Denoising Autoencoder with Dropout based Network Anomaly Detection

Safa Mohamed
Research Team in intelligent Machines
National Engineering School of Gabes Tunisia
Omar Ibn El Khattab, Avenue Zrig 6072
e-mail: safamohamed280@yahoo.fr

Ridha Ejbali, Mourad Zaied
Research Team in intelligent Machines
National Engineering School of Gabes Tunisia
Omar Ibn El Khattab, Avenue Zrig 6072
e-mail: ridha_ejbali@ieee.org, mourad.zaied@ieee.org

***Abstract*—A Network Intrusion Detection System (NIDS) plays an important role in ensuring information security. It helps system administrators identify and detect malicious activities in their companies. Many techniques have been devised by researchers to achieve reliable detection of anomalies. It is thus a challenging task to determine a network anomaly more accurately. To solve this problem, we propose a Denoising-Autoencoder (DAE) with a Dropout based network anomaly detection method because it forces the extraction of intrinsic features so as to increase the detection accuracy. A popular NSL-KDD dataset is used for the training and evaluation of our approach. The performance of our approach takes into consideration different metrics such accuracy, precision, recall, f-measure values and the detection rate. Experimental results show that our approach performs better than other detection methods, especially when we use a single hidden layer with 8 neurons.**

Keywords-Anomaly detection; NIDS; Denoising Autoencoder; NSL-KDD.

I. INTRODUCTION

The advent of networks offers immense services to users. Because these services are subject to several attacks and security mechanisms, it is necessary to protect them. Intrusion detection is one major research problem in network security, aiming at identifying unusual access or attacks to secure internal networks. In fact, an intrusion refers to any unauthorized access or misuse of information resources.

There are various systems designed to block attacks. We particularly cite the Network Intrusion Detection System (NIDS). A NIDS is security tools that, like other measures such as antivirus software, firewalls and access control schemes, are intended to strengthen the security of information and communication systems. It monitors and analyzes the network traffic entering into or exiting from the network devices of a company and raises alarms if an intrusion is observed [1].

Based on the methods of intrusion detection, NIDS can be classified into 2 types: Signature based NIDS (SNIDS) and Anomaly detection based NIDS (ADNIDS) [2].

The SNIDS, e.g. Snort (www.snort.org), is used to identify attacks in a form of signature or pattern. It uses the known pattern to detect attacks; the main disadvantage is that it fails to identify any unknown attacks to the network or system. In contrast, ADNIDS determines a normal network activity like the sort of bandwidth generally used, the protocols used, the ports and devices that generally connect to each other and alert the administrator or user when an anomalous (not normal) traffic is detected and it requires an understanding of what “normal” is [2]. However, they have the disadvantage of having high false positive rates, which can make the detector useless in practical areas. Analyzing and detecting anomalies is important because it reveals useful information about the characteristics of the generation process data.

Many NIDSs perform a feature selection task to extract a subset of relevant features from the traffic. Dimensionality reduction based anomaly detection method is one of the popular detection methods. It is based on the assumption that the features of normal data are correlated with each other [3]. In this respect, Principal Component Analysis (PCA) based methods belong to this method of detecting anomalies [4]. However, PCA is a linear transformation, which fails to capture the non-linear correlations between features [5].

With an increasing amount of features, the data have supplementary complicated nonlinear structures. As a solution to this weakness, Kernel Principal Component Analysis (KPCA) is used to generalize PCA to nonlinear dimensionality using techniques of kernel methods [6].

Recently, the Autoencoder (AE) is a novel dimensionality reduction method that uses unsupervised neural networks. It can find the optimal subspace, which captures the non-linear correlations between features [1]. For this reason, we propose a Denoising Autoencoder with Dropout based network anomaly detection of an extension of the basic AE and represent a stochastic version of it used to perform dimensionality reduction and force the extraction of intrinsic features. We use the NSL-KDD [7] dataset, with a separate training and testing set to evaluate their performances.

The rest of this paper is organized as follows: Section 2 presents the context of our work. In Section 3, we present our approach or methodologies. In Section 4, we present the evaluation and analyze the results. Section 5 concludes and suggests future works to be adopted later.

II. LITERATURE REVIEWS

Anomaly detection is applied in traffic detection, Card fraud detection, abnormal crowd behavior detection and network intrusion detection [8]. The widely-used anomaly detection methods can be divided into the following categories: Classification based methods, nearest neighbor-based methods, Clustering based methods, Statistic based methods and Dimensionality Reduction based methods [3][8].

Classification based methods learn a model from labeled data and then classify testing data into one of the classes using the learnt model. Nearest neighbor based methods show that normal data have relatively more neighbors than the anomalous data requiring a distance measurement to evaluate the resemblance between a testing sample and its neighborhoods. Clustering based methods group homogenous data into one cluster and suppose the outliers far away from their closest cluster center. Statistics based methods shape well a statistical model using the given training data and then apply a statistical inference to decide whether an unseen instance is an outlier or not. Dimensionality reduction based methods utilize the reconstruction error to classify the anomalies. PCA are an effective preprocessing method before anomaly detection [9].

Lakhina et al. [10] proposed a new hybrid algorithm; Principal Component Analysis Neural Network Algorithm (PCANNA) is used to reduce the number of computer resources, both memory and CPU time required to detect attacks. Ibraheem et al. [11] presented an intrusion detection model based on PCA and MLP to recognize an attack from normal connections. Ikram et al. [12] developed an intrusion detection model by using PCA as the dimensionality reduction technique and SVM as the classifier. Elkhadir et al. [6] compared the performance between (PCA) and (KPCA) in order to construct robust IDS with the highest anomaly detection rate. Experimental results showed that KPCA are more efficient than PCA. Paula et al. [13] proved that the AE can detect delicate anomalies and linear PCA fails to detect without corrupting the quality of the detecting performance. Sakurada et al. [4] proposed a comparison between the use of AE, PCA and KPCA in the anomaly detection task. Experimental results showed that AE is the most efficient and it can increase their accuracy by extending them to DAE.

For improved dimensionality reduction and better detection rate, we propose to use DAE with a Dropout that makes more objective and principled anomaly score than the reconstruction error of PCA and KPCA based method.

III. Proposed Methodologies

In this section, we present the different steps followed to reach our approach.

A. *AE to DEA*

AE is a specific type of feedforward neural networks where the input is the same as the output. AE aims to learn a compressed representation of data with minimum reconstruction loss [14]. It consists of 3 components: encoder, code and decoder [15]. The encoder compresses the input and produces the code; the decoder then reconstructs the input only by using this code (see Figure 1).

Keeping the code layer forced our AE to learn an intelligent representation of the samples. There is another way to force the AE to learn useful features. It is adding random noise to its inputs and making it recover the original noise-free data. This way the autoencoder can’t simply copy the input to its output because the input also contains random noise. We order it to subtract the noise and produce the underlying meaningful data. This is called a DAE [9] (see Figure 2).

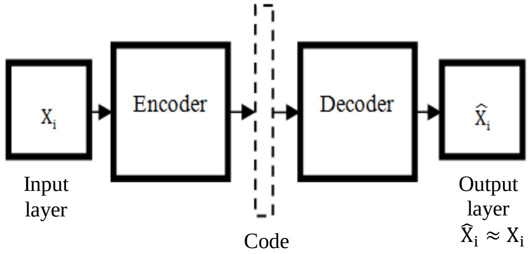

Figure 1. Autoencoder.

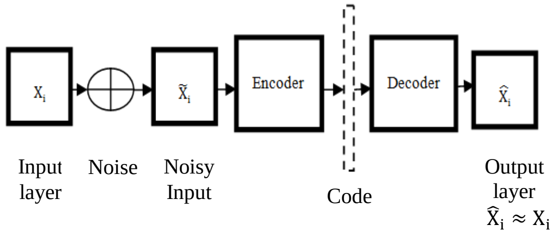

Figure 2. Denoising Autoencoder.

B. *DEA with Dropout- Based Anomaly Detection*

“Dropout” is a technique that aims to discourage brittle co- adoptions of hidden unit feature detectors. It can also be interpreted as a way of regularizing a neural network by adding noise to its hidden units [16]. The choice of which units to drop is random. In the simplest case, each unit is retained with a fixed probability p independent of other units, where p can be chosen using a validation set or can simply be set at 0.5 [17]. In our method, the Dropout (noise) is applied to the input layer of the Denoising Autoencoder.

We propose using a DEA with Dropout based anomaly detection method for only intrusion detection that is a deviation base anomaly detection method whose training here only contains instances for the normal instances of traffic without labeling. It uses the reconstruction error as the anomaly score. Our NSL-KDD dataset used consists of different steps such as the Numericalization and Normalization. These two steps were performed for both NSL-KDD train and test datasets. Later, the train dataset is used to train the DEA with Dropout. Our method is tested with test dataset and the results were analyzed (see Figure 3). The detailed development process is provided in the following sub-section.

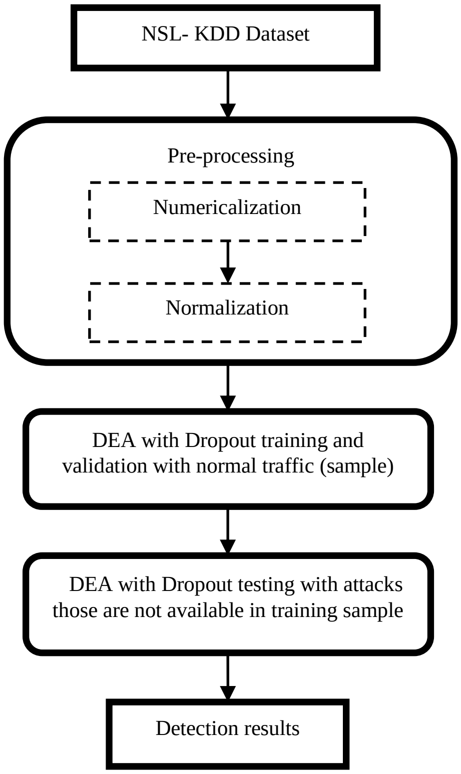
NSL- KDD Dataset
Pre-processing
Numericalization
Normalization
DEA with Dropout training and validation with normal traffic (sample)
DEA with Dropout testing with attacks those are not available in training sample
Detection results

Figure 3. Denoising Autoencoder with Dropout-based anomaly detection method.

C. NSL-KDD Dataset

NSL-KDD is an improved and reduced version of the KDD Cup 99 dataset. The KDD Cup dataset was prepared using the network traffic captured by 1998 DARPA IDS evaluation program [13]. It is continuously an index which is used to compare the NIDS models in common researches [7]. In the latest literature, all the researchers use the NSL-KDD as the benchmark dataset [18]. It includes 125,973 network traffic samples in the KDDTrain+ Dataset and 22,554 network traffic samples in the KDDTest+ Dataset. In each record, there are 41 attributes unfolding different features of the flow and a label is assigned to each sample either as an attack type or as a normal type. The features include 10 basic features (1- 10), 12 content features (11 - 22), and 18 traffic features (23 -.41) as shown in (Table I).

Apart from normal data, records for 39 different attack types exist in NSL-KDD dataset. All these attack types were grouped into four attack classes:

- **DOS (Denial of Service):** an attacker tries to prevent legitimate users from using a service.
- **Probe:** an attacker tries to find information about the target host.
- **U2R (User to Root):** an attacker has local account on victim's host and tries to gain the root privileges
- **R2L (Remote to Local):** an attacker does not have local account on the victim host and try to obtain it.

The summary of the attack classes and their attack types is given in (Table II).

TABLE I. FEATURES IN NSL-KDD [19]

Type	Features
Nominal	2,3,4
Binary	7,12,14,15,21,22
Numeric	1,5,6,9,10,11,13,16,17,18,19,20,23,24,25,26,27,28, 29,30,31, 32, 33,34,35,36,37,38,39,40,41

TABLE II. ATTACK TYPES IN NSL –KDD DATASET [20]

Attack Class	Training Set	Testing Set
DOS	Back, Land, Neptune, Pod, Smurf, , Teardrop	Back, Land, Neptune, Pod, Smurf Teardrop, **mailbomb, Apache 2, Udpstorm, Processtable, Worm**
R2L	Guess_Password, Ftp_write, Imap, Phf, Multihop, Waremaster, Warezclient, Spy	Guess_Password, Ftp_write, Imap, Phf, Multihop, Waremaster, Spy, **Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendfmail, Named**
U2R	Buffer_overflow, Loadmodule, Perl, Rotkit	Buffer_overflow, Loadmodule, Rotkit, Perl, **Sqlattack, Xterm, Ps**
PROBE	Satan, Ipsweep, Nmap, Portsweep	Satan, Ipsweep, Nmap, Portsweep, **Mscan, Saint**

D. Pre-processing

Before proceeding to experimental work, the NSL-KDD data sets first went through a data preprocessing operation and attribute a type of conversion by following the steps described in the following part:

1) Numericalization: The features 2, 3 and 4 namely the protocol_ type, service and flag were non-numerical. The input value of the Denoising AE should be a numeric matrix. We must convert these features into numeric form in the train and test data set. 'tcp','udp'and 'icmp'and its numeric values are encoded as binary vectors (1, 0, 0), (0, 1, 0) and (0, 0, 1). Similarly, the feature 'service' has 70 types of attributes, and the feature 'flag' has 11 types of attributes. Continuing in this way, we obtain a 41 dimensional feature map into 122 dimensional features after transformation.

2) Normalization: The values obtained after the operation of the numericalization are very varied and constitute a big interval. Some attributes take great values (1, 5 and 6) (duration, src_bytes, dst_bytes) while others take only small values. To help them accord to same features, we apply the logarithmic scaling method. Finally, the value of every feature is mapped to the [0, 1] using min-max where max denotes the maximum value and min denotes minimum value for each feature.

$$x_i = \frac{x_{i-min}}{max-min} \qquad (1)$$

E. Methodology

In this approach, we implemented a DEA with Dropout on the inputs. It consists of an input layer of 122 neurons due to the fact that the number of features for each sample is 122 followed by a Dropout layer with a fixed probability p= 0.5 and a single hidden layer with different number of neurons such as (8, 16, 24 and 32) units so the hidden representation of the autoencoder has a compression ratio of 122 to (8, 16, 24 o and 32) forcing it to learn interesting patterns and relations.

Finally, there is an output layer of 122 units; the activation of both the hidden layer and the output layer is the "Relu" function.

The training set has 125973 rows, but the DEA was trained using only the samples labeled "Normal" to capture the nature of normal behavior , and this was accomplished by training the model to minimize the mean squared error between its output and its input. We use 67343 samples labeled "Normal" with 60608 are used for training. The model is trained for 20 epochs using an Adam optimizer with a batch size of 150. Furthermore, we held out 6735 for validation that refer to 10% of the normal training samples to validate the model.

IV. Evaluation and Result Analysis

The model performs anomaly detection by calculating the reconstruction error of samples since the model was trained using normal data samples. Only the reconstruction error of samples that represent attacks should be relatively high compared to the reconstruction error of normal data samples. This intuition allows us to detect attacks by setting a threshold for the reconstruction error. If a data sample has a reconstruction error higher than the preset threshold then the sample is classified as an attack. Otherwise, it's classified as normal traffic.

A. Evaluation Based on Training and Data Validation

For the choice of a threshold, two values can be helpful to guide the process. Concerning the model loss over the training data and over the validation data, we found by experiment that a choice around these values produces acceptable results. For our experiments, we use the model loss over the training data as a threshold.

In Table III, we present the val_loss for 3 epochs using a single hidden layer with different neurons.

TABLE III. Val_ Loss in 3 Epochs

Single hidden layer	**Epoch 1/20**	**Epoch 2/20**	**Epoch 3/20**
32 neurons	loss: 0.0334 val_loss: 0.014	loss: 0.0128 val_loss: 0.0094	loss: 0.0102 val_loss: 0.0077
24 neurons	loss: 0.0316 val_loss: 0.0133	loss: 0.0124 val_loss: 0.0091	loss: 0.0101 val_loss: 0.0075
16 neurons	loss: 0.0335 val_loss: 0.0160	loss: 0.0139 val_loss: 0.0103	loss: 0.0096 val_loss: 0.0073
8 neurons	loss: 0.0339 val_loss: 0.0184	loss: 0.0160 val_loss: 0.0127	loss: 0.0129 val_loss: 0.0107

B. Evaluation Based on Test Data

In the section, we evaluate the performance over the test dataset which includes 22543 rows, 37 different attacks and one normal label that refers to 12832 for normal samples and 9711 for attack samples. The calculated losses are a helper function that accepts the original features and the predicted features and relies on the reconstruction loss of each data sample. Afterwards, each data sample is classified according to its reconstruction error and the preset threshold.

The nature of this approach is purely for anomaly detection. We evaluate the performance of DEA based anomaly detection on the following metrics

- Accuracy (A): Defined as the percentage of correctly classified records over the total number of records.

$$\text{A}= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (2)$$

- Recall (R): Defined as the % ratio of number of true positives records divided by the sum of true positives and false negatives (FN) classified records.

$$R = \frac{\text{TP}}{(\text{TP+FN})} \times 100\% \qquad (3)$$

- Precision (P): Defined as the % ratio of the number of true positives (TP) records divided by the sum of true positives (TP) and false positives (FP) classified.

$$\text{P} = \frac{\text{TP}}{(\text{TP+FP})} \times 100\% \qquad (4)$$

- F-measure (F): The harmonic average F combines recall and precision in a number between 0 and 1.

$$F = \frac{2.P.R}{(P+R)} \qquad (5)$$

TABLE IV. EVALUATION METRICS

Single hidden layer	Accuracy	Recall	Precision	F-measure
32 neurons	89.13%	94.07%	87.72%	90.78%
24 neurons	89.65%	95.66%	87.36%	91.32%
16 neurons	89.90%	**96.61%**	87.07%	91.59%
8 neurons	**90.32%**	95.04%	**88.12%**	**91.85%**

From Table IV, we can see that our results have demonstrated that our approach offers high levels of accuracy, recall, precision and F-measure especially when we use a little number of neurons (8 neurons) in a hidden layer. In addition, in the 4 metrics, our method is evaluated according to a Detection rate (see Table V).

TABLE V. DETECTION RATE

Single hidden layer	Normal	DOS	R2L	U2R	PROBE
32 neurons	17.40%	92.48%	93.72%	77.61%	**99.95%**
24 neurons	18.29%	93.07%	99.11%	**100%**	99.87%
16 neurons	18.95%	94.60%	**99.26%**	**98.50%**	99.91%
8 neurons	**22.27%**	**95.80%**	**98.85%**	**100%**	99.91%

Table V illustrates the detection rate for every type of attacks (DOS, U2R, R2L and PROBE) and normal data. The process of detecting anomalies using our Denoising autoencoder with dropout method produced a high detection rate. We can see that for testing data, U2R attack is detected with a rate of 100% using 8 and 24 neurons in a hidden layer. Also, we can note that DOS and PROBE attacks are highly detected with a rate of 95.80% (8 neurons) and 99.95 %.(32 neurons). R2L is also well identified as attacks with 99.26% (16 neurons). In contrast, the normal data are not well detected with a maximum rate of 22.27%.

These detection rates were better that the results produced by Elkhadir et al. [6] when using PCA and KPCA for detection of anomalous connection in NSL- KDD dataset (see Table VI).

TABLE VI. ATTACK'S DETECTION RATE OF PCA AND KPCA [6]

Method	DOS	R2L	U2R	PROBE
PCA	90.35%	93.6%	87.2%	85.15%
KPCA	90.2%	92.6%	87.25%	85.45%

Finally, according to the 5 metrics previously mentioned to evaluate the performance of DEA based anomaly detection; the best result is obtained when we used a single hidden layer with 8 neurons.

V. CONCLUSION AND FUTURE WORK

In this approach, we attempted to develop a Denoising Autoencoder with Dropout-based network anomaly detection method for improving intrusion detection. This method was trained only using normal traffic. The strength of this approach is its simplicity. It consists of only a single hidden layer with different neurons making it very easy to train. In terms of detection rates, our approach outperforms many methods in the existing literature.

In future work, we can build and evaluate a model with many hidden layers.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support of this work by grants from General Direction of Scientific Research (DGRST), Tunisia, under the ARUB program.

REFERENCES

[1] A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," In Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS), (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 21-26, 2016.

[2] B. C. Rhodes, J. A. Mahaffey, and J. D. Cannady, " Multiple self-organizing maps for intrusion detection," In Proceedings of the 23rd national information systems security conference, pp. 16-19, 2000.

[3] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," Wireless Telecommunications Symposium (WTS), pp. 1-5, 2018.

[4] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, ACM, 2014.

[5] S. Heni, R. Ejbali, M. Zaied, and C. B. Amar, "A Neural Principal Component Analysis for text based documents keywords extraction,"3rd International Conference on Next Generation Networks and Services (NGNS), pp. 112-115, 2011.

[6] Z. Elkhadir, K. Chougdali, and M. Benattou, "Intrusion Detection System Using PCA and Kernel PCA Methods," IAENG International Journal of Computer Science, 2016.

[7] S. Revathi, and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," International Journal of Engineering

Research & Technology (IJERT), vol. 2, no 12, pp. 1848-1853, 2013.

[8] D. Hou, Y. Cong, G. Sun, J. Liu, and X.Xu, "Anomaly detection via adaptive greedy model," Neurocomputing, vol. 330, pp. 369-379, 2019.

[9] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," vol.11, no Dec, pp. 3371-3408, 2010.

[10] S. Lakhina, S. Joseph, and B. Verma, "Feature Reduction Using Principal Component Analysis for Effective Anomaly–Based Intrusion Detection on NSL-KDD," International Journal of Engineering Science and Technology ,Vol. 2(6), pp.1790-1799, 2010.

[11] N. B. Ibraheem , M. M. Jawhar, and H. M. Osman, "Principle Components Analysis and Multi Layer Perceptron Based Intrusion Detection System," AL-Rafidain Journal of Computer Sciences and Mathematics, vol. 10, no 1, pp.127-135, 2013

[12] S.T Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using PCA and optimized SVM," Journal of computing and information technology, vol. 24, no 2, pp. 133-148, 2016.

[13] E. L. Paula, M. Ladeira, R. N. Carvalho, and T.Marzagao, "Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering," In 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 954-960, 2016.

[14] M. Gnouma, A. Ladjailia, R. Ejbali, and M. Zaied, "Stacked .sparse autoencoder and history of binary motion image for human activity recognition," Multimedia Tools and Applications, vol. 78, no 2, pp .2157-2179, 2019.

[15] S. Said et al. ,"Deep wavelet network for image classification," IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016.

[16] A. ElAdel, R. Ejbali, M. Zaied, and C. B. Amar, "Fast deep neural network based on intelligent dropout and layer skipping," IEEE International Joint Conference on Neural Networks (IJCNN), pp. 897-902, 2017.

[17] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8610-8613, 2013D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[18] Y. Ding and Y. Zhai, "Intrusion Detection System for NSL-KDD Dataset Using Convolutional Neural Networks," In Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence, pp. 81-85, 2018.

[19] C.YIN, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks, " Ieee Access, vol. 5, pp. 21954-219, 2017.

[20] 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," IEEE International Conference on Wireless Networks and Mobile Communications (WINCOM), 2016.