2018 32nd International Conference on Advanced Information Networking and Applications Workshops

A Deep Learning Framework to Enhance Software Defined Networks Security

Ahmed Dawoud
School of computing, Engineering, and Mathematics.
Western Sydney University, Sydney, Australia.

Seyed Shahristani
School of computing, Engineering, and Mathematics.
Western Sydney University, Sydney, Australia.

Chun Raun
School of computing, Engineering, and Mathematics.
Western Sydney University, Sydney, Australia.

***Abstract* -Software-Defined Networks (SDN) initiates a novel networking model. SDN proposes the separation of forward and control planes by introducing a new independent plane called network controller. The architecture enhances the network resilient, decompose management complexity, and support more straightforward network policies enforcement. However, the model suffers from severe security threats. Specifically, a centralized network controller is a precious target for two reasons. First, the controller is located at a central point between the application and data planes. Second, a controller is software which prone to vulnerabilities, e.g., buffer and stack overflow. Hence, providing security measures is a crucial procedure towards the fully unleash of the new model capabilities. Intrusion detection is an option to enhance the networking security. Several approaches were proposed, for instance, signature-based, and anomaly detection. Anomaly detection is a broad approach deployed by various methods, e.g., machine learning. For many decades intrusion detection solution suffers performance and accuracy deficiencies. This paper revisits network anomalies detection as recent advances in machine learning particularly deep learning proofed success in many areas like computer vision and speech recognition. The study proposes an intrusion detection framework based on unsupervised deep learning algorithms.**

Keywords—Software-defined networks; Deep Learning; Anomalies Detection; Autoencoders

I. INTRODUCTION

The conventional communication networking model consists of three planes. i.e., management, control, and forward or data. The management plane supports network monitoring and configuration. The control plane populates forwarding tables on the physical devices. Consecutively, the forward plane switches packets to ingress and egress ports based on the forwarding tables. For decades, both the

Control and the forward planes are integrated into the same networking devices, for instance. Switches or routers. The conventional model provided efficiency from a performance perspective. However, current networks became excessively complicated, and there is a necessity to adopt a more resilient architecture [1].

This paper introduces a framework to enhance the security deficiencies of SDN.The framework is anomalies detection based on machine learning. The next section discusses SDN model and related security threats. The third section investigates the deep learning and its current anomalies detection solution for network security. The fourth section represents our proposed framework.

II. SOFTWARE-DEFINED NETWORKS AND RELATED WORK

Software-Defined Networks (SDN) networking model detaches control and forward planes [2]. The devices provide forwarding capabilities to switch the data flow, while the control plane is decoupled to introduce a new entity called network controller. The forward plane located at the bottom of the stack includes hardware devices, e.g., switches, routers, and firewalls and intrusion detection systems (IDS). The devices do not possess the software intelligence needed to fill the forwarding tables. The network logic independently relocated to the controller layer.

The controller abstracts the devices and provides resources required to programme low-level forwarding devices. Controller aka Network Operating system (NOS) provides services like network state, and topology information. Additionally, the controller provides northbound, and southbound APIs. The northbound API to facilitate

978-1-5386-5395-1/18/$31.00 ©2018 IEEE
DOI 10.1109/WAINA.2018.00172

communication with the applications. Whereas, the southbound API to provide accessibility between the controller and forwarding devices. OpenFlow is a defacto SDN southbound protocol [3]. The application plane resides on the top of the SDN model stack. Network programmability is a key privilege achieved by SDN model, where applications in the top plane can access the physical devices through the controller. Programmability facilitates and accelerates the innovation with an enormous number of network applications, e.g., monitoring, traffic engineering, security, and cloud applications. Centralization is an essential characteristic of the SDN architecture. A controller is a central entity which provides a global view of the entire network; it eases the management and policies enforcement process. Additionally, it decreases the faults in configuring and deploying the network policies. The centralization enhances the network resilience and interoperability, for instance, multiple of devices from various industrials can be integrated and abstracted in one network.

Security threats are critical challenges in conventional networking systems. The threats are intensifying in SDN networks. The model's many advantages are accompanied by additional threats that were not possible in the traditional networks. For the southbound OpenFlow protocol, a security analysis study exposed various attacks derived from the SDN standard protocol, for example, flow tables and on the devices control channels between the devices and controller affected by a denial of service attacks (DoS). Application privilege conflicts propagate to flow rules. The control channel between the controller and the switch is initiated as a TCP connection, with an option for encryption protocol Transport Layer Security (TLS) to secure the channel. Without an encryption method, the communication between the controller and the forwarding devices are exposed to a man in the middle attacks. Kloti et al. have conducted a security analysis for the OF protocol [4]. The study has deduced that denials of services attacks have threatened the flow tables and the communication channels; as the attacker flood those components with OpenFlow rules and requests. Additionally, tampering attacks have substantially targeted the flow tables on the devices by installing rules from untrusted sources.

Kreutz et al. concluded seven threats vector for SDN [5]. Three threats are directly linked to the controller itself as follows,

- Attacks on the communications between the controller and the data plane devices.
- Attacks on the controller vulnerabilities
- Attacks on the controller originated from untrusted applications

Intrusion Detection Systems are software or hardware systems dedicated to monitoring the traffic for security threats. Standard intrusion detection process includes three phases, collecting data from the network, analyzing, and then launch a proper response if a threat exposed. There are three approaches to analyze the collected traffic named signature-based, anomaly detection, and specification based. Firstly, signature-based, whereas a system has a database of predefined violations' signatures, and the system matches those signatures against the network activity signatures. Secondly, anomalies or outlier analysis, the system concerns about differentiate between the normal and abnormal patterns. For the system, normal activities are identified in a baseline profile, which the system develops in a learning phase. Thirdly the stateful protocol analysis, in this method a predefined pattern of protocols' behavior is established, a comparison is made between network activities and the expected behavior defined by protocols, and in the case of profile violation, an alert is raised. A combination of methods is used to maximize the IDP performance [6]. A significant weakness in the signature-based method is the inability to detect new attacks while the anomaly detection has a higher false alarms rate. The majority of the commercial implementations use a hybrid approach [7].

Anomalies or outliers are unexpected patterns. In the context of networking, we assume the intrusive or attacks are unusual behavior [8]. So at any point, the majority of the traffic is normal. Several approaches were adopted .e.g. statistical methods, machine learning, and biological models. The proposed framework adopts a machine learning approach.

III. AUTOENCODERS

Machine learning is an artificial intelligence approach that focuses on acquiring knowledge from raw data (data representation aka features). In practice, retrieving knowledge means finding patterns. For more than half century, neural networks were an active topic for machine learning and neuroscience. However, there were no breakthroughs in accuracy and performance. Recently, deep learning has revived the neural networks. It has been successfully applied in various areas .e.g., objects and speech recognition [9].

Deep Learning (DL) is deep neural network architecture; the deepness term refers to multi hidden layers between the input and output layers. Figure 1 left side shows shallow neural network with one hidden layer. A deep network is a neural network with hidden layers between the input and output layers. Empirically more hidden layers mean more features to detect. Deep neural networks existed for a long time, however; it was not possible to train the network for three reasons, i.e., Vanishing Gradient Decent in

backpropagation algorithm, poor generalization, and computation power.

The recent advances in DL started in 2006 by a pre-training step using restricted Boltzmann Machines (RBM) [10]. Later, various algorithms were proposed to solve generalization problem these solutions include Rectifier Linear Units (ReLU) and dropouts. DL algorithms are classified into supervised and unsupervised. In supervised learning, the training dataset contains the input data and data labels. This approach is suitable for classification, and regression tasks. In the unsupervised, only an unlabeled dataset is available. Unsupervised applications include clustering, dimensionality reduction, and noise removal. For network anomalies detection we believe the unsupervised approach has the following advantages,

- Unsupervised can detect the internal representation of the dataset; this conforms to the online detection.
- Theoretically unsupervised algorithms will discover the unprecedented threats
- We can use the unsupervised method as a pertaining phase before supervised or Reinforcement Learning (RL)

Unsupervised DL algorithms include Autoencoder and Restricted Boltzmann Machines (RBM). Deep Learning (DL) is a set of non-linear algorithms for multilayers models. DL algorithms manipulate both supervised and non-supervised learning. Unsupervised deep learning algorithms aim to learn probability distribution of a specific dataset. While, the supervised learning algorithm learns to predict p (y |x) an input where x is input vector, and y is the output. Applying the unsupervised learning potentially reveals exciting features in the data sets. The automatic discovery of the features improves the probability of detecting new attacks in the contexts of network anomaly detection.

An autoencoder is a neural network that consists of two phases

- An *encoder* is a deterministic mapping function f_θ that transforms an input vector $\boldsymbol{x}$ into hidden representation y
  - $\theta = \{\mathbf{W}, b\}$, where $\mathbf{W}$ is the weight matrix and, b is bias
  - $f_\theta\,(x) \approx x'$
- A *decoder* reconstructs the hidden representation $\boldsymbol{z}$ (encoder's output) to x' via g_θ.

Autoencoder measures the reconstruction error between x'(reconstructed) and the input $\boldsymbol{x}$ and to minimize this error (information loss) to make x' as close as possible to $\boldsymbol{x}$.

$$J(W) = \sum ||x_n - x'_n|| \qquad (1)$$

J (W) is the cost function whose goal is to minimize the cost

Arg min $(J(W))_{\{w,w',b,b'\}}$

Where w and b are encoder weights and biases respectively, and w', b' are weights and biases for the decoder.

Various functions can be used as cost functions for example squared error. For The cost function optimization, several options are available for instance stochastic gradient descent SGD and AdamOptimizer.

Fiore et al. used a semi-supervised deep learning tool for network anomalies detection [11]. Authors introduced a discriminative form of restricted Boltzmann machines. The results were not promising specifically when testing the DRBM in a new network. Several research papers focus on improving the classical machine learning algorithms with deep learning. Salama et al. used Deep Belief Network (DBN) as a dimensionality reduction tool for Support Vector Machines (SVM) classifier [12]. The authors claimed a hybrid approach achieve approximately 93 % accuracy where the SVM and DBN scored 88 % and 90 % respectively. In another comparative study, authors compare three traditional algorithms, i.e., Bays networks, C4 and SVM against a hybrid SVM-RBM algorithm. The results showed the superiority of the hybrid method in various attack detection, e.g., DoS and user root attacks [13]. In a broader comparative study on anomalies detection, authors presented a deep structured energy-based model; The study compares their algorithms in two different decisions boundaries against five severe anomalies detection algorithms including PCA and SVM. The authors go further step by applying their algorithm to various data types, i.e., static, sequential, and spatial datasets [14]. Among the static datasets, they choose the KDD99 network dataset. Their results showed a comparable or better performance to methods like PCA and kernel PCA.

IV. DETECTION FRAMEWORK

We introduce a detection framework as a component of the control layer. Figure 1 shows the positioning of the framework; where the IDS is a module of the controller layer. This architecture provides centralization and flexibility. The integration of the system is beyond our research of this paper, as the primary goal is to investigate the algorithms.

We used Tensorflow (TF) as a deep learning development library. As the name indicates, Tensorflow is matrices flow in a graph model. TF graph consists of nodes and edges;

nodes represent mathematical operations, edges represent multi-dimensional data arrays (tensors).

Figure 2 depicts the work flow of the simulation. The first stage of the experiments is to build the Autoencoder network. The AE consists of two passes, the encoder, and the decoder. Both the encoder and decoder consist of multiple layers. The data set is loaded into Tensorflow tensor dimension (Training samples, 41). We build the weight and biases tensors for the encoder and decoder. The dimension of weighs and biases depends on the number of neurons (units in the hidden layer). For instance, if we decode the input into five units, this means we will have (41, 5) tensor where 41 is some input units (features of one network traffic record), and same dimensions will be used in the decoder. The next step is to train the network; in the forward pass, we use the logits as an activation function. Then we apply the activation function to reconstruct the record from the decoded units, weights, and biases for the output. The next step is to compare the original data against the reconstructed output. We use the cost function to

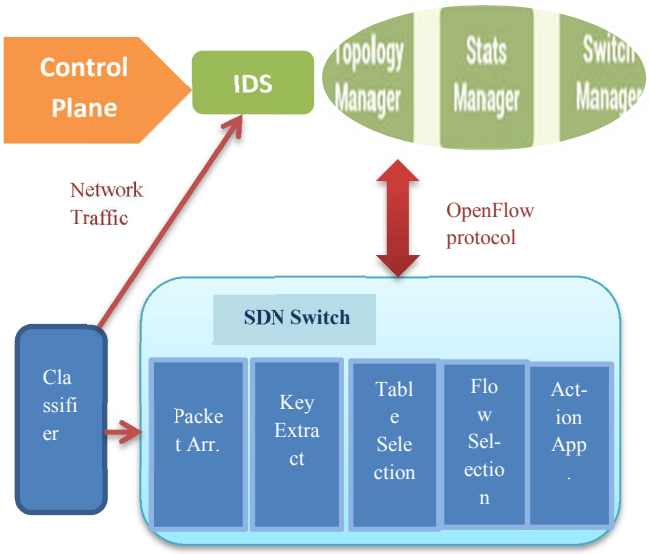

Fig.1. Proposed location of the detection system in SDN model

compute the data loss, for instance, the squared error function. The third step is to minimize the cost (in our case data loss). Several optimization algorithms are used to minimize the loss or reconstruction rate. For example, we used Adam optimizer. Once the network settles after various sweeps of data chunks (batches), the second phase testing is on.

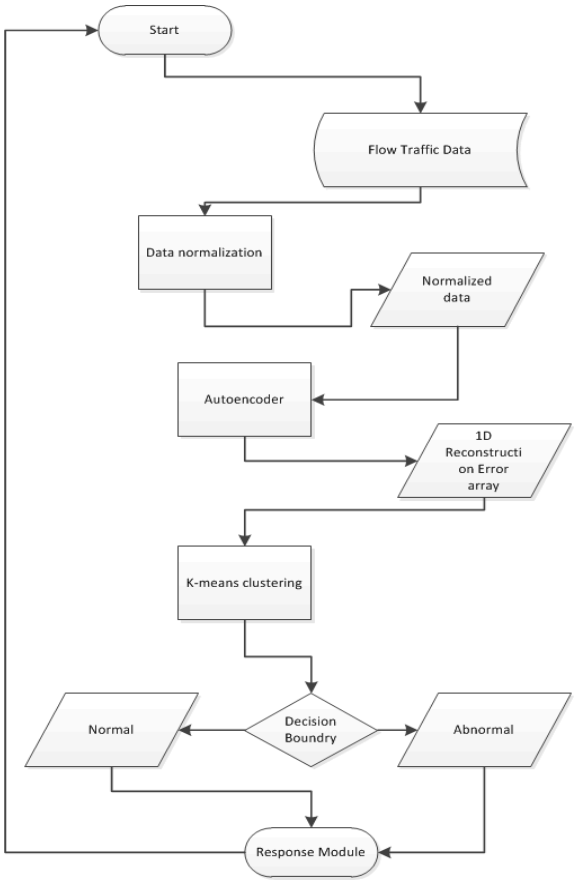

Fig.2. Framework flow diagram

During the testing, we feed the network with the testing sample and try to reconstruct the data. For clarification, if our network manipulates images, if we pass an image as an input we expect an image close enough (with minimum data loss). We used the same procedure to implement the considering model's variances.

For the anomalies detection, we measure the data loss between the input and the reconstructed record. If the loss is too high (we have to define thresholds) this means the input cannot be precise enough to be reconstructed. We consider inputs with high reconstruction error as anomalies. This concept is valid for Restricted Boltzmann machines, as both algorithms reconstruct the input.

The performance of the algorithm varies depending on various criteria.

- Type of the data, whether the input is binary or decimal.
- Activation function, for example, sigmoid works better with binaries while Relu is good for decimals.

- Cost function, for instance, squared error, and cross entropy
- Optimizer, Gradient Descent, Adam optimize, SGD (figure below shows cost optimization using two different optimizers). The autoencoder aims to minimize the reconstruction error over multiple sweeps of the input data. The y-axis represents the data loss calculated by the cost function (squared error), while the x-axis represents the data sweeps. The graph shows the loss is decreased till it reaches the minima.

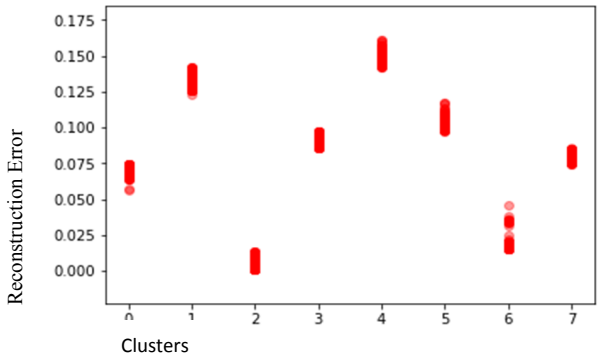

Fig.4. RE distribution in clusters

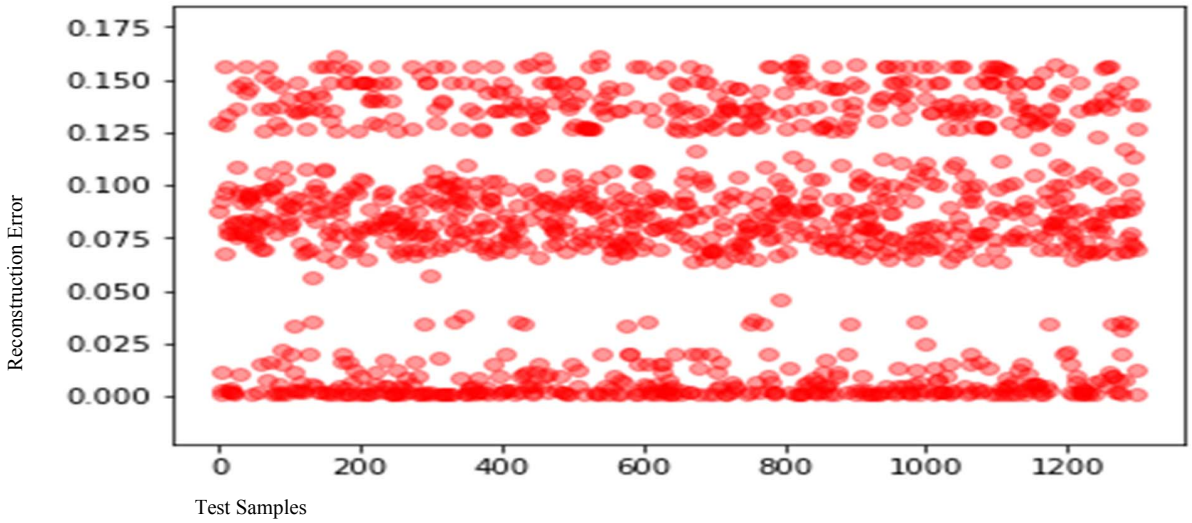

Fig.3. Test sample RE distribution

After the framework implementation, we pass various samples of different sizes. The samples contain normal and abnormal traffic. The output was clustered; in perfect results, those clusters only include normal or abnormal data. For example, table 1 shows the predicted clusters for 1300 samples.

TABLE 1. CLUSTERS PREDICTED BY THE FRAMEWORK

Cluster 1	normal	139	abnormal	3
Cluster 2	normal	4	abnormal	183
Cluster 3	normal	0	abnormal	302
Cluster 4	normal	172	abnormal	0
Cluster 5	normal	0	abnormal	150
Cluster 6	normal	100	abnormal	2
Cluster 7	normal	0	abnormal	61
Cluster 8	normal	184	abnormal	0

TABLE 2. FRAMEWORK PREDICTION WITH VARIOUS SIZES OF SAMPLES

Samples no.	Avg. Precision	Average Cluster error	Missed samples
1300	99.20699178	0.793008216	9
800	99.76851852	0.231481481	1
400	100	0	0

Figure 3 depicts the reconstruction error for each test sample. RE of similar samples are close; the AE successed in finding a pattern in the data. Figure 4 shows the cluster had been deduced by the framework. It is noticeable we have separated clusters for RE ranges. Table 2 shows the framework prediction. As a number of testing samples increase the accuracy declines; the main reason for this is the number of training samples. If the framework sees more training sample the accuracy expected to increase.

V. CONCLUSION

Deep learning algorithms achieved a breakthrough in neural networks. With a strong record of successful applications, deep learning is a promising approach for network anomalies detection. The paper showed the potential of unsupervised deep learning to enhance the security of SDN. We applied deep autoencoders to calculate a reconstruction error for network traffic records. Then we apply a K-means as clustering algorithm on REs. Our approach showed robust prediction with reasonable training data. However, further research should investigate dimensionality in traffic records, where the number of dimensions is relatively small.

REFERENCES

[1] Kreutz, D.; Ramos, F.M.V.; Esteves Verissimo, P.; Esteve Rothenberg, C.; Azodolmolky, S.; Uhlig, S., "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, Vol. 103, no. 1, pp. 14,76, Jan. 2015

[2] Open Networking Foundation (ONF), 2015. [Online]. Available: https://www.opennetworking.org/

[3] N. McKeown et al., ‘‘OpenFlow: Enabling innovation in campus networks,’’ SIGCOMM Comput. Commun. Rev., Vol. 38, no. 2, pp. 69–74, Mar. 2008.

[4] R. Klöti, V. Kotronis and P. Smith, "OpenFlow: A security analysis," *2013 21st IEEE International Conference on Network Protocols (ICNP)*, Goettingen, 2013, pp. 1-6.

[5] D. Kreutz, F. M. Ramos, and P. Verissimo,‘‘Towards secure and dependable software-defined networks,’’ in Proc.2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw., 2013, pp. 55–60.

[6] Ghorbani, Ali A., Lu, Wei, Tavallaee, Mahbod, *Network Intrusion Detection and Prevention Concepts and Techniques*, Springer US, 2010.

[7] Mudzingwa, D.; Agrawal, R., "A study of methodologies used in intrusion detection and prevention systems (IDPS)," *Southeastcon, 2012 Proceedings of IEEE*, vol., No., pp.1,6, 15-18 March 2012.

[8] V. Chandola, A. Banerjee and V. Kumar, "Anomaly Detection for Discrete Sequences: A Survey," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823-839, May 2012.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

[10] Hinton, G. E., Osindero, S., and Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554

[11] Fiore, U., Palmieri, F., Castiglione, A., De Santis, A.: Network Anomaly Detection with theRestricted Boltzmann Machine. Neurocomputing 122, 13 { 23 (2013)

[12] Salama, M.A., Eid, H.F., Ramadan, R.A., Darwish, A., Hassanien, A.E.: Hybrid Intelligent Intrusion Detection Scheme. In: Soft Computing in Industrial Applications, pp. 293{303. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)

[13] B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, Beijing, 2016, pp. 581-585.

[14] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. 2016. Deep structured energy based models for anomaly detection. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48* (ICML'16), Maria Florina Balcan and Kilian Q. Weinberger (Eds.), Vol. 48. JMLR.org 1100-1109.

[15] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.: A Detailed Analysis of the KDD CUP 99 Data Set. In: Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on, pp. 1{6 (2009). DOI 10:1109/CISDA:2009:5356528