Cyber Threat Detection using Machine Learning

Simanta Rajbangshi[1], Chemkai Wangpan[2], Ayushman Chaudhury[3], Nupur Choudhury[4] and Rupesh Mandal[5]

[1]Assam Don Bosco University
simanta.rajbangshi@gmail.com[1], kaimohtaham8020@gmail.com[2], 0026rim@gmail.com[3], nupur.choudhury@dbuniversity.ac.in[4], rupesh.mandal@dbuniversity.ac.in[5]

Abstract. Millions of users have been a victim of cyberattacks and thousands of companies are affected as well. This paper proposes Machine Learning to be used as a method to improve the detection rates of cyberthreats in a network which is better than the traditional signature or anomaly-based methods. Machine Leaming can be used to detect threats and protect systems in real time thereby reducing the damage caused by attacks to a very high extent. In this paper, five Supervised Machine Learning algorithms, Random Forest, Logistic Regression, SVM, Decision Tree and Naive Bayes have been used with optimized parameters and tuning and lastly, a deep learning algorithm; Convolutional Neural Network (CNN) has been used and the performances have been compared amongst them. The algorithms performed well with Random Forest model being the highest. The results achieved proves that Machine Learning can be implemented to develop a threat detection system for a network which would be much more secure compared to the existing methods of detection and prevention.

Keywords: Intrusion Detection System, Machine Learning, Convolutional Neural Networks, Networking

1 INTRODUCTION

This generation is highly dependent on the internet to carry out various tasks which are considered essentials. With an increase in the number of users on the internet, the number of attacks on users on the internet is also increasing. Denial of Service (DoS) is the most popular attack that has been rising lately, also harmless attacks like probing attacks which include port scanning that give information about a system or network have been resulted to more dangerous attacks like Privilege escalation attacks or Remote Access attacks. From the studies, it can be said that threat detection is basically a classification where the system/model has to classify a packet as safe or harmful. To secure our data from such attacks, Intrusion Detection Systems (IDS) are created which monitors and analyzes the data flowing in a network for identification of intrusions. There are two types of IDS, Signature Based IDS or S-IDS and Anomaly Based IDS or A-IDS. S-IDS identify threats in a network from the signatures stored in their database and can detect known attacks but fails for newer attacks. While, A-IDS creates a pattern for normal network behavior and classifies unknown networks as intrusion, but contains high false positive rates. Thus, machine learning is an optimal solution. The existing Intrusion Detection Systems are struggling to increase both detection speed and accuracy simultaneously, but no practical solutions are available. Thus, to combat this problem, we are doing a research work that uses Machine Learning with a Deep Learning algorithm to detect threats in a network. For achieving accurate attack detection, using signature-based machine learning is not suitable due to the limitations of using pre-configured databases and results in the failure to detect threats that are unknown to the system. Therefore, machine learning techniques should be adopted in our system and implemented in such a way that it works accurately and also be able to predict newer methods of attacks. Moreover, existing machine learning techniques using standard algorithms [1],[2] or based on IoT [5] does have limitations. Massive amounts of data that needs be classified in such cases. Therefore, deep learning methods [3] come into play which can provide better results and has good potential. In this research work, CNN, a deep learning algorithm which is mostly used for image processing, but not been used in the field of networking systems.

The remaining paper is constructed as follows: Section 2 represents the past works conducted in the field of IDS using machine learning or deep learning methods as well as work on the NSL-KDD dataset. Section 3 contains the description of NSL-KDD dataset and its analysis. Section 4 is the detailed information about the methodology used to conduct the research work. Section 5 describes the results achieved followed by Section 6 as the conclusion of our work.

2 RELATED WORKS

Our research shows that various works have been carried out in the field of Intrusion detection systems using Machine Learning algorithms. In 2013, Omar Salima and her team, [1] carried out an analysis on three perspectives of technical challenges in IDS based on machine learning, namely feature extraction, classifier construction and sequential pattern prediction. Recently, a study where deep learning was used for an IDS by Qu et al. [2] showed an IDS model based on Deep Belief Networks and using the NSL KDD dataset. They have applied long short-term memory (LSTM) architecture to RNN and trained the model using KDD Cup'99 dataset. Another study was done by Nutan Farah Haq and team [3] where a statistical comparison was done to show classifier design. They used single, hybrid and ensemble algorithms in the research. A study by Hamid, Yasir, Muthukumarasamy [4] was done where they used different machine learning methods using Weka on the KDD CUP 99 dataset. They used 10-fold cross validation to evaluate the performance in terms of True Positive Rate, False Positive Rate, precision, accuracy, etc. It also evaluates the machine learning models provided by Weka and describes the performance measures taken into consideration. Another related work named Internet of Things: A survey on machine learning-based intrusion detection approaches [5] shows the use of both new as well as traditional machine learning algorithms for handling security issues in IoT environments. Their system was able to detect three types of IoT attacks called jam, false and reply attack. A study by Ingre and Yadav evaluated the performance of NSL-KDD dataset using ANN. They obtained high detection rates in binary as well as multi class classification [6]. A study by Punam Bedi and team [7] shows class imbalance problem handling in the KDD-99 and NSL-KDD datasets using Siamese Neural Networks. The attack classes R2L and U2R being the minority, have less detection rate as compared to DoS and Probe attack classes and hence proposed a Siam-IDS that can detect them without oversampling or random under sampling techniques. Sun et. al. [8] created a dual layer model for efficient threat detection using KDD CUP 99 dataset. They did an oversampling of the minority class using synthetic samples and used Gradient Boosting Decision Tree, KNN and Fly Optimization algorithm for separation of normal and attack classes. In 2003, Mukkamala & Sung [9] used feature selection for intrusion detection with neural networks and SVM to rank the input features according to each specific class label. In 2014, Akhilesh [10] proposed ANN based IDS with net gain ratio feature selection technique. A data mining framework was introduced by Wenke Lee [11] for the same. Chandrasekhar A.M [12] used data mining techniques like fuzzy C-means clustering, Fuzzy neural network and SVM for developing and IDS using the KDD-99 dataset. Mohammad Reza Parsaei et al. [13] used SDN to categorize network traffic by applying different types of Neural Network estimators. They used data mining techniques with various ML algorithms. Gjorgji Ilievski and Pero Latkoski [14] classified network traffic using supervised ML algorithms by using a network functions virtualization environment and decision tree performed the best. Luong-Vy Le et al. [15] have applied big data, ML, NFV, SDN and built a practical and strong configuration for predicting and management of behaviors in network traffic.

3 DATASET DESCRIPTION AND ANALYSIS

NSL KDD dataset is the corrected version of KDD CUP 99 which was first publicized by the MIT Lincoln labs at the University of California. NSL-KDD was later published by the University of New Brunswick as a revised version of KDD'99. It was proposed to remove the issues of KDD 99 dataset that contained a lot of unnecessary records. Even though the dataset is old as of now, it serves as the benchmark to compare similar IDS/IPS systems as well as Machine Learning algorithm-based system's performances. The dataset contains the record of information collected through systems like simple Intrusion detection network systems. The dataset has 43 columns, and are divided into 5 main categories: Normal traffic, Denial of Service, Probe or Scans, User to Root and Remote to Local. The following two graphs are plotted to visualize the data distribution against different protocols and attack labels.

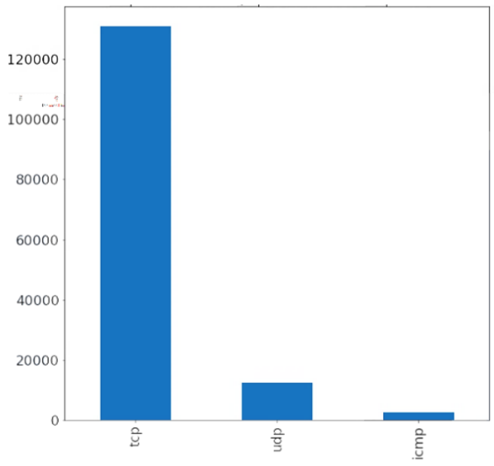

Fig. 1. Distribution of data points per protocol

Category	Name of Attack	ICMP	TCP	UDP	Data Points
DOS attacks	Back	0	956	0	45927
	Land	0	18	0	
	Neptune	0	41214	0	
	Pod	201	0	0	
	Smurf	2646	0	0	
	Teardrop	0	0	892	
Probe Attacks	Ipsweep	3117	482	0	3633
	Nmap	981	265	247	
	Portsweep	5	2926	0	
	Satan	32	2184	1417	
Privilege Escalation Attacks	Buffer Overflow	0	30	0	52
	Loadmoudle	0	9	0	
	Perl	0	3	0	
	Rootkit	0	7	3	
Access Attacks	Ftp Write	0	8	0	995
	Guess Passwd	0	53	0	
	Imap	0	11	0	
	Multihop	0	7	0	
	Phf	0	4	0	
	Spy	0	2	0	
	Warezclient	0	890	0	
	Warezmaster	0	20	0	
Normal packets		1309	53599	12434	67342

Table 1. Data points in each attack label

Statistical measurements and representation provide a better understanding of the dataset used. It can be seen that there are 22 types of attack from the figure given above. These attack categories are further divided into 4 types and one normal type. DOS attack presents 90% of attack types and 30% the dataset while other attack types compromises of only 9.25% among all attack types. At the first glance, the dataset appears unbalanced but it contains lots of information about the data packets.

In order to extract relevant feature from the dataset, we first removed any duplicate values present. We then mapped the labels as either '1' or '0' depending whether it is attack type or normal type using one-hot encoding. If features aren't selected properly, the performance of the model is highly affected. The table above shows the number of data points that are present categorically as well as in each protocol along with the attacks labels for analysis.

4 METHODOLOGY USED

We used 5 Supervised ML algorithms and one Deep Learning algorithm to compare the performances side by side. We have also used hyperparameters in each to fine-tune the algorithms for better scores compared to default parametric scores. Grid Search Cross Validation method is also used which will automatically find the best model according to the parameters given. It takes higher time but provides the best parameters and results

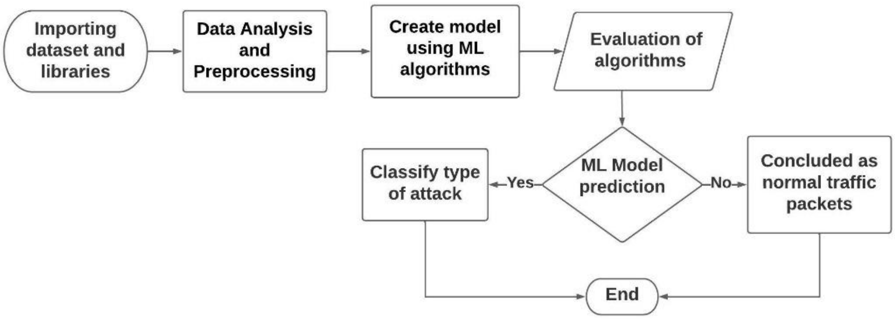
Importing dataset and libraries
Data Analysis and Preprocessing
Create model using ML algorithms
Evaluation of algorithms
ML Model prediction
Yes
Classify type of attack
No
Concluded as normal traffic packets
End

Fig. 2. Workflow diagram for the methodology used

4.1 ALGORITHMS USED

The following 6 Machine Learning Algorithms have been used to compare and contrast the performances and create our model:

1) Decision Tree: The maximum depth of the tree has been specified along with the number of samples that are required to split. The parameter to specify the number of cross validation iterations have been also used and specifying the number of cpus to use. Below are the hyperparameters used:
max_depth: 5-500, min_samples_split: 5-500, Crossvalidation iterations = 3, n_jobs=-1 (using all CPUs)

2) Random Forest: In this model, the maximum depth is specified as well as the number of estimators which are the decision trees.

Hyperparameters used: Max_depth: 5-1000, N_estimators: 5-500, min_samples_split=5-500, crossvalidation iterations =3, n_jobs= -1 (using all CPUs)

3) Support Vector Machine (SVM):. In this algorithm, SGD Classifier has been used with the loss function as ‘hinge’ which gives the linear SVM.
Hyperparameters used: Penalty: l1, l2, Loss= hinge, Cross validation iterations: 5, n_jobs= -1 (using all CPUs)

4) Logistic Regression: We again used the SGD classifier using the loss function as 'log' which represents the logistic regression function. Likewise, we have used GridSearchCV to tune the parameters and find the best model.
Hyperparameters used: Penalty: l1, l2, Loss=log, Cross Validation Iterations: 5, n_jobs= -1

5) Naïve Bayes: In this algorithm, the number of iterations has been specified to make sure the accuracy can be determined properly. Also, the number of jobs to run in parallel has been specified as -1 which indicates it can use all CPUs to perform the operation.
Hyperparameters used: Var_smoothing: 10**x for x in range (-9,3), Cross validation iterations: 5, n_jobs= -1

6) Convolutional Neural Network: For this algorithm, we have used relu as activation function for the first 3 layers and sigmoid for the output layer. We have given the learning rate as 0.9. The optimizer RMSprop has been used to create the model. Convolution Neural Network achieved very low accuracy rate and f1-score considering to the fact that CNN are best used for image processing.

4.2 EVALUATION METRICS

Accuracy is the most important evaluation parameter for an intrusion detection system that is used to measure the performance of the machine learning models. In addition to the accuracy, we considered the true positive and false positive rate.

We have used the following metrics to evaluate our models performance:

$$\text{Accuracy} = \frac{\text{TP+TN}}{TP+FP+FN+TN} \tag{1}$$

$$\text{True Positive Rate (TPR)} = \frac{True\ Positive}{True\ Positive\ \ +False\ Negative} \tag{2}$$

$$\text{False Positive Rate (FPR)} = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{3}$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{4}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive\ + False\ Negative} \tag{5}$$

$$\text{F1 Score} = 2 * \frac{Precison*Recall}{Precision+Recall} \tag{6}$$

5 RESULTS

The tested algorithms performed quite well and in comparison, to each other. The models have performed really well achieving good scores with random forest topping with precision score of 97.89, recall 97.91 and f1-score of 96.05. But as the dataset used is smaller in size and contains very less data points for many attack classes, it can be difficult for the models to work in real time. Moreover, newer datasets can be developed which contain the latest information about each attack type, and would increase the detection rate for newer attacks to a very high extent due to the learning capabilities of machine learning algorithms. Also, the deep learning CNN algorithm is capable of performing better with correct parameters and better optimized datasets for neural networks.

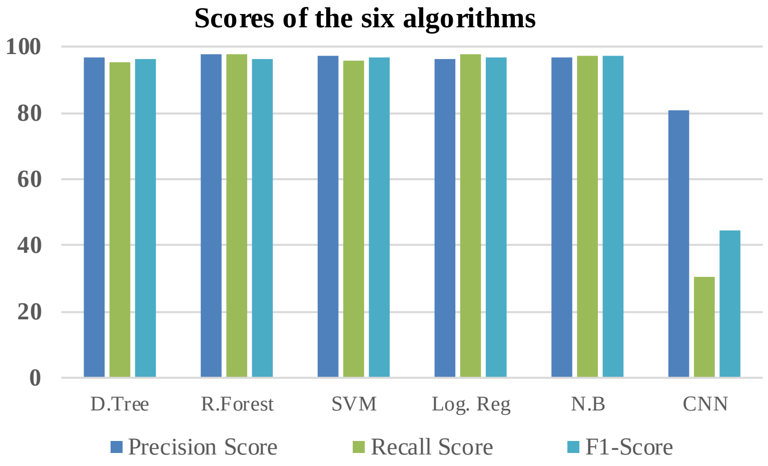

Fig. 3. Bar graph to show the scores of our tested algorithms

6 CONCLUSION

As we already know, existing threat detection systems like IDS or IPS fails to detect modern threats, and rely on signatures available in their databases to classify them as threats, meanwhile failing to detect the zero-day threats. We demonstrated how Machine learning is useful to effectively predict and detect threats in the network with great accuracy. Also, the model's performance can be improved by using the GPU libraries for Deep Learning instead of the CPU which results in 10x improvement in training times. This proves that machine learning algorithms can be further used to detect threats in real time as well as prevent zero-day attacks to a great extent if it is optimized correctly with proper parameters and configuration.

References

1. Omar, Salima & Jebur, Hamid & Benqdara, Salima. (2013). An Adaptive Intrusion Detection Model based on Machine Learning Techniques. International Journal of Computer Applications. 70. 10.5120/11971-6640.
2. Qu F, Zhang J, Shao Z, et al. An Intrusion Detection Model Based on Deep Belief Network[C]// Vi International Conference. 2017:97-101.
3. Nutan Farah Haq, Abdur Rahman Onik, Md. Avishek Khan Hridoy, Musharrat Rafni, Faisal Muhammad Shah and Dewan Md. Farid, "Application of Machine Learning Approaches in Intrusion Detection System: A Survey" International Journal of Advanced Research in Artificial Intelligence (IJARAI), 4(3), 2015.
4. Hamid, Yasir & Muthukumarasamy, Sugumaran & Journaux, Ludovic. (2016). Machine Learning Techniques for Intrusion Detection: A Comparative Analysis. 1-6. 10.1145/2980258.2980378.
5. Costa, Kelton & Papa, João & Lisboa, Celso & Munoz, Roberto & Albuquerque, Victor. (2019). Internet of Things: A Survey on Machine Learning-based Intrusion Detection Approaches. Computer Networks. 151. 10.1016/j.comnet.2019.01.023.
6. Li W, Yi P, Wu Y, et al. A New Intrusion Detection System Based on KNN Classification Algorithm in Wireless Sensor Network. Journal of Electrical and Computer Engineering, 2014, 2014(5):1-8
7. Punam Bedi, Neha Gupta, Vinita Jindal, Siam-IDS: Handling class imbalance problem in Intrusion Detection Systems using Siamese Neural Network Volume 171,2020, Pages 780-789
8. Sun, Chong, Kun Lv, Changzhen Hu, and Hui Xie. (2018) "A double-layer detection and classification approach for network attacks" 27th International Conference on Computer Communication and Networks (ICCCN). Hangzhou, China: IEEE: 1-8
9. S. Mukkamala and A. H. Sung, "Feature selection for intrusion detection with neural networks and support vector machines, Journal of the Transportation Research Board, Vol. 1822, 2003, pp. 33-39.
10. KumarShrivas, Akhilesh & Dewangan, Amit. (2014). An Ensemble Model for Classification of Attacks with Feature Selection based on KDD99 and NSL-KDD Data Set. International Journal of Computer Applications. 99. 8-13. 10.5120/17447-5392.
11. Wenke Lee, S. J. Stolfo and K. W. Mok, "A data mining framework for building intrusion detection models," Proceedings of the 1999 IEEE Symposium on Security and Privacy (Cat. No.99CB36344), 1999, pp. 120-132, doi: 10.1109/SECPRI.1999.766909.
12. A.M, Chandrashekhar & Raghuveer, (2013). Fortification of Hybrid Intrusion Detection System Using Variants of Neural Networks and Support Vector Machines. International Journal of Network Security & Its Applications.
13. M.R. Parsaei, M. J. Sobouti, S. Raouf Khayami, R. Javidan: Network Traffic Classification Using Machine Learning Techniques over Software Defined Networks, International Journal of Advanced Computer Science and Applications, Vol. 8, No. 7, July 2017, pp. 220 - 225.
14. Gjorgji Ilievski , Pero Latkoski: SERBIAN JOURNAL OF ELECTRICAL ENGINEERING Vol. 18, No. 2, June 2021, 237-254
15. L.- V. Le, D. Sinh, B.- S. P. Lin, L.- P. Tung: Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management, Proceedings of the 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, Canada, June 2018, pp. 168-176.