Recent Trends in Intensive Computing
M. Rajesh et al. (Eds.)

doi:10.3233/APC210243

An Intrusion Detection System for Network Security Using Recurrent Neural Network

Kishor P. Jadhav [a,1] and Dr Mohit Gangwar [b]

[a]*Ph.D. Research Scholar, Department of Computer Science and Engineering, Bhabha University, Bhopal, Madhya Pradesh, India*
[b]*Dean-Engineering, Bhabha University, Bhopal, Madhya Pradesh, India*

Abstract. To maintain the security of vulnerable network is the most essential thing in network system; for network protection or to eliminate unauthorized access of internal as well as external connections, various architectures have been suggested. Various existing approaches has developed different approaches to detect suspicious attacks on victimized machines; nevertheless, an external user develops malicious behaviour and gains unauthorized access to victim machines via such a behaviour framework, referred to as malicious activity or Intruder. A variety of supervised machine algorithms and soft computing algorithms have been developed to distinguish events in real-time as well as synthetic network log data. On the benchmark data set, the NLSKDD most commonly used data set to identify the Intruder. In this paper, we suggest using machine learning algorithms to identify intruders. A signature detection and anomaly detection are two related techniques that have been suggested. In the experimental study, the Recurrent Neural Network (RNN) algorithm is demonstrated with different data sets, and the system's output is demonstrated in a real-time network context.

Keywords. Recurrent Neural Network, KDDCUP99, Intrusion Detection System, Network security.

1. Introduction

The IDS is responsible for detecting a connection form of attack, such as a fragment of unknown attack, a DoS attack, a U2R attack, or an R2L attack. It then deploys a series of such components one by one in a sequential fashion. This accomplishes two objectives. For starters, each sub-phase can only train a limited number of characteristics that detect a specific form of attack. Second, the sub-size unit is still small enough to be helpful. A common disadvantage, similar to our system, is that it increases the amount of time it takes for modules to communicate. However, in our system, this can be easily prevented by making each sub-phase independent of the other layers. As a result, specific characteristics can be observed in more than one sub-phase. If an offense is committed without a centralized decision-maker, any thread will block it, depending on the channel's security policy. Numerous sub phases mainly function as filters blocking suspicious associations as long as they are formed during a specific layer, allowing for a quick response to the intrusion while also reducing analysis time in successive phases. It should be noted that in different sub-phases that rely on sight-trained attacks, completely different responses are often initiated.

[1] Kishor P. Jadhav,Ph.D. Research Scholar, Department of Computer Science and Engineering, Bhabha University, Bhopal, Email. kishor268@gmail.com.

At the first layer and in subsequent stages, the amount of system-analysed auditing information decreases further as more and more attacks are detected and blocked.

In the worst case, if no attacks are detected prior to the last sub-phase, all staggered sub-phases in phase 2 have the same load. However, as attacks are detected and blocked in any subsequent method, the average load is expected to be significantly lower. On the other hand, when the sub-phases are arranged in parallel rather than in a series, in a sequence configuration on a subsystem, the load is equal to the worst case. The initial step can be repeated in the sequential configuration to perform load balancing to improve performance.

2. Literature Survey

This article uses the ANN (Artificial Neural) of an Operating System Sensor to monitor malicious activities in Android and ios devices, based on the Flow anomaly system [1], based on the flow anomaly Detection Platform for Android mobile devices. The detection rate of this approach is 85 percent and 81% accuracy, respectively. Impersonation is considered in terms of CPU, space and better view, which helps to characterize a small, scalable and effective IDS after an Integration node to combat public attacks by various services. By using powerful data mining algorithms, the data sources are analysed. Improving the accuracy and classification rate requires the future scope.

PRADEEP and Dr. Yogesh Kumar [2] Effectual Secured Approach for the Internet of Things with Fog Computing and Mobile Cloud Architecture Using Ifogsim, this work cloud computing performance is assessed Simulation model world using iFogSim, where artifacts and Cloud services provide a greater degree of consistency and Precise.

Javier A. et al. proposed in [3] information security boosting using malware detection in a network environment. The platform designed would be an efficient algorithm for malware detectors for Ghana limited application security due to extensive Framework. In the final research, the participants are already confident and pleased with their reliability and functionality. The research revealed this device met that experiment's goal. "High Quality" analyzed the processes and solution to the proposed method. The development of the malware detection system for Asia Technology Security to maintain its position was successful.

Bholanath Mukhopadhyay et al. [4], cloud-based task scheduling and Protection using SSL for IaaS Application, implemented a new approach wherein we built both protection and authorization access policies. We also implemented the functionality of an Endpoint Protection choice search. In our configuration, numerous profiles can be built, one with its own different access policy, for various network applications. For illustration, for dynamic access point connections, and internet connectivity authentication policy can be developed. Using our unorthodox technique, it is possible to quickly classify the user, customer location, existing network situation at the time of connection, and server status.

Algorithm Design

Training Process

Input. Training dataset TrainData[], Various activation functions[], Threshold Th

Output. Extracted Features Feature_set[] for completed trained module.

Step 1. Set input block of data d[], activation function, epoch size,

Step 2 . Features.pkl ← ExtractFeatures(d[])

Step 3 . Feature_set[] ← optimized(Features.pkl)

Step 4 . Return Feature_set[]

Algorithm for system testing

Input. Training dataset TestDBLits [], Train dataset TrainDBLits[] and Threshold Th.

Output. Resulset <class_name, Similarity_Weight> all set which weight is greater than Th.

Step 1. For each testing records as given below equation; it works in convolutional layer for both training as well as testing

$$\text{testFeature(k)} = \sum_{m=1}^{n} (.\ \text{featureSet[A[i] ... . A[n]} \leftarrow \text{TestDBLits})$$

Step 2. Create feature vector from testFeature(m) using below function.

$$\text{Extracted\_FeatureSet\_x [t.....n]} = \sum_{x=1}^{n}(t) \leftarrow \text{testFeature (k)}$$

Extracted_FeatureSet_x[t] is the outcome of each pooling layer that is extracted from each convolutional layer and forward to net convolutional layer? This layer holds the extracted feature of each instance for testing dataset.

Step 3. For each train instances as using below function,

$$\text{trainFeature(l)} = \sum_{m=1}^{n} (.\ \text{featureSet[A[i] ... . A[n]} \leftarrow \text{TrainDBList})$$

Step 4. Generate new feature vector from trainFeature(m) using below function

$$\text{.Extracted\_FeatureSet\_Y[t.....n]} = \sum_{x=1}^{n}(t) \leftarrow \text{TrainFeature (l)}$$

Extracted_FeatureSet_Y[t] is the outcome of each pooling layer that is extracted from each convolutional layer and forward to net convolutional layer? This layer holds the extracted feature of each instance for training dataset.

Step 5. Now evaluate each test records with entire training dataset, in dense layer

$$\text{weight} = \text{calcSim}\ (\text{FeatureSetx} \,||\, \sum_{i=1}^{n} \text{FeatureSety}[y])$$

Step 6. Return Weight

3. Proposed System

Machine learning methods were used to identify and avoid intrusions in the current research methods. The runtime packets data block will conduct training, including packet selection for remote data monitoring. The role collection for a particular packet operation will then be submitted. Send it forward as a group if all is well. Misconduct samples will be examined for feature selection for different attributes in order to identify individual attacks. Figure 1 illustrates the system's entire execution using specified algorithms. To produce train modules and conduct research, various machine learning techniques were employed. The proposed network intrusion detection mechanism aims to increase the detection accuracy, cut the number of false positives, and minimise the amount of time to wait for a detected intrusion. This means that there are two distinct phases to the proposed system. during system preparation, NSLKDD data will be used; afterwards, it will be used to conduct system tests.

The proposed system would be arrayed with comprehensive support. Individuals with these same features would generate two or more models known as an ensemble model. This ensemble model has taken in the different classifier classifications from many sources and produced a single result. We've developed our classifier definition of numbers based on this information. The first step is that the programme obtains data from different sources, whether online or offline. Once all the data has been entered, classification algorithms are in place, other techniques for data mining will be employed.

System initially collects the input packet from various sources like KDD CUP, NSL KDD, ISCX and real time network packets. The entire execution holds three different phases which are listed as below.

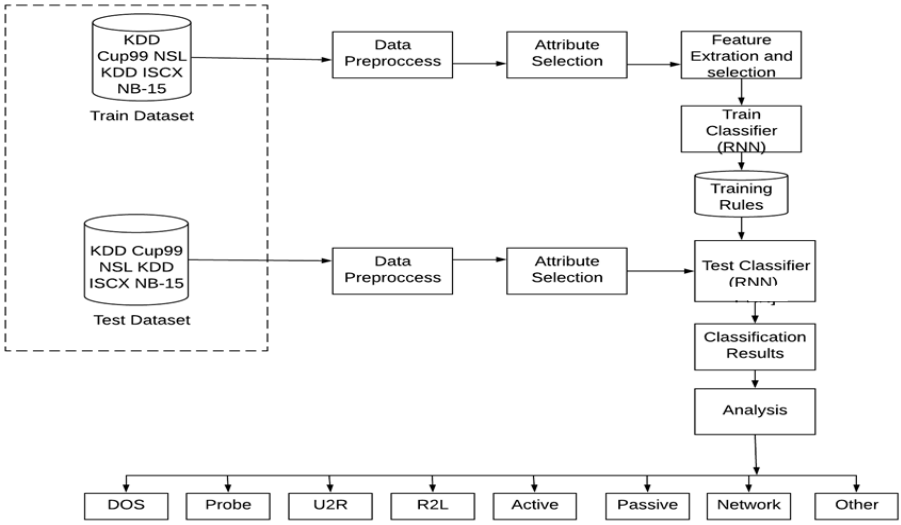

Figure 1. Proposed system architecture

4. Results and Discussion

We measure the performance evaluation of the system after it has been successfully implemented. The outcome of system collected on real time as well as synthetic traffic data and validates it with machine learning algorithm. The outcome of system has been shown in Figure 2 with multiple attack detection both environments. Figure 3 shows depict various methods, such as the RNN algorithm, were used to identify and predict the classification accuracy of the proposed system.

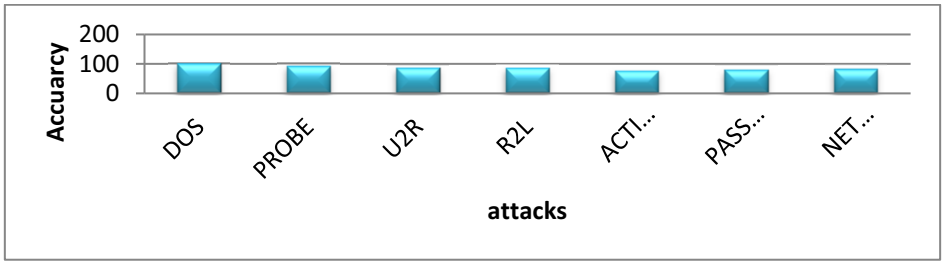

Figure 2. Detection accuracy with various attacks using RNN classification

According to the results of the second experiment, RNN with sigmoid has a higher classification accuracy than the other two activation functions, ReLU and TanH (see Figure 3). Based on the results of the above experiment, we may infer that the proposed framework improves the accuracy of trust computation in the IoT in-service environment. The entire study is driven by a collection of simulation environmental conditions and a mix of machine learning techniques. With regards to machine learning algorithms, a variety of computation specifications have been used clusters distinction and id.mi.com.

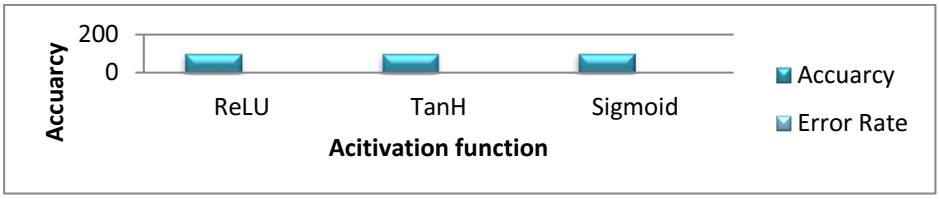

Figure 3. Experimental analysis of RNN with three activation function

5. Conclusion

In this research we proposed an efficient IDS scheme, this research proposes an RNN-IDS approach focused on deep learning. We used the numerous real time networks as well as some synthetic dataset to evaluate anomaly detection and classification accuracy. We also use deep learning to apply IDS in the cloud environment in the future. In addition, we examine and compare different deep learning approaches, such as. During the data search, the software basically functions as an RNN classification and soft computing algorithms to evaluate the unknown type of connection and attacks. To improved classification and high-class identification are possible important to the powerful rule structure. Several studies have been used for experimental investigation for evaluate the algorithm's effectiveness using a variety of methods, and we came to the conclusion that we were getting satisfactory results.

References

[1]. Panagiotis I. Radogloa-Grammatikis; Panagiotis G. Sarigannidis, "Flow anamoly based Intrusion Detection System for Android Mobile Devices", 2017 6th International Conference on MOCAST, May 4-6, 2017, Kazani, Greece.

[2]. PRADEEP, S.; SHARMA, Dr Yogesh Kumar. Effectual Secured approach for Internet of Things with Fog Computing and Mobile Cloud Architecture Using IFogSim. WE C-2019-London, UK, DOI, 2019, 978-988.

[3]. Jaevier A. Villanueva, Luisito L. Lacatan, Albert A. Vinluan, Information Technology Security Infrastructure Malware Detector System, International Journal of Advanced Trends in Computer Science and Engineering, pp. 1583-1587 ,Volume 9, No.2, 2020.

[4]. Bholanath Mukhopadhyay , Dr. Rajesh Bose, Dr. Sandip Roy, A Novel Approach to Load Balancing and Cloud Computing Security using SSL in IaaS Environment, International Journal of Advanced Trends in Computer Science and Engineering, pp. 2130-2137,Volume 9, No.2, 2020.

[5]. Arthur, Menaka Pushpa. "Detecting Signal Spoofing and Jamming Attacks in UAV Networks using a Lightweight IDS." 2019 International Conference on Computer, Information, and Telecommunication Systems (CITS). IEEE, 2019.

[6]. Otomo, Safa, Burak Kantarci, and Hussein T. Mouftah. "On the feasibility of deep learning in sensor network intrusion detection." IEEE Networking Letters 1.2 (2019). 68-71.

[7]. Vinayakumar, R., et al. "Deep learning approach for the intelligent intrusion detection system." IEEE Access 7 (2019). 41525-41550.

[8]. Sheu, Ruey-Kai, et al. "IDS-DLA. Sheet Metal Part Identification System for Process Automation Using Deep Learning Algorithms." IEEE Access 8 (2020). 127329-127342.

[9]. Abou Khamis, Rana, M. Omair Shafiq, and Ashraf Matrawy. "Investigating Resistance of Deep Learning-based IDS against Adversaries using min-max Optimization." ICC 2020-2020 IEEE International Conference on Communications (ICC). IEEE, 2020.

[10]. Panagiotis I. Radogloa-Grammatikis; Panagiotis G. Sarigannidis, "Flow anamoly based Intrusion Detection System for Android Mobile Devices", 2017 6th International Conference on MOCAST, May 4-6, 2017, Kazani, Greece.

[11]. Trae Hurley, Jorge E. Perdomo, Alexander Perez-pons, "HMMBased Intrusion Detection System for software-defined networking", 2016 15th IEEE Conference on Machine Learning and Application, Dec 18-20, 2016, Miami, Florida.

[12]. Loganathan, Gobinath, JagathSamarabandu, and Xianbin Wang. "Sequence to sequence pattern learning algorithm for real-time anomaly detection in network traffic." 2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE). IEEE, 2018.

[13]. Kumar, A., Vengatesan, K., Vincent, R., Rajesh, M., & Singhal, A. (2019). A novel Arp approach for cloud resource management. International Journal of Recent Technology and Engineering (IJRTE), 7(6).

[14]. Mayank Agarwal, SankethPurwar, Santosh Biswas, Sukumar Nandi, "Internal Detection System for PS-Poll DOS attack in 802.11 networks using real-time discrete event system", IEEE,vol.4,issue4,2017.