0.45	0.75
	Medium KNN	0.46	0.79	0.70	0.18	3.47
	Coarse KNN	0.39	0.75	0.68	0.18	4.02
	Cosine KNN	0.46	0.79	0.70	0.19	4.25
	Weighted KNN	0.72	0.90	0.69	0.18	4.79
Ensemble	Boosted Trees	0.39	0.76	0.69	34.00	4.81
	Bagged Trees	0.81	0.91	0.79	40.00	3.68

Table 9-Performance Parameters

References

[1] L. Cui, S. Yang, F. Chen, Z. Ming, N. Lu, and J. Qin, "A survey on application of machine learning for Internet of Things," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 8, pp. 1399–1417, Aug. 2018.

[2] Q. Liu, P. Li, W. Zhao, W. Cai, S. Yu, and V. C. M. Leung, "A survey on security threats and defensive techniques of machine learning: A data driven view," *IEEE Access*, vol. 6, pp. 12103–12117, 2018.

[3] C. Applications, "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling ☆," vol. 87, no. November 2016, pp. 185–192, 2017.

[4] P. Ravi Kiran Varma, V. Valli Kumari, and S. Srinivas Kumar, "A Survey of Feature Selection Techniques in Intrusion Detection System: A Soft Computing Perspective," 2018, pp. 785–793.

[5] UCI Machine Learning Repository, "KDD Cup 1999 Data," 2015. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data. [Accessed: 04-Dec-2018].

[6] N. Moustafa, J. Slay, and I. Technology, "Intrusion Detection systems," 2015.

[7] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," no. Cisda, pp. 1–6, 2009.

[8] K. Afdel, "DoS Detection Method based on Artificial Neural Networks," no. May, 2017.

[9] M. AL-Hawawreh, N. Moustafa, and E. Sitnikova, "Identification of malicious activities in industrial internet of things based on deep learning models," *J. Inf. Secur. Appl.*, vol. 41, pp. 1–11, 2018.

[10] M. H. Kamarudin, C. Maple, T. Watson, and N. S. Safa, "A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks," *IEEE Access*, vol. 5, pp. 26190–26200, 2017.

[11] M. Belouch, S. El Hadaj, and M. Idlianmiad, "Performance evaluation of intrusion detection based on machine learning using apache spark," *Procedia Comput. Sci.*, vol. 127, pp. 1–6, 2018.

[12] T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," *IEEE Int. Symp. Ind. Electron.*, pp. 1881–1886, 2017.

[13] N. Moustafa and J. Slay, "The evaluation of Network Anomaly Detection Systems : Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set The evaluation of Network Anomaly Detection Systems : Statistical analysis of," vol. 3555, no. January, pp. 0–14, 2016.

[14] A. F. M. Agarap, "A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data," in *Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018*, 2018, pp. 26–30.

[15] M.-Y. Su, "Using clustering to improve the KNN-based classifiers for online anomaly network traffic identification," *J. Netw. Comput. Appl.*, vol. 34, no. 2, pp. 722–730, Mar. 2011.

[16] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 32, no. 1, p. 61, Jun. 2004.

[17] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," no. Cisda, pp. 1–6, 2009.