2nd IEEE International Conference on Electronic Engineering, Menoufia University, Egypt
ICEEM2021

An Efficient Intrusion Detection System for SDN using Convolutional Neural Network

Heba A. Hassan
Department of Electronics and Communication Engineering, Faculty of Electronic Engineering, Menoufia University,
Ommolham2010@gmail.com

Ezz E. Hemdan
Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University
ezzvip@yahoo.com

Walid El-Shafai
Department of Electronics and Communication Engineering, Faculty of Electronic Engineering, Menoufia University,
eng.waled.elshafai@gmail.com

Mona Shokair
Department of Electronics and Communication Engineering, Faculty of Electronic Engineering, Menoufia University,

shokair_1999@hotmail.com

Fathi E. Abd El-Samie
Department of Electronics and Communication Engineering, Faculty of Electronic Engineering, Menoufia University,
fathi_sayed@yahoo.com

2021 International Conference on Electronic Engineering (ICEEM) | 978-1-6654-1842-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICEEM52022.2021.9480383

Abstract— With the accelerated development of computer network utilization and the enormous growth of the number of applications running on top of networks, network security has become more significant. Intrusion Detection Systems (IDS) are considered as essential tools that can be utilized to protect computer networks and information systems. Software-Defined Network (SDN) architecture is used to provide network monitoring and observation of functions. Generally, an IDS is developed to observe the regular traffic to the SDN in order to maintain a high level of security. This paper introduces an efficient IDS using Convolutional Neural Network (CNN). This IDS is applied on a new attack-specific SDN dataset called InSDN. The proposed IDS is compared in performance with different machine-learning-based systems such as Decision Tree Classifier (CART), Logistic Regression (LR), Support Vector Machine (SVM), Naïve Bayes (NB), Random Forest (RF) classifier, and AdaBoost (AB) classifier.

Keywords — Intrusion Detection System (IDS), Software-Defined Network (SDN), Convolutional Neural Network (CNN).

I. Introduction

Due to continuous rise of cyber-attacks all over the world [1], the research in IDS grows quickly in the academic and industrial communities. The most common cyber-attacks are web-based attacks, denial-of-service attacks, and malicious insider attacks. These cyber-attacks may allow malicious software to creep into the system. Hence, to avoid unauthorized access, some programs like antivirus software, firewalls, and IDS are deployed by several organizations to protect them from losing their intellectual property.

To determine the cyber-attacks rapidly, we should identify the attack process firstly [1] based on the IDS. Then, we should use the IDS to identify malicious activities, including viruses, worms, and Distributed Denial-of-Service (DDOS) attacks. Irregularity detection speed, accuracy, and reliability are the basic assessment factors for an IDS. The IDS have been developed with the evolution of CNNs. This is attributed to the ability of CNNs to work on network traffic analysis as a pattern recognition problem.

The main property of an SDN architecture is that it has the ability to separate forwarding functions and network control. Hence, network control can be accomplished, directly [2]. This separation makes network management easy [3]. This feature of SDN introduces several advantages. First, it facilitates network system management and reduces human intervention. In addition, it enables IT administrators to manage network devices without limitation to a particular vendor. Finally, it decreases operation cost compared with those of the conventional networks, since no programming language is required for the underneath infrastructure devices. Although SDN technology has several benefits, SDN is vulnerable to new security threats. This becomes more dangerous, when the SDN controller is accessed by the attacker. Therefore, IDS techniques can be deployed in SDN networks to detect anomalies.

There are numerous difficulties for deploying IDS systems on the SDN standard. First, there is no public dataset that is available for anomaly detection systems. Most researchers use intrusion detection datasets. Unfortunately, current datasets just show specific types of attacks like DoS and DDoS, and the other attacks are ignored. In addition, these datasets belong to intrusions produced in a single component of the SDN. Attack vectors for different SDN layers are ignored.

Information and communication technology (ICT) systems have become more significant in every aspect of organization and daily life. On the other hand, the enormous growth of the attacks on ICT systems requires an effective integrated network security solution.

The IDS are considered the most widely-used tools for recognizing different sorts of attacks. At the same time, the SDN architecture is used to provide network monitoring and observation of functions. The IDS are developed to monitor the incoming traffic to the SDN in

978-1-6654-1842-3/21/$31.00 © 2021 IEEE
https://iceem2021.conferences.ekb.eg

order to maintain a high level of security. An IDS based on a CNN is proposed and applied on a new attack-specific SDN dataset named InSDN. The proposed system provides good results compared with various machine-learning-based systems [5], such as CART, LR, RF, SVM, NB and AB.

This paper is organized as follows. Sec. II gives a concise description of the IDS followed by the architecture of the CNN. Sec. III introduces the SDN and the attacks on different SDN layers. Sec. IV provides a new attack-specific SDN dataset named InSDN. Sec. V gives a discussion SDNs that adopt IDS. Sec. VI gives the evaluation study of the proposed system. Sec. VII gives the concluding remarks of the paper.

II. Related Work

Intrusion Detection System (IDS) is a basic research area in the cybersecurity field. The IDS operation depends on a classification task for the traffic in either a binary or multi-class classification scenario. In binary classification, we distinguish between normal and anomalous traffic classes. On the other hand, in multi-class classification, several classes such as Root-to-Local (R2L), User-to-Root (U2R), Denial-of-Service (DOS), and Probing (Probe) classes are considered. The major objective of intrusion detection is to successfully recognize the intrusive behavior and increase the accuracy of classifiers. Generally, IDS are intended to identify attacks, early. Due to the dynamic nature of attacks, various issues should be considered, while implementing IDS such as the adaptability of the detection method. In addition, the dimensions of the dataset should be reduced. Hence, feature selection is badly needed.

Techniques of intrusion detection can be divided into anomaly detection and misuse detection techniques [6]. In anomaly detection, the objective is to detect both network and computer intrusions by checking the system activity, and then classifying the type of traffic as being normal or anomalous. The classification here relies upon heuristics or rules. Most anomaly detection systems have two phases. The first is the training phase in which a profile of normal behavior is determined. The second is the testing phase in which current traffic is compared with the profile made in the training phase.

On the other hand, the main objective of misuse detection is how to detect computer attacks. This is done by defining an abnormal system behavior at first, and then any deviation can be considered as a normal behavior. Misuse detection depends on patterns, signatures, or attempts. The advantage of using misuse detection is the simplicity of adding known attacks to the model. So, it is used more generally to refer to all kinds of computer misuse. The fundamental weakness of misuse detection is the inability to recognize unknown attacks. Hence, most intrusion detection systems depend on a combination of two techniques and are often deployed on the network, on a specific host, or even on an application within the host.

Anomaly detection has the ability to discover novel attacks in contrast to misuse detection. So, IDS with anomaly detection are applied on the SDN standard, but there is a problem. Threshold computing methods or statistical measures are generally used to overcome network intrusions. These methods may not be efficient with complex attack patterns.

Machine learning offers solutions to network intrusions and allows real-time IDS, but several solutions have restrictions on the learning patterns of normal and attack records. Deep learning is viewed as an intricate model of machine learning algorithms. It can be suitable for the abstract representation of complex hierarchical features globally with sequence information of TCP/IP packets.

Deep learning algorithms have become more significant due to their main important characteristics. There is a need for learning of the long-term dependencies of temporal patterns in large-scale sequence data, in addition to hierarchical features. Object detection, detecting network intrusion, and visual object recognition [4] are some applications that use deep learning algorithms. Supervised and unsupervised ways are used to train a deep learning algorithm [7]. The CNN [4] is illustrated as an example of deep learning algorithms that uses a supervised way for training. The CNN architecture is utilized in general in applications such as face recognition [4] and 2D images [8].

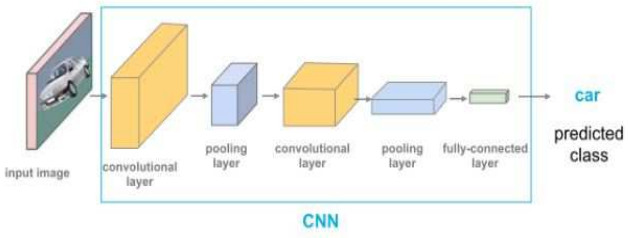

Figure 1. Architecture of ConvNet [9]

Medical image analysis, financial time series analysis, image classification, image and video recognition are some applications that use convolutional neural networks (ConvNets) [9], which are deep neural networks. As seen in Figure 1, the ConvNet has three main layers: input layer, output layer, and multiple hidden layers which consist of a series of convolutional layers like pooling layers, fully-connected layers, and normalization layers. These layers are called hidden layers, since their outputs and inputs are veiled by the activation function and the final convolution. The most widely-used activation function is the ReLU function.

III. Software Defined Networking

A software-defined network is a new technology that has the ability to separate the network control and forwarding functions. Hence, programming of the network control can be achieved, directly [2]. The separation feature of SDN makes network management easy [3] and introduces several advantages. First, it facilitates innovative applications. Then, it helps for dictating a new networking paradigm with the ability to implement IDS [10]. To maintain a high level of security and network monitoring, it is required to allow machine learning and deep learning (ML/DL) approaches to be merged with

SDN controllers [11]. On the other hand, ML/DL approaches can be merged with SDN-based intrusion detection to introduce several advantages such as high Quality of Service (QoS), security enforcement, and virtual management. Other advantages introduced by SDN are enhancing the network security, eliminating hardware dependency and achieving flexibility to program network devices.

A. SDN architecture

The most widely-used SDN protocol that manages the relation between the controller and the switches is the OpenFlow [12, 13]. An SDN depends on Application Programming Interfaces (APIs). These APIs, usually called northbound APIs, empower effective administration coordination and computerization. Accordingly, the SDN empowers a system to shape movement and send administrations to address-changing business needs, without touching any individual switch or any switch in the sending plane. The SDN is another system that tends to empower more agile and financially savvy systems. The Open Networking Foundation (ONF) leads the pack in SDN standardization [2]. Figure 2 introduces a simple logical representation of SDN architecture. The ONF/SDN engineering comprises three layers that are available through open APIS:

- The application layer is specialized for expanding the SDN communication services. Both application layer and control layer are separated by the northbound API.
- The control layer is specialized for overseeing the network forwarding behavior through an open interface by leveraging from the centralized control.
- The infrastructure layer is specialized for packet switching and forwarding by using Network Elements (NEs) and devices.

B. Attack vectors in SDN elements

The SDN network has become more vulnerable to different types of attacks due to the centralization feature of the SDN architecture. Three layers of the SDN architecture are subject to various types of attacks [14] like attacks in the SDN controller, attacks in the communication between the control and data plane, attacks in the application layer, and attacks in data plane elements. The effects of four vectors network are illustrated as follows:

- *Attacks in the data plane:* The network elements in the SDN are the objective of these attacks. These attacks ruin the regulator assets or data related to OpenFlow switchs. The network traffic is deviated by the attacker, who conveys a phony switch in the SDN network. Hence, network resources are damaged.
- *Attacks in control plane communication:* These attacks, in general, separate the communication between the controller and the data plane elements. Hence, the SDN controller is disengaged from the entire network elements, and it cannot deal with the data plane devices through correspondence channels.
- *Attacks in SDN controller:* The SDN controller is the target here. The attacker tends to disrupt the whole system by obtaining entrance or cutting down the SDN controller. Hence, the attacker becomes able to control the whole network in the event that he effectively misuses the vulnerable northbound API.
- *Attacks in the application plane:* The security policy is violated by the attacker, who runs malicious applications. Hence, the IDS applications are bypassed.

It is noticed that the second and third types of attacks exist in SDNs, because the data plane is separated from the control plane. The first and fourth types of attacks are common in both SDN and conventional networks. Therefore, conveying IDS to recognize anomalies in the SDN traffic is fundamental in the network architecture.

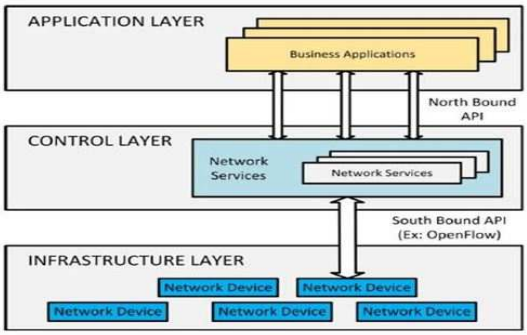

Figure 2. SDN architecture [2]

IV. DATASET DESCRIPTION

TABLE I. INSTANCES AND SIZE OF THE USED DATASET

Dataset	Traffic Distribution		Number of Instances	Total	PCAP size
InSDN (metasolit)	*Normal*		68424	68424	4.23332 GB
	DDoS	*Attack*	73529	136743	
	Probe		61757		
	DoS		1145		
	Brute-force-attack		295		
	Exploitation (R2L)		17		

As seen from Table (1), the number of InSDN [16] instances is 205167 for normal and attack traffic. Normal traffic includes a total of 68424 instances, while attack traffic contains 136743 instances. The names of PCAP files are picked depending on the objective convention layer or the devices that are utilized for creating each file. This dataset includes 5 classes of attack and normal traffic.

V. PROPOSED SYSTEM

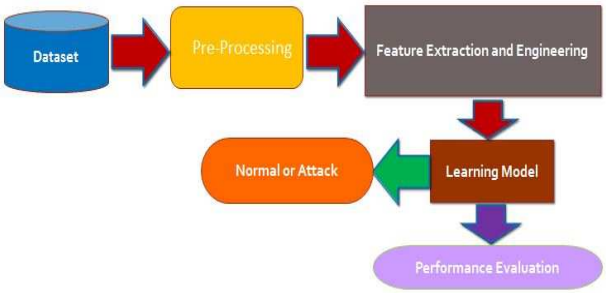

Figure 3. Proposed SDN-based IDS.

The programming environment of the SDN controller allows SDN architecture to provide a network monitoring and analysis mechanism. Then, again an IDS is developed to observe the incoming traffic to the SDN. Hence, it enables the SDN to adjust the security service insertion. The SDN-based IDS is mainly concerned with the SDN controller. This IDS is utilized to recognize the malicious flow by examining the traffic intended to the SDN. After analyzing traffic, the SDN switches receive the traffic coming from outside the network. Then, the SDN switches check flow table entries concerned with transmitted packets.

Figure 3 shows the proposed SDN-based IDS. First, pre-processing is performed on the dataset, which incorporates transformation to numbers and normalization [15]. In the transformation to numbers, non-numeric features are changed over into numeric ones by using encoding. On the other hand, in normalization, features are scaled i.e. the value of every feature is scaled to the interval [0,1]. Hence, the dataset turns out to be more appropriate for the training phase and the over-fitting problem is avoided. Then, in the feature selection step, optimal features are chosen from the SDN. These features are separated from the SDN controller via OpenFlow calls, and then intended to SDN switches. Finally, six machine learning algorithms are analyzed to estimate the usability and quality of the proposed architecture. These algorithms are CART, RF, AdaBoost, NB, LR, and SVM. The CNN algorithm is also analyzed as an example of deep learning algorithms. To assess the proposed system, we should use the most important performance indicators like precision, recall, and F-score.

VI. SIMULATION RESULTS

This section studies the evaluation of the proposed system.

A. Experimental evaluation

All experiments were executed through Python programming language with Anaconda spyder3 that has numerous libraries like Keras, Scikit-Learn, and Tensorflow. We have used a workstation characterized by Intel(R) HD Graphics 6000, Core(TM)i5-5250U CPU@1.60GHz,Ubuntu 15.10 with 4 GB of RAM.

TABLE II. COMARTIVE RESULTS OF THE CNN-BASED PROPOSED SYSTEM AND OTHER ALGORITHMS

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression (LR)	0.921	0.997	0.883	0.936
Naive Bayes (NB)	0.953	0.957	0.938	0.947
Decision Tree Classifier (CART)	0.943	0.927	0.957	0.941
AdaBoost Classifier(AB)	0.947	0.932	0.960	0.945
Random Forest Classifier (RF)	0.951	0.936	0.963	0.949
Support Vector Machine (SVM)	0.965	0.953	0.973	0.962
Proposed system using CNN	0.992	0.992	0.989	0.990

Different performance indicators have been utilized to assess the efficiency of supervised learning techniques. These indicators include F-score, recall and precision.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F-score} = \frac{2 \times \text{Precission} \times \text{Recall}}{\text{Precission} + \text{Recall}} \quad (4)$$

True Positive and True Negative are abbreviated as TP and TN, respectively. In addition, False Positive and False Negative are abbreviated as FP and FN. Table II reveals that the proposed CNN model provides good results compared with different machine learning algorithms like CART, LR, RF, SVM, NB and AB.

VII. CONCLUSION

Recently, IDS have been presented to identify attacks, early. Due to the dynamic nature of attacks, we should consider various issues, while implementing the IDS such as the adaptability of the detection method, but there are numerous difficulties like monitoring and implementing real-time intrusion detection in high-speed networks. Hence, SDN-based IDS architectures should be developed. This paper introduced an efficient IDS based on CNN to be applied on new attack-specific SDN datasets. The simulation results showed that the CNN model provides good performance compared with different machine learning algorithms like CART, LR, LDA, SVM, NB and AB.

REFERENCES

[1] H. P. Enterprise, "2015 cost of cyber crime study: global", independently conducted by Ponemon institute LLC, October 2015.

[2] M. Mousa, A. M. Bahaa-Eldin, and M. Sobh, "Software Defined Networking concepts and challenges," in International Conference on Computer Engineering and systems (ICCES), 2016 (pp-79-90)

[3] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning review", Weekly journal of science in nature international. Nature 521, May 2015.

[5] J. Brownlee, "Supervised and unsupervised machine learning algorithms", in Machine Learning Algorithms, March 2016.

[6] K. Lahre, T. Diwan, S. Kashyap, P. Agrawal, "Analyze Different approaches for IDS using KDD 99 Data Set", International Journal on Recent and Innovation Trends in Computing and Communication , Volume: 1, Issue: 8, AUG 2013, pages 645-651.

[7] A. Sforzin, M. Conti, F. G. Marmol, and J.-M. Bohli, "RPiDS: raspberry Pi IDS a fruitful intrusion detection system for IoT", in International IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, pp. 440–448 July (2016).

[8] Q. Lan, Z. Wang, M. Wen, C. Zhang, and Y.Wang, "High Performance Implementation of 3D Convolutional Neural Networks on a GPU", in Computational Intelligence and Neuroscience, November 2017.

[9] P. Kim, " MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence", National Rehabilitation Research Institute of Korea, June 2017.

[10] S. Sezer et al., "Are we ready for SDN? Implementation challenges for software-defined networks," in IEEE Communications Magazine, vol. 51, no. 7, pp. 36-43, July 2013, doi: 10.1109/MCOM.2013.6553676.

[11] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi and M. Ghogho, "Deep learning approach for Network Intrusion Detection in Software Defined Networking," 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), 2016, pp. 258-263, doi: 10.1109/WINCOM.2016.7777224.

[12] A. Lara, A. Kolasani and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 493-512, First Quarter 2014, doi: 10.1109/SURV.2013.081313.00105.

[13] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, J. Rexford, , and J. Turner, "OpenFlow: enabling innovation in campus networks," ACM SIGCOMM Computer Communication Review 38, no. 2 March (2008): 69-74.

[14] K. Benzekki, A. ElFergougui, and A. E. Elalaoui, “Softwaredefined networking (sdn): a survey,” Security and communication networks, vol. 9, no. 18, pp. 5803–5833, February 2016.

[15] P. S. Autade, P. N. Kalavadekar, "Intrusion Detection System using Recurrent Neural Network with Deep Learning", International Journal of Innovative Research in Computer and communication Engineering. Vol. 7, Issue 4, April 2019

[16] M. S. Elsayed, N. -A. Le-Khac and A. D. Jurcut, "InSDN: A Novel SDN Intrusion Dataset," in IEEE Access, vol. 8, pp. 165263-165284, September 2020.