IJEW 国际工程期刊 International Journal of Engineering Works انجینئرنگ کاموں کا بین الاقوامی جرنل

International Journal of Engineering Works
ISSN-p: 2521-2419
ISSN-e: 2409-2770
Vol. 7, Issue 09, PP. 293-297, September 2020
https://www.ijew.io/
https://doi.org/10.34259/ijew.20.709293297

Intrusion Detection System for SDN based IoT Devices using Deep Neural Network

Naqib Ullah[1], Dr. Abdus Salam[2]
[1,2]Abasyn University Peshawar
naqibullah14044@gmail.com[1]

Received: 16 August, Revised: 25 August, Accepted: 29 August

***Abstract*—** One of the emerging technologies in the field of networking is the Software Defined Networking (SDN). Since it is a centrally controlled networks, it provides us with a better control to improve the security within our network against the potential threats. In this work we are using Deep Neural Network (DNN) model to detect the flow-based anomaly within the network. The model was trained on NSL-KDD dataset and out of forty-one features only six of the most relevant features of NSL-KDD were used. The results show that Deep Learning approach shows some promising results in detecting the anomaly in the SDN environment.

***Keywords*—** Internet of Things, Software-defined networking, anomoly detection, machine learning

I. INTRODUCTION

Traditional Network Architectures have not changed much in the past decades but their use has exponentially increased. Providing the additional benefits like manageability, cost effectiveness, adaptability and being dynamic, Software Defined Networks (SDN) are emerging due to the dynamic applications of today's world [1]. SDN gives a higher degree of control to the programmers by decoupling control and data forwarding functions, hence enabling a better abstraction for applications. Due to the additional advantages OpenFlow and SDN is a centre of interest in both the industry and academic domains. Large enterprises and datacentres (e.g. Google B4, Huawei) are harnessing the promising advantages that SDN has to offer. A new market for SDN software providers like NOX, Floodlight and Ryu is also emerging. A variety of supporting hardware are also offered by mainstream vendors like HP, Cisco, Dell and Intel. The first communication interface standard between control and forwarding layer is OpenFlow protocol. In OpenFlow traffic of the network is identified using the concept of data flow. Counters are used to record information. Flow is referred to as IP packets having similar properties, that are passing from a specific point of observation. The dynamics of SDN makes it very venerable to attacks by foreign agents. Alongside, the decoupled control and centralized system architecture also makes it easy to detect and react to the attacks. As Kreutz states SDN contains seven threat vectors. Three of the seven vectors are related to the data control interface. A system named NIDS (Network Intrusion Detection System) guards the network against software attacks [2]. NIDS employs two strategies to detect network attacks. Among the two, the first one is signature-based detection. It uses a prebuilt data base for intrusion identification. Despite being unable to detect novel attacks, signature-based detection is the most common approach used. The second strategy uses a machine learning based approach, in this technique normal behaviour is compared with the new incoming data and the quantitative distance is used to detect anomaly. This strategy is referred to as anomaly-based detection and is able to detect zero-day attacks that system has not seen before. This second approach is usually combined with flow-based data monitoring system, when deploying NIDSs. Being based on packet header information, flow-based NIDSs has significantly less amount of data, if compared to payload-based NIDSs. Machine learning (ML) has been very popular in face detection and speech recognition but the domain of anomaly detection has not been explored extensively. Vern Paxson and Robin Sommer has described the factors that directly affects the use of ML in anomaly detection in networks [3-4]. Deep Learning is now emerging as a new standard for many of the computer science problems and is extensively in use for the vocal and visual data applications. Since deep learning offers the capabilities of corelating data, it displays itself as a potent standard for the upcoming generation of intrusion detections. Since the system can detect zero-day attacks, a high detection rate can easily be expected [5-8].

In this paper a flow-based intrusion detection using Deep Learning approach is proposed. We deploy a Deep Neural Network and use it for NIDS in the software defined network. The model is trained and tested using the NSL-KDD dataset. Optimal hyperparameters of the neural network are extracted by experimentation and performance results (false alarm rate and detection rate) are obtained. Performance accuracy of 75.75% is observed, which is a reasonable performance taking into consideration that only six network features were used.

Contribution:

1. Proposing a fully connected deep neural network for flow-based intrusion detection for IoT network.

2. The intelligent SDN orchestration is proposed that can monitor the network states globally for achieving better results.

Organization of the paper is as follows: section 2 displays related work; section 3 briefly explains the architecture of our deep learning model and introduces the NSL-KDD dataset; section 4 is dedicated to the performance analysis; section 5 states the conclusion and proposes our future work.

II. Related Work

The Flow based anomaly detection is particularly a centre of research now a days. Multi-layered Perceptron and Gravitational optimization-based algorithm was proposed by [11]. One-Class SVM (Support Vector Machine) was used for NIDS by [12], hence achieving a low false alarm rate. Anomaly detection systems in traditional networks are extensively studied and are applied to software defined networks. The privilege of programmability control in SDN is used by [13] to secure SOHO (Small Office, Home Office) network environments. Four of the most prominent anomaly detection algorithms are implemented in SDN using OpenFlow compatible switching and NOX controller. These detectors include Threshold Random Walk with credit-based rate limiting algorithm (TRW-CB), NETAD, Max. entropy detector and rate limiting algorithm. Experimentation has shown a significant improvement in accuracy, if compared to traditional Internet Service Provider (ISP) without introducing additional overhead on the network, especially for (SOHO) networks.

The will connected and centrally controlled architecture of SDN makes it venerable to DDoS (Distributed Denial of Services) attacks. The plot for DDoS attacks are to make the network resources scares for the potential user. The attack is attempted by two or more agents and launched to target application layer, control layer and infrastructure layer plane. These sorts of attacks are hard to identify but are easy to execute. DDoS are getting common due to the networks constructed by bots or machine containing malwares. [14] States the increase of DDoS attacks by 125.36% in one year in 2015. [15] Presents a light weight DDoS attack having 6-tuple features: Avg. duration per flow (ADF), Percentage of pair flow (PPF), Growth of single flow (GSF), Growth of Different Ports (GDP), Avg. bites per flow (ABF) and Avg. Packets per flow (APF). For calcification Self Organizing Maps (SOMs) are used. [16] proposes the combination of OpenFlow and s-Flow to effectively scale the intrusion detection and its mitigation. [17] uses the approach of combining the threshold detection and fuzzy inference system (FIS) for risk prediction of DDoS attacks. It uses three features for attack identification: Distribution of packet quantity per flow, interval time and flow quantity observed by the server. Other features are used by different researchers to improve detection accuracy.

We use Deep Neural Network for detection of the intrusion and the six basic features that we are using in order to detect the anomaly are as follows: srv_count, dst_bytes, count, src_bytes, protocol_type and duration. We are using simplex per-processing in SDN context, that makes our work different from the previous researchesThe will connected and centrally controlled architecture of SDN makes it venerable to DDoS (Distributed Denial of Services) attacks. The plot for DDoS attacks are to make the network resources scares for the potential user. The attack is attempted by two or more agents and launched to target application layer, control layer and infrastructure layer plane. These sorts of attacks are hard to identify but are easy to execute. DDoS are getting common due to the networks constructed by bots or machine containing malwares. [14] States the increase of DDoS attacks by 125.36% in one year in 2015. [15] Presents a light weight DDoS attack having 6-tuple features: Avg. duration per flow (ADF), Percentage of pair flow (PPF), Growth of single flow (GSF), Growth of Different Ports (GDP), Avg. bites per flow (ABF) and Avg. Packets per flow (APF). For calcification Self Organizing Maps (SOMs) are used. [16] proposes the combination of OpenFlow and s-Flow to effectively scale the intrusion detection and its mitigation. [17] uses the approach of combining the threshold detection and fuzzy inference system (FIS) for risk prediction of DDoS attacks. It uses three features for attack identification: Distribution of packet quantity per flow, interval time and flow quantity observed by the server. Other features are used by different researchers to improve detection accuracy.

We use Deep Neural Network for detection of the intrusion and the six basic features that we are using in order to detect the anomaly are as follows: srv_count, dst_bytes, count, src_bytes, protocol_type and duration. We are using simplex per-processing in SDN context, that makes our work different from the previous researches.

The above techniques perform well for small datasets. Nowadays, the IoT networks comprise of dynamic traffic generated from heterogeneous networks and it is not possible to detect the DDoS attacks from dynamic traffic. Thus, this study will use the DNN approach for improving the accuracy and reducing the FAR of highly dynamic IoT traffic. The proposed algorithm will be compared with the benchmark algorithm [15] using the Deep Neural Network (DNN) for intrusion detection.

III. Proposed Architecture

The proposed architecture is shown in Figure 3. The DNN algorithm will be implemented in the NIDS module in the SDN controller. The controller will have the global view of the network and will monitor the the states and traffic generated from the network. The network statistics will be forwarded to the SDN controller using the ofp_flow_stats_request request. The controller will forward the traffic to NIDS module that will correlate and analyze the forwarded traffic for network intrusion. The traffic forwarded from the SDN switches will be forwarded to the SDN controller that will be monitoring the

traffic using the DNN algorithm. If Intrusion detection is found in the traffic, it will be notified to the network administrator.

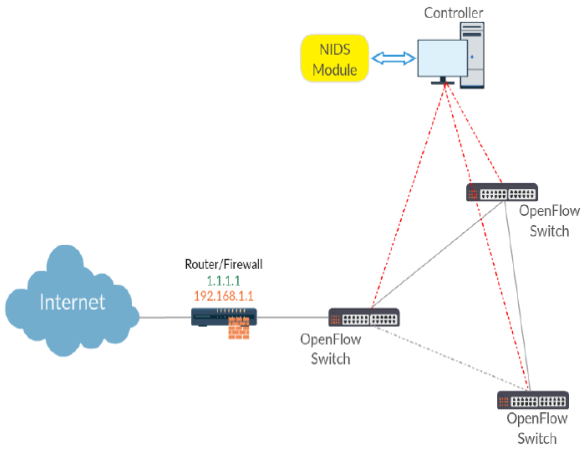

Figure 1. Proposed SDN-based architecture

IV. METHODOLOGY

a) Deep Learning Model

If we consider traditional machine learning model, important features are needed to be manually designed and then the system trains the model according to the provided features. In contrast to that deep learning multiple layers of features are automatically detected and are used to produce output. These features are hierarchically layered, which implies that features in later layers are discovered from previous layers. Our neural network is five layered networks containing three hidden layers and two input/output layers as shown in Fig.1. It has six input dimensions and two output dimensions. Hidden layers have twelve, six and three neurone layers respectively. Batch size of 10 and 100 epoch are used for model initialization. Learning rate is concluded after experimentation. The architecture of neural network can be seen in figure 1.

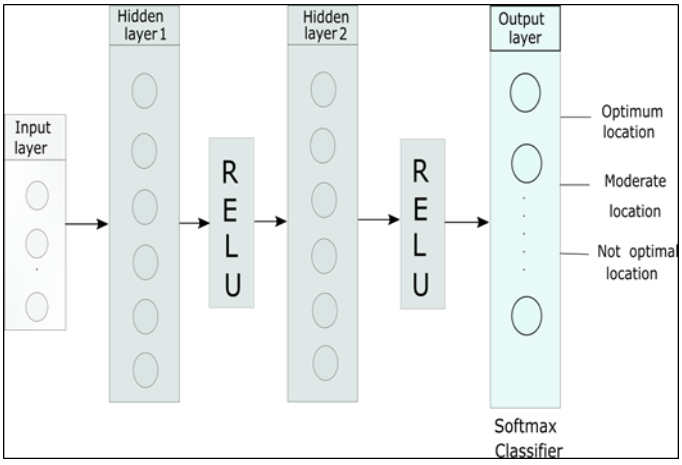

Figure 2. Architecture for Deep neural network

b) Dataset

NSL-KDD dataset is the perfect representation for the current real-world networks. The dataset was proposed for a data mining competition KDD Cup in 1999 [18]. It is used as a standard for NIDS benchmark for software defined networks in research, hence providing us with an ability to compare our performance with the previous work. KDDTrain+ contains 125,973 samples and 22,554 network traffic samples are available for KDDTest++. Each sample is defined by forty-one feature vector, that are divided into three following types: basic features, content- based features, traffic-based features. Similarly, attacks are also divided into for separate categories according to their characteristics. Table 1 describes the categories. Detection task is made more realistic due to the lack of some categories in the training dataset. Out of the forty-one available features in NSL-KDD Dataset six above mentioned features are used for our model. These features are traffic-based basic features that can be easily obtained in an SDN environment and are described in Table 2.

c) SDN-Based IDS Architecture:

As displayed in Fig.2 we proposed an SDN-based NIDS architecture. As the figure shows NIDS module is implemented inside the controller. As all the switch activity is monitored by the controller, we can leverage the global overview of the network for detecting the intrusions and anomalies in the network. In order to request the network statistics a request message of ofp_flow_stats_request is sent from controllers to all OpenFlow switches. A ofp_flow_stats_reply message is sent back to the controller that contains all the statistics. The centralized architecture is the key feature of the network that is put into use for analysing and detecting network intrusions. In case of a anomaly in the network flow table is modified in order to mitigate the intrusion and security policies are sent to the switches for the intrusion prevention.

V. PRELIMINARY EXPERIMENTAL RESULTS

1. Evaluation Metrices

The criteria for the performance evaluation are accuracy (AC), recall (R), precision (P) and F-measure (F). The target is to get high accuracy and detection rate while keeping the false alarm rate low. We use confusion matrix to converge to these parameters. True Positive (TP) and True Negatives (TN) refer to correctly classified attack and normal records. Similarly, False Positive (FP) and False Negative (FN) are the number of attacks and normal records incorrectly classified.

- The percentage of True detection is referred to as Accuracy (AC):
- Precision (P) is the ratio of True Positives with the total number of detections by the system:
- The percentage of predicted intrusion compared to all intrusions is displayed using Recall (R):

- F-measure (F) is for an improved measure of accuracy for NIDS, considering both the precision and recall:

2. Experimental Results

NIDS is initially implemented for two-way classification. Model performance is dependent on the initial hyperparameters. After optimizing the hyperparameters we obtained the optimal classification. Learning rate was varied by the following range {0.1, 0.01, 0.001, 0.0001}. The goal was to maximise the accuracy. Table I below, displays a comparison of accuracy for training phase and testing phase. We can observe accuracy has directly proportional relation with the learning rate respectively. While in testing phase the learning rate of 0.001 performs better than the learning rate of 0.0001. The reason is the model will be overfitted due to the very small learning rate. Hence, 0.0001 is the choice for the learning rate, so the generalizability of the model is not compromised. As a result, new intrusions could be caught by the system.

We also calculated precision, recall and f-measure of the model. The performance was evaluated using the provided test data, which is displayed in the Table II. It can be observed that the best results are obtained with 0.001 learning rate. Evaluation metrices are grown as we decrease learning rate from 0.1 to 0.001 but the performance deteriorates as it is further decreased to 0.0001.

TABLE I. ACCURACY FOR DIFFERENT LEARNING RATES

Learning rate	Train data Accuracy (%)	Test data Accuracy (%)
0.1	89	73
0.01	91	75
0.001	93	76
0.001	95	77

TABLE II. METRICS FOR DIFFERENT LEARNING RATES

Learning rate	Precision (%)	Recall (%)	F1-score
0.1	80	73	73
0.01	83	75	75
0.001	85	76	77
0.001	86	77	78

Comparison of our results with other machine learning algorithms was also conducted. As in [15] six features (APF, ABF, ADF, PPF, GSF and GDP) for DDoS were used and false alarm rate of 0.46% and detection rate of 99.11% was extracted. Results from our system were low if compared to their systems because of the features that we have selected for testing and training. The features that they have selected directly targets detection of DDoS attacks while we opted for the six very basic features. The future goal is to apply six features to our model for further evaluation. [10] uses machine learning algorithm using forty-one training and testing features.

TABLE III. ACCURACY COMPARISION WITH BENCHMARK

Algorithms	Precision (%)
SVM	74
Decision tree	76
Proposed DNN	85

Results of these experiments show the performance evaluation of these algorithm. As can be observed in Table III DNN approach performs best compared to benchmark algorithmns. It can be observed that Deep learning algorithms perform better than other algorithms. Our system performs more accurately, giving us low false positive if compared with other algorithms. Hence our system (DNN approach) is capable of genializing the characteristics of the network with reasonable accuracy using fewer number of features

CONCLUSION

The paper evaluates NIDS and implements deep learning algorithm for network intrusion detection. Despite our results not begin reliable enough for commercial purposes, our system yields significant advantages in certain areas and have scope for further improvements. Our paper shows the potential for deep learning in flow-based anomaly detection. Deep learning approach also has potential in the context of SDN, due to the centralized design of SDN. A deep learning model can easily extract real time basic features from the network controllers and analyse them. We plan to propose other types of features in order to improve our accuracy. The architecture of SDN gives us flexibility to use specific features for particular types of attacks, for example DDoS to increase the accuracy of NIDS. We will try to tune our DNN model for better performance and apply this approach to real SDN networks to extract the latency and throughput performances.

REFERENCES

[1] "Software Defined Networking Definition," Available:https://www.opennetworking.org/sdn-resources/sdn-definition,[Accessed 04 Jul. 2016].

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson,J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, pp. 69–74, 2008

[3] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu et al., "B4: Experience witha globally-deployed software defined wan," ACM SIGCOMM Computer Communication Review, vol. 43, no. 4, pp. 3–14, 2013.

[4] C. T. Huawei Press Centre and H. unveil world's first commercial deployment of SDN in carrier networks, "[online]. available: pr.huawei.com/en/news/hw-332209-sdn.htm."

[5] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "Nox: towards an operating system for networks," ACM SIGCOMM Computer Communication Review, vol. 38, no. 3, pp. 105–110, 2008.

[6] "Ryu," Available: http://http://osrg.github.io/ryu/.

[7] "Floodlight," Available: http://www.projectfloodlight.org/.

[8] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking.ACM, 2013, pp. 55–60.

[9] R. Sommer and V. Paxson, "Outside the closed world: On using machinelearning for network intrusion detection," in 2010 IEEE symposium onsecurity and privacy. IEEE, 2010, pp. 305–316.

[10] M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications, 2009.

[11] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan, and M. Sheikhan,"Flow-based anomaly detection using neural network optimized with gsa algorithm," in 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, 2013, pp. 76–81.

[12] P. Winter, E. Hermann, and M. Zeilinger, "Inductive intrusion detection in flow-based network data using one-class support vector machines,"in New Technologies, Mobility and Security (NTMS), 2011 4th IFIPInternational Conference on. IEEE, 2011, pp. 1–5.

[13] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in International Workshop on Recent Advances in Intrusion Detection. Springer, 2011, pp. 161–180.

[14] "Q1 2016 State of the Internet / Security Report," Available: https://content.akamai.com/PG6301-SOTI-Security.html, [Accessed 07Jul. 2016].

[15] R. Braga, E. Mota, and A. Passito, "Lightweight ddos flooding attackdetection using nox/openflow," in Local Computer Networks (LCN),2010 IEEE 35th Conference on. IEEE, 2010, pp. 408–415.

[16] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments," Computer Networks, vol. 62, pp. 122–136, 2014.

[17] P. Van Trung, T. T. Huong, D. Van Tuyen, D. M. Duc, N. H. Thanh, and A. Marshall, "A multi-criteria-based ddos-attack prevention solution using software defined networking," in Advanced Technologies forCommunications (ATC), 2015 International Conference on. IEEE,2015, pp. 308–313.

[18] "KDD Cup 1999," Available: http://kdd.ics.uci.edu/databases/kddcup99/,[Accessed 04 Jul. 2016].