International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 07 | July 2020 www.irjet.net p-ISSN: 2395-0072

Network Intrusion Detection using Recurrent Neural Network Algorithm

Mr. Gunjal Somnath P[1], Prof. Aher.S.M[2]

[1]*Student of Computer Engineering, VACOE' College of Engg. Ahmednagar, India*
[2]*HOD, Dept. of Computer Engineering, VACOE' College of Engg. Ahmednagar, India*

Abstract - *Internet is a widely used platform nowadays by people across the word. This has led to the advancement in science and technology. Many surveys conclude that network intrusion has registered a consistent increase and lead to personal privacy theft and has become a major platform for attack in the recent years. Network intrusion is unauthorized activity on a computer network. Hence there is a need to develop an effective intrusion detection system. In proposed system acquaint an intrusion detection system that uses improved recurrent neural network(RNN) to detect the type of intrusion. In proposed system also shows a comparison between an intrusion detection system that uses other machine learning algorithm while using smaller subset of kdd-99 dataset with thousand instances and the KDD-99 dataset.*

***Key Words*: Intrusion detection, Feature selection, linear correlation coefficient, deep learning, RNN**

1. INTRODUCTION

Internet has become part of daily life and essential tool today. Along with its boons, the internet has given rise to many vices. This has led to an increase in the number of attacks. These attacks may affect individuals as well as organizations. Therefore, the security of computer and network systems has been in the focal point of research for a long time. All organizations working in the field of information technology have been agreed that the subject of information protection is very critical and important issue that cannot be ignored. It is necessary to achieve the three basic principles that any secure system rests on its (confidentiality, integrity, and availability). The National Institute of Standards and Technology has defined intrusion detection as "the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network" [1],[2]. IDS detect intruder's actions that threaten the confidentiality, availability and integrity of resources. IDSs can be used on detect difference types of malicious network communications and computer systems usage, whereas the conventional firewall can not perform this task. Intrusion detection is based on the assumption that the behavior of intruders is different from legal user [3]

In general IDSs can be divided into two groups: 1) anomaly 2) misuse (signature) detection based on their detection approaches [4]. In Anomaly detection, the system classifies unknown or unusual behavior in network traffic by studying the structures of normal behavior in network traffic. Network traffic that deviates from a normal traffic pattern is classified as an intrusion. In Misuse (signature) detection, attack signatures are pre-installed in the IDS. A pattern matching is performed for the traffic against the installed signatures to detect an intrusion in the network [5]. The current situation will reach a point whereby reliance on such techniques leads to ineffective and inaccurate detection

In recent years, one of the main focuses within IDS research has been the application of machine learning and shallow learning techniques such as Naive Bayes, Decision Trees and Support Vector Machines (SVM) [6]. The application of these techniques has offered improvements in detection accuracy. However, there are limitations with these techniques, such as the comparatively high level of human expert interaction required; expert knowledge is needed to process data Similarly, a vast quantity of training data is required for operation (with associated time overheads), which can become challenging in a heterogeneous and dynamic environment [7] .To address the above limitations, a research area currently has switched towards deep learning. Deep learning is an advanced subset of machine learning, which can overcome some of the limitations of shallow learning. Deep learning is an advance machine learning techniques where there are multiple information-processing layers in hierarchical architectures which are utilized for classifying patterns and for feature or representation learning [8]. Today, deep learning has become a very important and successful research trend in the ML community because of its great success in these fields [9].In this paper, proposes a deep learning version to enable IDS operation inside modern networks.

2. REVIEW OF LITERATURE

Fahimeh Farahnakian et al. proposed a Deep Auto Encoder (DAE) model which is trained in a greedy layer-wise fashion in order to avoid overfitting and local optima. Their suggested Deep Auto Encoder based IDS (DAE-IDS) is made

up of four auto encoders, in which the result of the AE at the existing layer is utilized as the AE input in the following layer. Moreover, an AE at the existing layer is trained prior to the AE at the following layer. After the 4 auto-encoders are trained, they have utilized a SoftMax layer for classifying the inputs to normal and attack. They have utilized the KDDCUP 1999 data-set for evaluating the efficiency of DAE-IDS due to the fact that this data-set has been used largely for the evaluation of the IDSs. The suggested method has reached a detection precision equal to 94.71% on the total of 10% KDD-CUP 1999 testing data-set [1].

Ni GAO et al. suggested an approach which has been based on the multilayer DBN for the DoS attacks detection. DBN consists of numerous RBMs. Here in advance in the learning process, the training of the RBM is carried out. Then the trained features of RBM are used as an input data for learning RBM of the next layer of the DBN stack. The effectiveness of the DBN method is tested on the KDD CUP 1999 data set. The detection precision of the DBN model had shown to be better than the SVM and ANN methods [2].

Sanghyun Seo et al. study compared the rates of intrusion detection between the NIDS with the use of only a classification model and the NIDS trained with data where noise and outliers are eliminated with the use of the RBM. Noise and outliers in KDD Cup '99 Data are eliminated via applying the data to RBM and constructing new data. The study proposed a training approach for classification models to be capable of detecting network intrusions with the use of the data that has been reconstructed based on those RBM features [3].

Khaled Alrawashdeh et al. considered a method of deep learning for detecting anomalies with the use of an RBM and a deep belief network. Their approach made use of a 1-hidden layer RBM for performing unsupervised reduction of features. The resulting weights from this RBM are passed to some other RBM that produces a deep belief network. The pretrained weights are passed to a fine tuning layer that consists of a Logistic Regression (LR) classifier that has multiclass soft-max. Their architecture has performed better than previous approaches of deep learning that have been implemented by Li and Salama [23], [24] in accuracy and speed of detection. They achieved a detection rate equal to 97.9% on the total 10% KDD-CUP 1999 testing data-set. As a future extension, they suggested applying their ML strategy on larger and more challenging data-sets that included wider range of attacks [4].

Jihyun Kim et al. constructed a model for IDS with deep learning method. They have applied Long ShortTerm Memory (LSTM) architecture to an RNN and have trained their IDS with the use of the KDDCup-99 data-set. For the stage of training, they have produced a data-set via the extraction of samples from the KDDCup-99 data-set by comparing it with other IDS classifiers; they have discovered that the attacks are efficiently detected via LSTM-RNN classifier. Due to the fact that they have the best accuracy and Detection Rate although the Rate of False Alarms is a little bit above the others. Through the performance tests, they have confirmed that the method of deep learning is sufficient for the IDS [5].

Yin Chuan-long et al. [6], [7] presented the design and implementation of the detection system based on recurrent NNs. In addition to that, they have investigated the model efficiency in binary and multi-class classifications, the number of neurons and various learning rate effects on the precision. On the other hand, they have investigated the efficiency of the na¨ıve Bayes, multi-layer perceptron, random forest, SVMs and other approaches of ML in multi-class classification on the benchmark KDD-Cup 1999 dataset, and they have performed a comparison of the efficiency of the RNN-IDS with other approaches of ML both in binary and multi-class classifications.

The research of Tuan Tang et al. proposed a Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) which has enabled IDSs for SDNs. The presented method has been tested with the use of the KDDCup-99 data-set, and they have accomplished a precision equal to 89% with only 6 raw features. Their experimental results have also shown that the presented GRU-RNN doesn't degrade the performance of the network. Their approach has utilized the smallest number of features when compared with other conventional methods. And that raises the computational efficiency of the model for real time detection. Moreover, the evaluation of the efficiency of the network has shown that their method doesn't considerably impact the efficiency of the controller. This work might be further enhanced by optimizing the model and using other features for the aim of increasing the accuracy. It is also possible to attempt to implement their method in a distributed manner for the sake of reducing the overhead on the controller [8].

YAO Yu et al proposed a method of anomaly intrusion detection which is based on Hybrid MLP/CNN (Multilayer Perceptron/Chaotic NN). A hybrid MLP/CNN NN is generated with the aim of improving the detection rate of time-delayed attacks. The simulation tests have been conducted with the use of the DARPA 98 data-set. The hybrid MLP/CNN NN model takes the result from the MLP as a chaotic neuron input in a way that chaotic neurons number has to be equivalent to the number of output nodes of the MLP. When the result of the classification of an input is analyzed by MLP, it may be forwarded and retained by the CNN which is connected to the MLP output node. They have realized classification with memory of anomaly events with the use of the real-time MLP classification and the memorial CNN functionality Due to the hybrid NN has flexible time-delay criterion and capability; it can achieve high rates of intrusion detection and low rate of false alarms. The method has a considerable potential of high scalability and the ability

of recognizing new patterns of attacks by the detection of the BSM strings [9].

Kehe Wu et al. proposed a NIDS model utilizing CNNs. They have CNN to select traffic features from raw dataset automatically, and set the cost function weight coefficient of each class based on its numbers to solve the imbalanced dataset problem. The model not only reduces the false alarm rate (FAR) but also improves the accuracy of the class with small numbers. To reduce the calculation cost further, they have converted the raw traffic vector format into image format. They have utilized the original KDDCup-99 data-set for evaluating the efficiency of the suggested CNN model. The experimental results have shown that the precision, FAR and computational cost of the presented model has a better performance compared to the conventional standard algorithms. More improvements can be made for the detection accuracy of this work. It is possible modifying the CNN model structure for the sake of achieving the goal. In addition to that, due to the fact that the detection time is also key to intrusion detection, it is necessary to ensure that the model is capable of meeting the time requirements of the IDS when enhancing the accuracy of detection [10].

Jin Kim et al. proposed utilizes the DNN model for effectively detecting attacks. They have utilized the popular KDDCup 1999 data-set for intrusion detection for testing and training. The testing data has been created via data pre-processing and extraction of samples in order to meet the aim of the study. A DNN model which consists of 4 hidden layers and 100 hidden units has been utilized for the proposed IDS of the presented study as its classification algorithm and utilized the ReLU function as the activation function of the hidden layers. In addition to that, this study utilized the adaptive moment (Adam) optimizer, a stochastic approach of optimization for DNN learning. The results showed a considerably high precision and detection rate, which has reached approximately 99%. Moreover, the FAR has reached approximately 0.08% [11].

Tuan A Tang et al. proposed a deep learning approach for flow-based anomaly detection in an SDN environment. They have built a Deep Neural Network (DNN) model for an intrusion detection system and train the model with the NSLKDDDataset. In the work they have proposed, they have utilized only 6 main characteristics (which can easily be obtained in an SDN environment) taken from the 41 features of NSLKDD Data-set. Through the experimental work, they have discovered an optimal hyper-parameter for DNN and confirmed the rates of detection and the false alarms. The model has reached the efficiency with a precision of approximately 75.75% which is rather reasonable from merely utilizing 6 main network features. As a future work, they have proposed implementing this method in a real SDN environment with real network traffic and evaluated the efficiency of the entire network according to latency and throughput [12].

3. PROPOSED METHODOLOGY

3.1. Data Collection

Data collection is the first and a critical step to intrusion detection. The type of data source and the location where data is collected from are two determinate factors in the design and the effectiveness of an IDS. To provide the best suited protection for the targeted host or networks, this study proposes a network-based IDS to test our proposed approaches. The proposed IDS runs on the nearest router to the victim(s) and monitors the inbound network traffic. During the training stage, the collected data samples are categorized with respect to the transport/Internet layer protocols and are labeled against the domain knowledge. However, the data collected in the test stage are categorized according to the protocol types only.

3.2. Data Preprocessing

The data obtained during the phase of data collection are first processed to generate the basic features such as the ones in KDD Cup 99 dataset. This phase contains three main stages shown as follows.

3.2.1 Data transferring

The trained classifier requires each record in the input data to be represented as a vector of real number. Thus, every symbolic feature in a dataset is first converted into a numerical value. For example, the KDD CUP 99 dataset contains numerical as well as symbolic features. These symbolic features include the type of protocol (i.e., TCP, UDP and ICMP), service type (e.g., HTTP, FTP, Telnet and so on) and TCP status flag (e.g., SF, REJ and so on). The method simply replaces the values of the categorical attributes with numeric values.

3.2.2 Data normalization

An essential step of data preprocessing after transferring all symbolic attributes into numerical values is normalization. Data normalization is a process of scaling the value of each attribute into a well-proportioned range, so that the bias in favor of features with greater values is eliminated from the dataset. Data used in Section 5 are standardized. Every feature within each record is normalized by the respective maximum value and falls into the same range of [0-1]. The transferring and normalization process will also be applied to test data. For KDD Cup 99 and to make a comparison with those systems that have been evaluated on different types of attacks we construct five classes. One of these classes contains purely the normal records and the other four hold different types of attacks (i.e., DoS, Probe, U2R, R2L), respectively.

3.2.3 Feature selection

Even though every connection in a dataset is represented by various features, not all of these features are needed to build an IDS. Therefore, it is important to identify the most informative features of traffic data to achieve higher performance. In the previous section using Algorithm 1, a flexible method for the problem of feature selection. However, the proposed feature selection algorithms can only rank features in terms of their relevance but they cannot reveal the best number of features that are needed to train a classifier. Therefore, this study applies the same technique proposed in to determine the optimal number of required features. To do so, the technique first utilizes the proposed feature selection algorithm to rank all features based on their importance to the classification processes. Then, incrementally the technique adds features to the classifier one by one. The final decision of the optimal number of features in each method is taken once the highest classification accuracy in the training dataset is achieved. The selected features for all datasets, where each row lists the number and the indexes of the selected features with respect to the corresponding feature selection algorithm. In addition, for KDD Cup 99, the proposed feature selection algorithm is applied for the aforementioned classes.

3.2.3.1 Module 1:

Input Dataset The input dataset is NSL-KDD dataset. It contains Normal, Probe, U2R, R2L and DoS attacks. Since the NSL-KDD dataset was retrieved unlabeled data, one of the first important step to add columns headers to it. The total 41 columns headers are added that contain information such as duration, protocol type, service, src bytes, dst bytes, flag, land, wrong fragment, etc. The classification of attacks are given as:

• Denial of Service Attacks: In a Denial of Service Attacks (DoS), the attacker tries to render a resource or system feature unusable by legitimate users by making it too busy with false requests. There are different types of Denial of Service Attacks. Some attacks try to exploit bugs in network software and protocol stack by sending malformed packets. The remote access is sufficient to perform Denial of Service Attacks. The examples are back, ping of death, smurf, Neptune, teardrop etc.

• Probes: The probes do not cause any damage by themselves but they provide valuable which can be used later to launch an attack. The attacker tries to search for valid IP addresses, services running on each machine or for known vulnerabilities. The examples of probes and probing tools are ipsweep, mscan, nmap, saint, Satan etc. • Remote to user: In remote to user attack, the attacker has remote access to the system but not local access. The attacker tries to exploit some vulnerability in the system to gain local access. The vulnerabilities include buffer overflows in network server software, weakly configured and misconfigured systems. The examples of remote to user attacks are dictionary attacks, guest login, ftpwrite, sshtrojan, httptunnel etc.

• User to root: In user to root, the attacker has local access to the system. The attacker tries to exploit some vulnerability in the system to gain superuser access. The common vulnerability is the buffer overflow and other vulnerabilities are bugs in management of temporary files and race conditions. The examples are eject, loadmodule, casesen, anypw, yaga etc

3.2.3.2 Module 2:

Data Preprocessing The data should be preprocessed to increase the efficiency of the system. Instead of giving direct input data, the raw data is preprocesses to avoid some issues i.e. detection rate ratio, false alarm, training overhead. For example, consider a one single vector from dataset At the time of preprocessing, the presence of comma ',' and other symbolic characters (tcp, ftp data and SF etc.) are removed. The last word gives the information about the class i.e. normal or anomaly. Data normalization is a process of scaling the value of each attribute into a well-proportioned range, so that the bias in favor of features with greater values is eliminated from the dataset. To identify the most informative features of traffic data to achieve higher performance. The proposed feature selection algorithms can only rank features in terms of their relevance but they cannot reveal the best number of features that are needed to train a classifier

3.2.3.3 Module 3:

Classifier Training Once the optimal subset of features is selected, this subset is then taken into the classifier training phase where LS-SVM is employed. Since SVMs can only handle binary classification problems and because for NSL KDD Dataset five optimal feature subsets are selected for all classes, five LS-SVM classifiers need to be employed. Each classifier distinguishes one class of records from the others. For example the classifier of Normal class distinguishes Normal data from non-Normal (All types of attacks). The DoS class distinguishes DoS traffic from non-DoS data (including Normal, Probe, R2L and U2R instances) and so on. The five LS-SVM classifiers are then combined to build the intrusion detection model to distinguish all different classes.

3.2.3.4 Module 4:

Attack Recognition The classifier is trained using the optimal subset of features which includes the most correlated and important features, the normal and intrusion traffics can be identified by using the saved trained classifier. The test data is then directed to the saved trained model to detect intrusions. Records matching to the normal class are considered as normal data, and the other records are reported as attacks. If the classifier model confirms that the record is abnormal, the subclass of the abnormal record (type of attacks) can be used to determine the record's type. Output as normal or anomaly (detection accuracy, false positive rate, reduce detector generation time).

A. Architecture

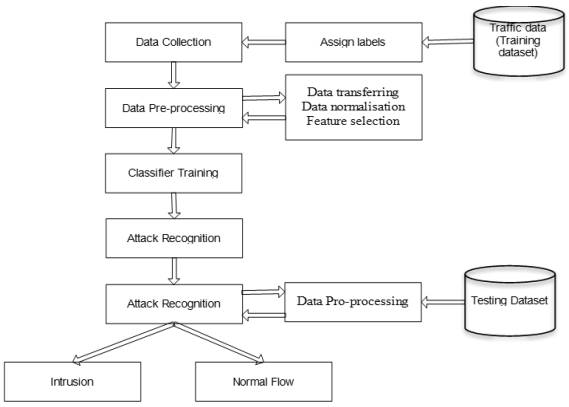

Fig. 1. Proposed System Architecture

B. Algorithms

Recurrent Neural Network

Recurrent Neural Network(RNN) are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

Steps:

Suppose there is a deeper network with one input layer, three hidden layers and one output layer. Then like other neural networks, each hidden layer will have its own set of weights and biases, let's say, for hidden layer 1 the weights and biases are (w1, b1), (w2, b2) for second hidden layer and (w3, b3) for third hidden layer. This means that each of these layers are independent of each other, i.e. they do not memorize the previous outputs.

- A single time step of the input is provided to the network.
- Then calculate its current state using set of current input and the previous state.
- The current ht becomes ht-1 for the next time step.
- One can go as many time steps according to the problem and join the information from all the previous states.
- Once all the time steps are completed the final current state is used to calculate the output.
- The output is then compared to the actual output i.e the target output and the error is generated.
- The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained.

C. Mathematical Model

x1 is a sample from the training distribution for the RNN ∈ is a learning rate for the stochastic gradient descent in Contrastive Divergence W is the RNN weight matrix, of dimension (number of hidden units, numbr of inputs)
b is the RNN offset vector for input units
c is the RNN offset vector for hidden units
Notation: Q(h2i = 1|x2) is the vector with elements Q(h2i = 1|x2)
Step 1: for all hidden units i do
Step 2: compute $Q(h_{1i} = 1|x_1)$ (for binomial units, $sigm(c_i + \sum_j W_{ij}x_{1j})$)
Step 3: sample $h_{1i} \in \{0, 1\}$ from $Q(h_{1i}|x_1)$
Step 4: end for
Step 5: for all visible units j do
Step 6: compute $P(x_{2j} = 1|h_1)$ (for binomial units, $sigm(b_j + \sum_i W_{ij}h_{1i})$)
Step 7: sample $x_{2j} \in \{0, 1\}$ from $P(x_{2j} = 1|h_1)$
Step 8: end for
Step 9: for all hidden units j do
Step 10: compute $Q(h_{2i} = 1|x_2)$ (for binomial units, $sigm(c_i + \sum_j W_{ij}x_{2j})$)
Step 11: end for
Step12: $W \leftarrow W + \in (h1x'_1 - Q(h_{2i} = 1|x_2)x'_2)$
Step 13: $b \leftarrow b + \in (x_1 - x_2)$
Step 14: $c \leftarrow - c + \in (h_1 - Q(h_{2i} = 1|x_2))$

4. RESULTS AND DISCUSSION

The experiments on these schemes are conducted on a laptop running Windows operation system with the following settings: CPU: Intel core i5 CPU at 2.5GHz; RAM memory: 4 GB

Parameters	Existing System(RF)	Proposed System(RNN)
Precision	46.02	52.09
Recall	81.27	91.66
F-Measure	54.00	64
Accuracy	83.87	92.55

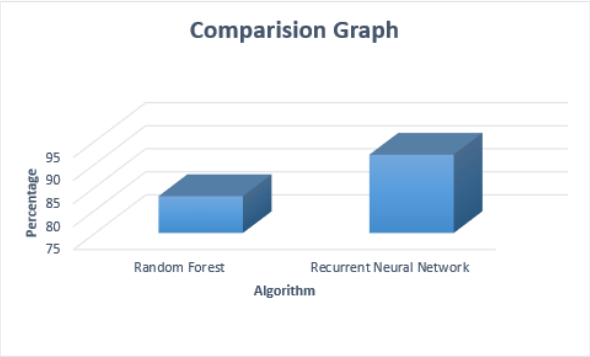

Chart-1 Comparison graph

5. CONCLUSION

In this paper, we have proposed deep learning approach for intrusion detection. Some widely used deep learning architectures are investigated and selected applications to intrusion detection. In response to this proposed the proposed approach for feature learning. After then built upon this by proposing a novel classification model constructed from Recurrent neural network classification algorithm. The result shows that given approach offers high levels of accuracy, precision and recall together with reduced training time. The proposed NIDS system is improved only 8% accuracy using recurrent neural network.

REFERENCES

[1] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system," in Advanced Communication Technology (ICACT), 2018 20th International Conference on, 2018, pp. 178–183.

[2] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in Advanced Cloud and Big Data (CBD), 2014 Second International Conference on, 2014, pp. 247– 252.

[3] S. Seo, S. Park, and J. Kim, "Improvement of Network Intrusion Detection Accuracy by Using Restricted Boltzmann Machine," in Computational Intelligence and Communication Networks (CICN), 2016 8th International Conference on, 2016, pp. 413–417.

[4] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on, 2016, pp. 195–200.

[5] J. Kim, J. Kim, H. L. T. Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in Platform Technology and Service (PlatCon), 2016 International Conference on, 2016, pp. 1–5.

[6] 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. 21954–21961, 2017.

[7] S. Althubiti, W. Nick, J. Mason, X. Yuan, and A. Esterline, "Applying Long Short-Term Memory Recurrent Neural Network for Intrusion Detection," in SoutheastCon 2018, 2018, pp. 1–5.

[8] T. A. Tang, S. Ali, R. Zaidi, D. Mclernon, L. Mhamdi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2018, pp. 25–29.

[9] Y. Yao, Y. Wei, F. Gao, and G. Yu, "Anomaly intrusion detection approach using hybrid MLP/CNN neural network," in Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on, 2006, vol. 2, pp. 1095–1102.

[10] K. Wu, Z. Chen, and W. Li, "A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks," IEEE Access, vol. 6, pp. 50850–50859, 2018.

[11] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of intrusion detection using deep neural network," in Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on, 2017, pp. 313–316.

[12] 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," in Wireless Networks and Mobile Communications (WINCOM), 2016 International Conference on, 2016, pp. 258–263.