IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Anomaly Detection in Networks Using Different Machine Learning Algorithms

[1]Vinod Kumar, [2]Vinay Choudhary, [3]Vinay Kumar, [4]Vivek Sahrawat

[1]Associate Professor, Department of Computer Engineering, Delhi Technological University, Delhi
[2]Undergraduate Student, Department of Computer Engineering, Delhi Technological University, Delhi
[3]Undergraduate Student, Department of Computer Engineering, Delhi Technological University, Delhi
[4]Undergraduate Student, Department of Computer Engineering, Delhi Technological University, Delhi

Abstract: Nowadays, everyone is using internet services to communicate. Millions of folks and, many organizations communicate with one another using internet services every day. In conjunction with these developments, the number of attacks as well as attackers over the internet is increasing exponentially day by day. Though there are some techniques, based on signature, which are used to forestall these attacks, but they are unsuccessful against zero-day (unrecognized earlier) attacks. The technique based on detection is an alternative technique to forestall network's attacks, and also has the power to detect zero-day attacks. A secure machine-controlled anomaly detection system is more practical procedure to help in network analysis. An anomaly detection systems (ADS) examine the network flow and focuses on detecting uncommon network behaviour, and classify them into attacks. In this paper, we planned to implement an anomaly detection method using Naïve Bayes, Decision Tree (ID3), ensemble learning and Multi-Layer Perceptron (MLP), and compare their efficiencies. A subset of attributes (significant) is chosen from the primitive set of attributes using random forest regressor technique, and then, the chosen set of significant features are used to train different types of classifiers.

Index Terms **- Classification, Anomaly Detection System (ADS), CICIDS2017, Machine learning.**

I. INTRODUCTION

Established institutions and ample folks communicate with one another over the internet every day. Within past twenty years, where the number of individuals using the net services has been hyperbolic. Today this range has exceeded four billion and increasing exponentially. Online services and networks are often used for communicating and exchanging sensitive information. Attackers often peep into these networks and get access to sensitive and confidential information which suggests the requirement for some mechanism to detect such actions and then alert the authorities about the same. Various machine learning approaches such as Decision Tree, MLP etc. can be used to improve management, analytics, and security of the systems.**[3]**

The most common and intuitive technique to forestall these attacks is the signature-based technique. The signature-based techniques use database to store information about attacks. They create databases to detect attacks. This technique is quite a sort of booming however, the databases have to be perpetually updated and process new attack information. Moreover, even if the databases are up-to-date, they're open to the zero-day attacks (unrecognized earlier). They cannot forestall the zero-day attacks as they aren't present in the information database.

In this paper, we aim to implement task for detecting anomaly in networks by making use of effective machine learning techniques. Also, for evaluating most useful features or attributes and consider them for further study, random forest regressor algorithm is used during pre-processing.

The rest of the research paper is structured into various sections. The related work that has been done in the field of anomaly detection is highlighted in section 2. Section 3 presents the background details regarding the dataset and algorithms used for our study. Section 4 consists of the details of the pre-processing stage and the experimental setup. Section 5 then compare the results of various algorithms and analyse them. Section 6 concludes our work and Section 7 presents the scope for future work.

II. Related Work

For anomaly detection in networks by machine learning techniques, we require a large quantity of harmful and harmless network traffic data for training and testing steps, However, real-time network traffic can't be used in public due to security problems**[2]** and thus to develop efficient intrusion detection systems, several datasets have been simulated in almost real environments. DARPA 98, KDD 99, CAIDA (Centre of Applied Internet Data Analysis), NSL-KDD, ISXC 2012 and, CICIDS 2017**[1]** are various datasets for network anomaly detection.

For anomaly detection, these datasets are used as a benchmark in performance evaluation. There is no single "Fit-For-All" algorithm for detecting various attacks differing enormously based on their source, target, and mode of application. The anomaly detection systems are often classified based on the type of environment in which they are deployed as "Host Anomaly Detection Systems" and "Network Anomaly Detection Systems" **[4]**. We aim to detect attacks by finding anomalies in the "Network Anomaly Detection Systems" in this paper.

Another classification of ADS can be on the basis of techniques employed in implementing them. There are mainly two categories: "Signature-based ADS" and "Anomaly based ADS". Signature-based ADS were prevalent in the real-world applications since the last decade. But due to their inability to detect zero-day attacks, they are replaced by anomaly-based ADS. Signature-based ADS were based on concepts like string matching, regular expressions, and finite state machines **[5]**.

Today most of the systems employ more efficient and effective anomaly-based systems using the machine learning algorithms which can generalize the type of attacks from the real-time network traffic. Machine learning algorithms generally consists of two phases – the "learning phase" and the "inference phase".**[6]** The learning phase enables us to build the network traffic profile, which generalizes the benign and normal traffic passing through the network. During the inference phase, any abnormal behaviour is treated as an anomaly and indicates the presence of attack and the system administrators are alerted about the same.

III. Background

3.1 Dataset

We have used the CICIDS2017**[1]**- processed data file (CSV files) to be used in the implementation section. As a result of the comparison process, the foremost necessary factors during this preference are the fact that the dataset is up-to-date and offers a wider protocol and attack pool. Additionally, working on this dataset might have large enough impact to the literature because work done in past using this dataset is few.

DoS (Denial of Service) HULK, DDoS (Distributed DoS), FTP-Patator, DoS Slowloris, SSH-Patator, DoS SlowHTTPTest,DoS Goldeneye, Botnet, PortScan, Web Attack, Infiltration, Heartbleed are various types of attacks present in the datasets available.

At the University of New Brunswick, the Canadian Institute for Cyber security created **CICIDS2017**. This dataset consists of a 5-day (3rd July- 7th July 2017) data stream on a network created by computers having up-to-date operating systems like Windows7 / 8.1 / 10, Ubuntu 12/16, Kali and Mac.

Recording Day	**pcap file size (GB)**	**Recording Hours**	**CSV file size (MB)**	**Attack**
Mon	10.0	Full Day	257	-
Tues	10.0	Full Day	166	SSH-Patator, FTP-Patator
Wed	12.0	Full Day	272	DoS Hulk, GoldenEye, Slowloris, Slowhttptest, Heartbleed
Thurs	7.7	Morning	88	Web Attacks
		Afternoon	103	Infiltration
Fri	8.2	Morning	72	Bot
		Afternoon	190	PortScan, DDos

Table 1. (Details About CICIDS2017 Dataset)

The 2017 CICIDS dataset has the subsequent benefits over other datasets:

1) The obtained information is the real-world data; obtained from a test bed consisting of real computers.

2) Data streams are collected from computer systems with the up-to-date operating system. There's operating system diversity (Windows, Linux and Mac) between each victim computer.

3) Datasets are labelled. To apply the machine learning techniques, the feature extraction, that is an important step, was applied and 85 features (see Appendix A for the feature list) were obtained.

4) Both raw information (pcap files - captured network packets files) and processed data (CSV files- comma-separated data files) are out there to work on.

On the other hand, CICIDS2017 dataset has subsequent disadvantages:

1) Raw data files and processed data files are terribly giant (97.9 GB and 1147.3 MB respectively).

Unlike the KDD99 and NSL-KDD datasets, CICIDS2017 doesn't have separate files dedicated to training and testing. These sections ought to be created by users. The way to do that is handled within the creation of training and testing section.

3.2. Decision Trees

Decision tree is one among the favoured classifiers used in machine learning techniques. During this approach, the rules used are fairly easy and apprehensive. Every decision tree consists of root nodes, internal nodes and leaf nodes connected with branches. Within every node, there are statements responsible for decision taking. In keeping with the result of this decision, in the step, one among the different branches (the number of branches could be more than two in some sub-algorithms) is chosen. In this way, the algorithm makes progression downwards in the tree. This action continues until it reaches the leaf node (result).

Decision trees technique applies the divide-and-conquer strategy. It makes terribly giant and insignificant data small and group them into a significant one. It's therefore widely used for the classification of large and complex data. Unfortunately, these complex trees result in over-fitting, and stop obtaining fair results. Another disadvantage is that a small misclassification of the data at the beginning of decomposition will result in completely different branches and misleading results. To address this, usually, the data to be used must go through pre-processing.

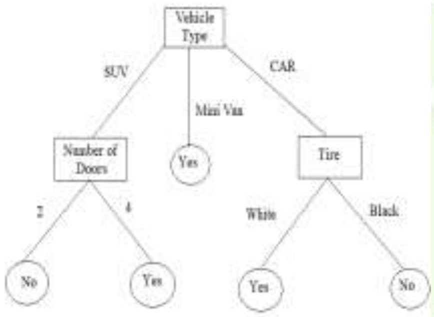

Fig. 1. (Decision Tree Example)

3.3. Naïve Bayes

Naïve Bayes classifier is a probability based classifier which is based on the Bayes' theorem. It strongly assumes that there is no dependence between attributes. Bayes' theorem is expressed by the subsequent equation:

$$P\left(\frac{X}{Y}\right)=\frac{P\left(\frac{Y}{X}\right)}{P(X)P(Y)}$$

$P\left(\frac{X}{Y}\right)$: Chances of happening of event X after event Y.
$P\left(\frac{Y}{X}\right)$: Chances of happening of event Y after event X.
P(Y) and P(X): Prior chances of event Y and event X.

Naïve Bayes structure is similar to Directed Acyclic Graph (DAG) in which each node is a state and link between nodes represent dependency of each node on one another. Naïve Bayes is a probabilistic network containing unobserved state i.e. parent node and observed states i.e. multiple children nodes. In this statistical network, it's assumed that the child nodes are independent of each other. Though this hypothesis is the basis of Naïve Bayes' theory, the assumption that sub-nodes are independent of one another is usually not correct. This can be the underlying reason for the Naïve Bayes methodology to attain lower accuracy compared to other machine learning techniques.

Despite this disadvantage, Naïve Bayes is one of the foremost popular technique because the training period is very short, and the computational cost is extremely low.

3.4. Multi-Layer Perceptron (MLP)

"Multi-Layer Perceptron (MLP) is a genre of artificial neural networks". "Artificial Neural Networks" (ANN) is a machine learning technique that takes inspiration from the working of human brain. The intention of this methodology is to imitate the properties of the human brain, such as learning, high cognitive processes, and deriving new information. While the human brain is made up of interconnected cells known as neurons, artificial neural networks are made up of interconnected hierarchical artificial cells.

MLP consists of three stages. These stages are: one is Input layer, next one is hidden layer, and last but not the least output layer. The input layer is that stage of the MLP that's responsible for receiving data. No information processing is performed on this layer. Only the received information is transmitted to subsequent layer, the hidden layer. Every neuron in this step bonds with all the neurons within the hidden layer.

In the output layer, that is the last layer, every cell is tied to all the cells within the hidden layer, and therefore, the result of the processed data from the hidden layer is served at this stage.

The advantages MLP provides are as follows:

1) It is good at managing complicated problems.

2) It can work with missing data.

3) It generalizes after the learning process. Thus, it serves a wider space than the other machine learning algorithms.

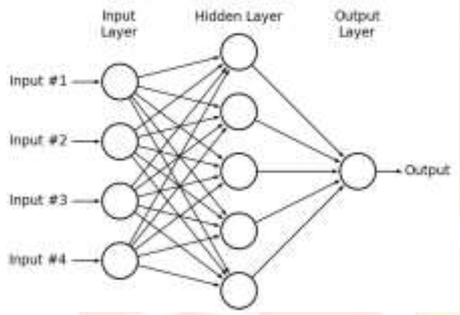

Fig. 2. (Simple Multi-Layer Perceptron Model)

3.5. Ensemble Learning

The goal of **ensemble methods** is to mix the prediction results of many base classifiers / regressors designed and trained totally different learning algorithm to improve generalizability / strength over one classifier / estimator. The concept behind the ensemble classifier is to mix totally different machine learning algorithms and based on majority vote also called hard voting or the average prediction chances also called soft voting to make predictions. Such a classifier is often useful for a group of equally well performing classifier / regressor to balance out their individual weaknesses. In majority vote, the output for a selected sample is that category label that represents the bulk of the category labels classified by each regressor / classifier.

e.g. if different model's predictions for a given sample are

model 1 -> yes
model 2 -> yes
model 3 -> no

then using ensemble learning model will predict "yes" based on the majority class prediction.

IV. EXPERIMENTS

In this section, details of data pre-processing along with the specifications of the experimental setup are provided.

4.1 Data Pre-Processing

Various techniques for data pre-processing are performed. Dataset is pre-processed to get rid of mistakes and defects present. It's necessary to pre-process the dataset before training and testing. For this, some defects within the CICIDS2017 dataset are processed and corrected. The dataset contains 3119345 records. Once the records are examined, it is seen that around 2.5 lacks records are incomplete or incorrect. The primary step is to delete the surplus records.

The next error within the dataset is within the columns that form up the features. The dataset contains 86 columns that define the properties e.g. Flow ID, Source IP, Source Port, etc. The Fwd Header Length feature (which tells about the forward direction data flow for total bytes used) was repeated two times one in 41st column and second time in 62nd column. This error is removed by deleting the repeating columns. Another editing that needs to be done in the dataset is to convert the features value including string and categorical values (Flow ID, Source IP, Timestamp, External IP, Destination IP) into numerical data to be easily understandable by different machine learning algorithms. This is often be done using LabelEncoder() function provided by Sklearn package. In this way, all string values that don't seem to be understandable by machine learning operations can get integer values and become more efficient for further processing.

The "Label" tagged column is also modified, if there is an attack then it's value will be 1 and if there is no attack then it's value will be 0. Finally, some structural changes even be done to the dataset, including:

1) In the Label feature, the character "–" (Unicode Decimal Code &#8211) used to identify the web attack sub types (Brute Force and SQL Injection) should get replaced with the character "-" (Unicode Decimal Code &#45), as utf-8, the default Pandas library, won't recognize it. Otherwise, Pandas library won't recognize this character and can fail.

2) "Flow Bytes/s", "Flow Packets/s" features have valued to like "Infinity" and "NaN" with numerical values, that is changed to -1 and 0 respectively to make them efficient for various machine learning algorithms.

4.2 Experimental Setup

The technical specifications of the computer used in the implementation:

CPU:	"Intel(R) Core (TM) i5-7200U CPU @ 2.70GHz"
RAM:	8 GB
OS:	"Windows 10 Pro 64-bit"
GPU:	"Nvidia 940 mx 8gb ddr4"

Jupyter-Notebook 5.7.8 on Anaconda 2019.03 with Python 3.7.3 is used for implementing data pre-processing steps and various machine learning algorithms. Various python libraries are used: pandas, matplotlib and sklearn.

V. RESULTS AND DISCUSSION

In this section, results of study carried out are presented. The performance evaluation criterion is based on the values of the F1-measure. Histograms are created to visualize the comparison between different features and different machine learning algorithms.

We used random forest regressor for necessary feature selection. First, we used simple Principal Component Analysis (PCA) for feature selection, but PCA is not efficient when there is an enormous amount of feature in the dataset. Below figure shows the importance of various features.

FEATURES	IMPORTANCE WEIGHT (*10$^{(-5)}$)
Standard Backward Packet Length	24656
Flow Byte/s	17876
Forward Packets Total Length	11721
Standard Forward Packet Length	6389
Standard IAT	1017
Minimum IAT	658
Total IAT	499
Flow Duration	413
Maximum Flow IAT	354
Maximum Backward Packet Length	339
Mean IAT	329
Backward Packets Total Length	135
Minimum Forward Packet Length	63
Flow Packet/s	58
Mean Backward Packet Length	53
Mean Forward Packet Length	51
Total Backward Packet/s	23
Total Forward Packet/s	13
Maximum Forward Packet Length	12
Minimum Backward Packet Length	8

Table 2. (Important features with importance weights computed using random forest)

Table 2 exhibits the importance weight of top 20 important features computed using random forest after data pre-processing.

We use four machine learning algorithms, ID3 (Decision Tree), MLP, Ensemble Learning and Naïve Bayes, and compare them on the premise of various performance evaluation criterion i.e. precision, recall, accuracy and F1-Score. Table 3 exhibits the value of precision, recall, accuracy and F1-Score of various algorithms.

“ML Algorithm”	**Precision**	**Recall**	**Accuracy**	**F1-Score**
Naïve Bayes	0.82	0.82	0.8165	0.82
ID3	**0.95**	**0.95**	**0.9515**	**0.95**
Ensemble Learning	0.89	0.89	0.891	0.89
MLP	0.84	0.84	0.8387	0.84

Table 3. (Performance evaluation data for different algorithms)

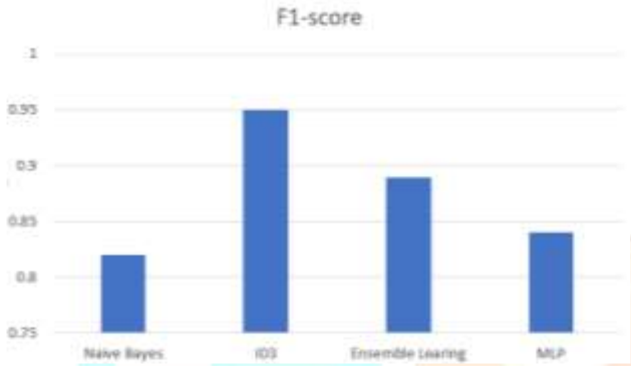

Fig. 3. (different algorithm’s F1-Score comparison)

Figure 3 exhibit the accuracy comparison and F1-Scorecomparison respectively that are obtained after applying various machine learning algorithms. It is evident from figure 3 that ID3 and Ensemble Learning perform best out of the various algorithms applied after data pre-processing and feature selection.

5.1. Why ID3 comes out with best results?

1) ID3 (Iterative Dichotomiser 3) is very intelligible, versatile, and easy to debug. It’ll work with classification and regression problems. If you are attempting to predict a categorical value or to predict a continuous value ID3 will handle both problems, but it can handle categorical data efficiently than Naïve Bayes and MLP.

2) All Decision Trees is that they solely want a part of data, and they can build a classifier directly from that data. Whereas in other algorithms, choosing features is up to you. There's no way to just toss a part of data, and it picks the most important features that it’ll used to classify.

3) ID3 is really quick regarding computation than other algorithms. ID3 does not produce classifier using all dataset. It’s an iterative method in which a rule is computed and every data record that satisfy the rule is going to be removed and additional rules are generated until all data records aren’t covered.

VI. CONCLUSION

The aim of the study is to implement task for detecting anomaly in networks by making use of effective machine learning techniques and compare their efficiencies. CICIDS2017 dataset has been used because of its wide attack diversity and up-to-datedness. Network flow is described by the features (over eighty) contained within the dataset. A subset of attributes is chosen from the primitive set of attributes using the Random Forest Regressor technique to make our mind up that which feature are used in machine learning algorithms. Finally, four machine learning algorithms i.e. Naive Bayes, ID3, MLP and Ensemble Learning, are practiced on the dataset. The resulted performance based on F1-measure are as follows (F1-measure takes a value between 0 and 1): Ensemble Learning (NB, ID3 and, MLP): 0.89, ID3: 0.95, MLP: 0.84 and Naive Bayes: 0.82.

VII. FUTURE WORK

In this paper, we aim to implement numerous machine learning algorithms for anomaly detection in networks. Further, a comparative study of various algorithm used for implementing ADS are conferred.

This work is open to future enhancements. In this study, the dataset used, consist of CSV (comma-separated values) files, was obtained from a network of computers created by the Canadian Institute for Cyber security. Unluckily, in real time systems, this technique isn't much viable. However, this drawback is resolved by adding a module that catches real network data and makes it practicable with various machine learning algorithm.

In this study, numerous machine learning strategies is applied independently of one another and experimental results were obtained. However, this technique has a weak practical applicability in real life to overcome this drawback, a multi-layered / hierarchical machine learning structure is often designed. Moreover, because of such a structure, it's attainable to save time, CPU power, and memory. For example, in a two-tiered structure, the primary layer is often created from quick and computationally low-cost algorithms such as Naïve Bayes, therefore network traffic can be observed continuously and at a minimal cost. The primary step, when detecting an anomaly, is to transmit it to an upper layer that composed of algorithms with higher performance such as ID3, Ensemble learning etc. The ultimate layer that creates up the determination mechanism takes the preventive decision to protect the network against various attacks.

REFERENCES

1. A detailed analysis of CICIDS2017 dataset for Designing Intrusion Detection Systems https://www.sciencepubco.com/index.php/ijet/article/view/22797/11274
2. R. Sommer, V. Paxson, Outside the Closed World: On Using Machine Learning For Network Intrusion Detection, in: IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 305–316. doi:10.1109/SP.2010.25.
3. Maciá-Pérez F, Mora-Gimeno FJ, Marcos-Jorquera D, Gil-Martínez-Abarca JA, Ramos-Morillo H, Lorenzo-Fonseca I., "Network intrusion detection system embedded on a smart sensor", IEEE Transactions on Industrial Electronics. 2011; 58(3):722-32.
4. Abhishek Pharate , Harsha Bhat , Vaibhav Shilimkar "Classification of Intrusion Detection System" IJCS Volume 118 – No. 7, May 2015.
5. Subaira AS, Anitha P., "Efficient classification mechanism for network intrusion detection system based on data mining techniques: a survey. In international conference on intelligent systems and control", 2014 (pp. 274-80).
6. T. Sree Kala, Dr .A. Christy "A Survey and Analysis of Machine Learning Algorithms for Intrusion Detection System" Jour of Adv Research in Dynamical & Control Systems, 04-Special Issue, June 2017.
7. "Machine Learning Approaches to Network Anomaly Detection" Tarem Ahmed, Boris Oreshkin and Mark Coates, Department of Electrical and Computer Engineering, McGill University Montreal, QC, Canada
8. M. S. Uzer, "Feature Selection Algorithms Developed by Using Artificial Intelligence And Feature Transform Methods In Pattern Recognition Applications," Ph.D Thesis, The Graduate School of Natural and Applied Science Selçuk University, 2014.
9. A. Özgür and H. Erdem, "A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015," PeerJ PrePrints, vol. 4, p. e1954v1, 2016.
10. H. Hajji, "Statistical analysis of network traffic for adaptive faults detection," IEEE Trans. Neural Networks, vol. 16, no. 5, pp. 1053–1063, Sep. 2005.
11. "Survey on Anomaly Detection using Data Mining Techniques" Shikha Agarwal, Jitendra Agarwal, Department of Computer Science and Engineering, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India.
12. J. Brutlag, "Aberrant behavior detection in time series for network monitoring," in Proc. USENIX System Admin. Conf. (LISA), New Orleans, LA, Dec. 2000.
13. "1998 DARPA Intrusion Detection Evaluation Data Set," Lincoln Laboratory, Massachusetts Institute of Technology, [Online]. Available: https://www.ll.mit.edu/rd/datasets/1998-darpa-intrusion-detection-evaluation-data-set.
14. "Intrusion Detection Evaluation Dataset (CICIDS2017)," Canadian Institute for Cybersecurity, University of New Brunswick, [Online]. Available: http://www.unb.ca/cic/datasets/ids-2017.html
15. S. Mukherjee and N. Sharma, "Intrusion detection using naive Bayes classifier with feature reduction," Procedia Technology, vol. 4, pp. 119-128, 2012.
16. S. Chebrolu, A. Abraham, and J. P. Thomas, "Feature deduction and ensemble design of intrusion detection systems," Computers & security, vol. 24, no. 4, pp. 295-307, 2005.
17. Patcha A., Park J. M., An overview of anomaly detection techniques: Existing solutions and latest technological trends; Computer Networks; 51(12); 2007; p. 3448-3470
18. Qin, M. and Hwang, K., "Frequent episode rules for internet anomaly detection," In Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications, 2004, IEEE Computer Society.
19. Mahoney, M. V. and Chan, P. K., "Learning rules for anomaly detection of hostile network traffic"' In Proceedings of the 3rd IEEE International Conference on Data Mining, 2003, IEEE Computer Society, pp. 601.