IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 1, Ver. I (Jan - Feb 2019), PP 32-37
www.iosrjournals.org

Intrusion Detection System using Apache Spark Analytic System

Asst. Prof. Dr. Sefer KURNAZ, Ali Jameel Abdulrazzaq Al-Rawi
Head of Department Computer Engineering
Corresponding Author: Asst. Prof. Dr. Sefer KURNAZ

Abstract: - *In this study, an Intrusion Detection System (IDS) is proposed based on the use of machine learning and distributed computing. The proposed system uses classification techniques that are implemented in the built-in machine learning library in Apache Spark distributed computing framework. As the use of distributed computing allows the proposed method to provide rapid predictions for the packets flowing in the network, two classifiers are cascaded in order to combine their decisions for more accurate decisions. The Multi-Layer Perceptron (MLP) classifier is used as a binary classifier, where the output of this classifier only indicates whether the packet is a normal or attack packet. Packets predicted to be normal by this classifier are allowed through the network. However, packets predicted as attacks are classified again using the random forest classifier, which provides the state of the packet and the type of the attack as its output. If the packet is classified as a normal packet, it is also allowed to the network, otherwise it is filtered out. The results show that he proposed methodology has been able to improve the performance of IDS to 99.12%, which outperforms the state-of-the-art systems in the literature.*

Keywords: *- Distributed Computing, Machine Learning, Anomaly Detection, Network Security.*

Date of Submission: 04-01-2019 Date of acceptance: 19-01-2019

I. Introduction

The rapid growth of the number of services being provided online emerges the net of protecting the servers that provide these services in order to maintain the quality of these services. However, the intrusion techniques being used to perform attacks against these servers are getting more complex and become more similar to normal traffic, which increases the difficulty of detecting such intrusion. Thus, different techniques are proposed, to distinguish the normal traffic from that coming from intrusions, using machine learning techniques. Machine learning techniques have the ability of extracting domain knowledge from different inputs, where the same machine learning technique extract different knowledge depending on the environment that it is applied in. Classification if one of the widely used machine learning fields, where the relations between the values that characterize a tuple and the label given to that label are extracted as the domain knowledge. This knowledge can furthermore be used with new tuples in order to predict a label for that tuple, depending on the values in the tuple and the knowledge extracted from the training dataset. Thus, classification is used with Intrusion Detection Systems (IDS) to predict the incoming packets whether to be of normal traffic or intrusion attempt. These predictions are used to control access to the network, where only packets predicted to be of normal traffic are granted access to the network.

Moreover, the time required to process each packet has a direct impact on the quality of the services being provided by the network, where longer predictions time result in delays for the packets to go through. Such delay is magnified in larger networks, where the number of packets passing through the network is extremely high and faster decisions are required to avoid affecting the quality of the services provided by the network. The use of a single server to classify the incoming traffic also imposes the risk of losing the IDS when the server is down. Thus, it is important to distribute the classification process on different servers, so that, the resources of these services are combined to provide faster performance, in order to maintain the quality of the services provided on the network, and eliminate the risk of losing the IDS when one of the servers goes down.

II. Literature Review

In order to use machine learning for IDS, it is important to use a labeled dataset the contains packets' information and the type of traffic that they belong to, so that, the classifier can extract the relations between this information and the labels given to the packet. For this reason, different datasets are provided in the literature to train a classifier for an IDS. One of the earliest datasets is the KDD-CUP'99 dataset [1], which has 41 features that characterize each packet into a normal or attack packet. Moreover, each attack packet is categorized into one of four kinds of attack, which are the Denial of Service DoS), probing attack, Remote to Local (R2L) and User to Root (U2R). This dataset suffers from two main drawbacks, where there too many duplicates in the dataset, around 75% in the testing dataset and 78% in training dataset, as well as the low difficulty level of classifying this dataset. These drawbacks have enabled most of the classifiers to provide accurate predictions, not according to their performance, but according to the simplicity of the knowledge extraction especially with the duplicate tuples [2].

A newer version of the KDD CUP'99 network traffic dataset, known as NSL-KDD'99, is proposed to overcome the issues exist in the original dataset. The newer version includes the same attack types of the original dataset but has reduced the number of supplicate tuples and reduces the predictions simplicity level. However, the setup of the setup of the experiment is questionable and concerns about using the synthetically generated dataset to train a classifier for intrusion detection have also raised about this dataset [3].

Another recent and popular network traffic dataset is the UNSW-NB15 dataset [4], which includes information about 2,540,047 network packets, characterized by 45 features. Each packet is provided with two labels, one label indicated whether the packet is a normal or an intrusion packet, while the other label provides the type of attack that the packet comes from, if it is a part of an intrusion attempt. There are nine different types of attacks in the dataset, which are Analysis, Dos, Backdoor, Worm, Generic, Fuzzers, Shellcode, Reconnaissance and Exploit attacks. The dataset consists of 2,219,410 tuples that represent normal packets and 320,637 attack packets. As this dataset is the best among the other network traffic dataset, it is widely used in intrusion detection systems that use machine learning.

The IDS proposed in [5] uses a two-stage decision tree to predict the state of the packet, normal or intrusion packet, and the type of the intrusion, in case the packet is an intrusion packet. The first stage uses binary classification to predict a normal or intrusion class for each packet, while the other stage classifies the packet predicted as intrusions into one of the nine types of attacks in the dataset. Different classifiers are used for both stages, such as the Decision Tree, Artificial Neural network, Naïve Bayes, Random Tree and ReTree classifiers. The highest accuracy achieved in this study with the UNSW-NB15 dataset is 87.80% for the binary classification and 79.20% for the attack type classification. Both accuracies are achieved using the RepTree classifier, while the remaining classifiers have achieved lower accuracies.

Another IDS is proposed in [6], where the employment of Random Forest and Multilayer Perceptron are evaluated for this purpose. The proposed IDS uses only binary classification to provide decisions whether to grand packets access to the network, or deny it. The highest accuracy achieved in this study, using the UNDW-NB15 dataset is 95.50%, which is achieved by the Random Forest classifier, while the Multilayer Perceptron has achieved only 83.50%. Moreover, a deep neural network classifier is used in [7] for the same purpose, and has achieved a higher prediction accuracy of 98.99%, which illustrates the significantly better performance of the deep neural networks in classifying data.

III. Proposed Method

Apache Spark analytic system [8] has been proposed in 2012 to replace Hadoop's MapReduce computations distribution framework, according the limitation imposed by storing the data on the disk, in order to be processed by Hadoop's MapReduce. Apache Spark uses Resilient Distributed Data (RDD) stored in the memory of the computers, for faster processing, while the MapReduce framework uses Hadoop Distributed File System. Moreover, the built-in machine library of spark is optimized for distributed computing, which makes the maximum use of the distribution capabilities of the Sparks framework. Thus, most of the recent applications are being implemented in Spark framework, instead of MapReduce.

As larger networks have an enormous number of packets going through the network to the servers that are providing the intended services, it is important to handle these packets without affecting the quality of the services provided by the network, by consuming longer time to provide predictions for each packet. Another important benefit behind using the Spark framework is the ability of using the existing servers on the network to combine small portions of the resources available on these servers, instead of using a standalone server, which results in a more cost-effective IDS. Thus, in this study, the different classifiers in Spark's machine learning library, such as the Random Forest, Support Vector Machine and Feed-Forward Neural Network are going to be tested in order to evaluate the performance of each of these classifiers. The performance is measured by means of time required to provide a prediction and the accuracy of the predictions provided by each classifier.

The classifiers with the highest performance are selected for further improvements, using techniques such as features selection and cascade classification, in order to produce an intrusion detection method that outperforms the existing techniques in both accuracy of the predictions provided to the firewall and the time taken to provide these predictions, using the hierarchy shown in Figure 1. The main benefit of using the Spark framework is going to be the ability of using more complex techniques without consuming long time that may affect the quality of the services provided by the network, according to the fast performance of the Spark framework that distributes data and processing on the clusters' computers. Using such hierarchy with Spark framework can achieve the reliability and speed for an IDS, i.e., a packet detected by the binary classifier to be normal is instantly allowed access to the network, while if the binary classifier predicted the packet to be of an attack, this packet's information is forwarded to the multi-class classifier in order to make a decision. A packet is denied access if both classifiers predicted that it is an attack packet, while the packet is allowed access if any of the classifiers predict that it is a normal packet.

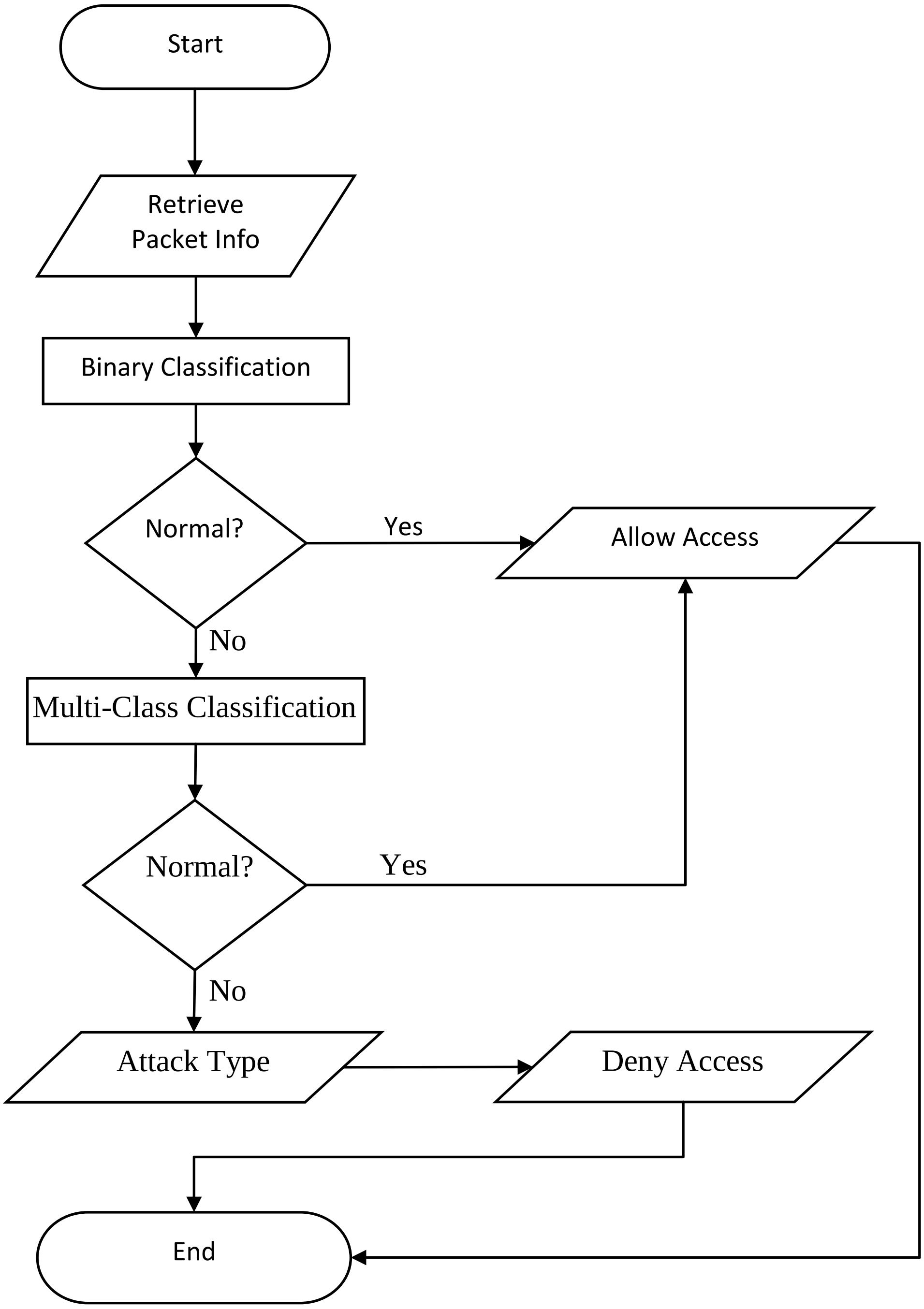

Figure 1: Block diagram of the proposed intrusion detection system.

IV. Experimental Results

In order to evaluate the performance of the proposed method, two of the widely used classifiers, which are implemented in the Spark's machine learning library (MlLib), are used to classify the UNSW-NB15 dataset. This dataset consists of 2,540,047 instances, each holds information of a single network traffic and described using 47 attributes. Each instance is labeled with two labels, one indicated the state of the packet, to be normal or attack, and the other presents the type of the attach that the packet is coming from, in case it comes from an attaching computer. The socket information of the sender and receiver of the packet present in the attributes, which produce biased classification, in which the classifier emphasis on the source and destination socket information. Such emphasis allows future attacks from other hosts to go through the intrusion detection system, while normal traffic from sources recognized by the classifier as sources of attacks is filtered out. According to the types of attacks being executed, such as Distributed Denial of Service (DDOS) attacks, make use of victim hosts to execute the attacks, and according to the possibility of using dynamic IP addresses subnetting, such biased training can dramatically reduce the quality of the services provided by the network being protected. Thus, these attributes are removed from the dataset.

The binary labels provided to the instances in the dataset are used to train the Random Forest classifier, of the Spark's MlLib. The dataset is divided into five bins, each is used once for evaluation and four times for training, in order to produce more accurate performance measures. The accuracy of the predictions provided by the Random Forest classifier is 99.56%, where each prediction consumed an average of 0.083uS. These predictions are summarized in the confusion matrix shown in Table 2.

Table 1: Confusion matrix for the prediction of the Random Forest classifier.

		Predicted	
		Normal	**Attack**
Actual	**Normal**	2213792	4972
	Attack	6185	315098

Next, the Multi-Layer Perceptron (MLP) classifier, from the Spark's MlLib, is also evaluated using the same dataset, which is also divided into the same bins as in the Random Forest. The results of the predictions provided by this classifier are illustrated in the confusion matrix shown in Table 3, which show that the predictions accuracy of this classifier is 99.37%. The average time consumed by the classifier in order to provide a prediction for a single packet is 0.006uS.

Table 2: Confusion matrix for the prediction of the MLP classifier.

		Predicted	
		Normal	**Attack**
Actual	**Normal**	2209127	9637
	Attack	6266	315017

Despite the slightly more accurate predictions provided by the Random Forest classifier, the time required by the MLP classifier, to provide a prediction per a packet, is 7.2% of the average time consumed by the Random Forest classifier. This difference in execution time illustrate that the MLP classifier is capable of providing 13.83 predictions per each prediction provided by the Random Forest classifier. As the intrusion detection system relies on these predictions to allow or deny packets from accessing the network, the MLP classifier is selected for the first stage of the proposed method.

The model shown in Figure 1 is then implemented, where the MLP classifier is used for the binary classification, according to the faster performance, while the Random Forest classifier is used for the multi-class classification. This model makes use of the rapid performance of the MLP performance to allow normal traffic access the network, while access of suspicious packets is decided by the Random Forest classifier, which has more accurate prediction. Eventually, the proposed model outputs a single label per each packet, which is one of the ten labels exist in the dataset. One of these labels is normal, while the remaining labels represent nine different attacks. Packets predicted by the MLP **OR** the Random Forest classifiers to be from normal traffic are granted access to the network, while packets predicted by the MLP **AND** the Random Forest classifiers to be from attacking hosts are denied access to the network. This combination can maintain the quality of the services provided on the network, while securing the network by filtering out packets from attacking hosts.

Table 4 shows the confusion matrix for the predictions provided by the proposed model, which shows that this model has been able to achieve an overall accuracy of 99.12% for all classes.

Table 3: Confusion matrix of the predictions provided by the proposed method.

		Predicted									
		Analysis	**Backdoors**	**DoS**	**Exploits**	**Fuzzers**	**Generic**	**Normal**	**Reconnaissance**	**Shellcode**	**Worms**
Actual	**Analysis**	600	0	609	479	228	18	743	0	0	0
	Backdoors	0	354	619	1103	229	2	19	3	0	0
	DoS	1	2	15669	370	235	26	39	9	2	0
	Exploits	0	4	4190	37159	470	65	2597	34	4	2
	Fuzzers	0	2	631	1658	21838	7	97	11	2	0
	Generic	15	5	645	1207	5	210540	3059	3	2	0
	Normal	0	0	34	77	17	593	2217931	109	3	0
	Reconnaissance	1	0	655	1337	3	6	173	11812	0	0
	Shellcode	0	0	0	0	0	0	0	0	1511	0
	Worms	0	0	2	11	0	0	0	0	0	161

The time required to make a proper decision for a packet depends on the characteristics of that packet, where packets predicted to be normal directly by the MLP classifier require only 0.006*u*S, while normal packets that the MLP model is unable to detect require a total of 0.089*u*S. Moreover, comparing the results of the proposed method to the results of the method proposed by Mustapha Belouch, et al. [5], which also uses multi-stage method for intrusion detection, show that the proposed method has significantly better performance, where this method has only been able to achieve an accuracy of 87.80%. The labels provided by the multi-class classifier in this method do not include the normal label, i.e., the access decisions are granted based on the predictions of the binary classifier only, while the duty of the multi-class classifier is only to predict the type of the attack. Moreover, the proposed method has been able to achieve better accuracy than the model implamented by Malek Al-Zewairi, et al. [7], which uses a deep feed-forward artificial neural network to provide binary predicitons.

V. Conclusion

The rapid growth of services provided over the internet imposes the need to protect these networks from external intrusions. As these intrusions are becoming more sophisticated, the use of traditional protection techniques is no longer applicable. Thus, machine learning techniques are used to implement intrusion detection systems that use training data to extract the relations between each packet information and the state of that packet, normal or attack. Moreover, in larger networks, the number of packets is extremely high and providing a prediction for each packet is a time-consuming process that may degrade the quality of the services provided on the network. Thus, it is important to use a distributed system that distributes data and computation on different computers, so that, faster and more reliable predictions are provided to the firewall in order to protect the network without affecting the quality of the services provided on it.

In this study, the classifiers in the Apache Spark analytic system's machine library are going to be evaluated by measuring the accuracy of the predictions each classifier provides and the time required to provide the prediction. Classifiers with higher performance measures are going to be selected for further improvements in order to propose an intrusion detection system that has the ability to protect networks without degrading the quality of the services provides on these networks. The proposed hierarchy allows faster decisions for normal packets, to maintain the quality of the provided services, with higher accuracy, where packets predicted to be of attack traffic are double checked by the multi-class classifier. These measures are going to be compared to other studies in order to make sure that the predictions provided by these classifiers are of the same quality of those provided by other libraries. Such intrusion detection system is expected to provide better protection to the network using the same servers on the network to reduce the cost of such system and increase the efficiency of the power consumption, by avoiding the use of new servers.

The results show that the combination of the MLP and Random Forest classifiers has been able to improve the predictions that are used to make decision, whether to allow packets through the network, or deny their access. This combination makes use of the rapid predictions provided by the MLP classifier, and the better accuracy of the Random Forest classifier. Using these classifiers in two stages, where the MLP classifier provides binary predictions and the Random Forest classifier provide multi-class predictions, decisions are made by combining these prediction. Packets predicted to be of normal state, by the MLP classifier, are granted access to the network, directly, while those predicted by the MLP to be of attack state are forwarded to the Random Forest classifier. This classifier predicts one of ten available labels in the dataset, one of them is the normal. If the predicted label is normal, the packet is allowed through the network, otherwise, the access is denied and the attack type is predicted.

In future work, the use of cloud servers is evaluated to create a Spark cluster, instead of the existing on-premises server. Such system can also be used to protect other cloud servers and may be more cost-effective than the use of the servers in the network, to protect an on-premises network. However, it is important to measure the network latency, which may affect the quality of the services provided on the network as the information of each packet must be sent to the cloud cluster to provide a prediction and then retrieve that prediction through the network before granting access to that packet.

References

[1]. K. Cup, "Dataset," *available at the following website http://kdd. ics. uci. edu/databases/kddcup99/kddcup99. html,* vol. 72, 1999.

[2]. G. Nápoles, I. Grau, R. Falcon, R. Bello, and K. Vanhoof, "A granular intrusion detection system using rough cognitive networks," in *Recent Advances in Computational Intelligence in Defense and Security*, ed: Springer, 2016, pp. 169-191.

[3]. J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Transactions on Information and System Security (TISSEC),* vol. 3, pp. 262-294, 2000.

[4]. *The UNSW-NB15 Dataset.* Available: https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/

[5]. M. Belouch, S. El Hadaj, and M. Idhammad, "A Two-Stage Classifier Approach using RepTree Algorithm for Network Intrusion Detection," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS,* vol. 8, pp. 389-394, 2017.

[6]. R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," in *Data and Software Engineering (ICoDSE), 2017 International Conference on*, 2017, pp. 1-6.

[7]. M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental Evaluation of a Multi-layer Feed-Forward Artificial Neural Network Classifier for Network Intrusion Detection System," in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, 2017, pp. 167-172.

[8]. A. Spark, "Apache Spark: Lightning-fast cluster computing," *URL http://spark. apache. org,* 2016.

Asst. Prof. Dr. Sefer KURNAZ. " Intrusion Detection System using Apache Spark Analytic System" IOSR Journal of Computer Engineering (IOSR-JCE) 21.1 (2019): 32-37.