International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161
Available at http://inpressco.com/category/ijcet

Research Article

Network Intrusion Detection using a Deep Learning Approach

Miss.Deepika R.Pede and Dr.Vaishali P.Vikhe

Department of Computer Engineering Pravara Rural Engineering College Savitribai Phule Pune University Loni, India

Received 10 Nov 2020, Accepted 10 Dec 2020, Available online 01 Feb 2021, **Special Issue-8 (Feb 2021)**

Abstract

Network based communication is more vulnerable to outsider and insider attacks in recent days due to its widespread applications in many fields. Intrusion Detection System (IDS) a software application or hardware is a security mechanism that can monitor network traffic and find abnormal activities in the network. As attackers always change their techniques of attack and find alternative attack methods, IDS must also evolve in response by adopting more sophisticated methods of detection.The huge growth in the data and the significant advances in computer hardware technologies resulted in the new studies existence in the deep learning field, including intrusion detection. Deep learning (DL) is a subset of Machine Learning (ML) that is based on learning data representations. This paper proposes a novel deep learning model to enable IDS operation within modern networks. The model shows a combination of deep learning and machine learning, capable of correctly analyzing a wide range of network traffic. The novel approach proposes non-symmetric deep autoencoder (NDAE) for unsupervised feature learning. Moreover,it additionally proposes a novel classification model built utilizing stacked NDAEs.The performance is evaluated using a network intrusion detection analysis dataset, particularly the WSN Trace dataset. The contribution work is to implement advanced deep learning algorithm contains IDS functionality but more sophisticated systems which are capable of taking immediate action to prevent or reduce the malicious behavior.

Keywords: *Intrusion Detection System (IDS), NonSymmetric Deep Auto-Encoder (NDAE), Deep Learning (DL), WSN, Machine Learning (ML).*

Introduction

The 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 a very critical and important issue that cannot be ignored. It is necessary to achieve the three basic principles that any security 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 to detect different types of malicious network communications and computer systems usage, whereas conventional firewall can not perform this task. Intrusion detection is based on the assumption that the behavior of intruders is different from a 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 has reached 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 the 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 technique 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]. This paper proposes a deep learning approach to enable NIDS operation within modern networks.

A. Objectives

- To devise a technique capable of providing reliable unsupervised feature learning, which can improve the performance and accuracy of existing techniques.
- To study existing Network Intrusion Detection Systems (NIDSs) and types of NIDSs.
- To study various deep learning algorithms for traffic classification.
- To study stacked Non-Symmetric Deep Auto-Encoder for unsupervised feature extraction.
- To analyze the experimental results of proposed stacked NDAE and RF classifier algorithms for the intrusion detection system.
- To reduce the training time.

Review of Literature

Fahimeh Farahnakian et al. proposed a Deep Auto Encoder (DAE) model which is trained in a greedy layer-wise fashion to avoid overfitting and local optima. Their suggested Deep Auto Encoder based IDS (DAE-IDS) is made up of four autoencoders, 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 before 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 because 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 a total of 10% KDD-CUP 1999 testing data-set [10].

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 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 [11].

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 [12].

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 pre-trained 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 [13], [14] 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 a wider range of attacks [15].

Jihyun Kim et al. constructed a model for IDS with the 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. Because 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
[16].

Yin Chuan-long et al. [17], [18] 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.

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 to improve 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 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 the 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 to recognize new patterns of attacks by the detection of the BSM strings [20].

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,Because 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 [21].

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 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 to 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% [22].

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 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 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 [23].

III. PROPOSED METHODOLOGY

This system proposes a deep learning model to enable IDS operation within modern networks. The model proposed is a combination of deep learning and machine learning, capable of correctly analyzing a wide range of network traffic. More specifically, combine the power of stacking our proposed Nonsymmetric Deep Auto-Encoder (NDAE) (deep learning) and the accuracy and speed of Random Forest (RF). This paper proposes NDAE, which is an auto-encoder featuring nonsymmetrical multiple hidden layers. NDAE is utilized as a

hierarchical unsupervised feature extractor that scales properly to deal with excessive-dimensional inputs. It learns non-trivial features using a similar training approach to that of a regular auto-encoder. Stacking the NDAEs offers a layer-wise unsupervised representation learning algorithm, which will allow our model to learn the complex relationships between different features. It also has feature extraction capabilities, so it can refine the model by prioritizing the most descriptive features. The input traffic data is the WSN Trace dataset with 12 features. The training dataset contains data preprocessing which includes three steps: data preprocessing, data normalization and data transformation. After that uses two NDAEs arranged in a stack, which are used for selecting the number of features. After that apply the Random Forest Classifier for attack detection.There are 8 rule actions when the attack is detected or not, the system will take the action using the following list:

- ALERT - Generate an alert using the selected ALERT method, and then log the packet
- LOG - Log the packet
- PASS - Ignore the packet
- ACTIVATE - Alert and then turn on another dynamic rule • DYNAMIC - Remain idle until activated by an activate rule, then act as a log rule
- DROP - Block and log the packet
- REJECT - Block the packet, log it, and then send a TCP reset if the protocol is TCP or an ICMP port unreachable message if the protocol is UDP

- SDROP - Block the packet but do not log it.

A. Architecture

The Proposed system consists of mainly three steps. In this section, we discuss each step in detail.

1. Data Preprocessing: This phase is made up of preprocessing, normalization and transformation.

a) Preprocessing

Neural network-based classification only uses numerical values for training and testing.WSN Trace dataset consists of different data types. Hence a preprocessing stage is needed to convert the non-numerical values to numerical values.

Two main tasks in pre-processing are:

1) Converting the non-numerical features in the dataset to numerical values.

2) Convert the attack types into its numeric categories

b) Normalization

The features of the WSN Trace dataset have either discrete or continuous values. The ranges of the value of feature are different and this makes them incomparable. So the features are normalized by using min-max normalization to map all the different values for each feature to [0, 1] range. Thus, the normalization procedure is done on the numeric values to bring the dataset into the same range.

c) Transformation

During this stage numeric normalized values of the features is converted into its optimal form.

2. Feature Selection: Auto Encoder is an unsupervisedneural network-based feature extraction algorithm that applies backpropagation in setting target value to be equal to the input. The Objective of the autoencoder is to minimize the reconstruction error between input and output. The proposed NDAE, which is an autoencoder featuring non-symmetrical multiple hidden layers. The reason behind this is, it reduces both computational and time overhead with minimal impact on accuracy and efficiency. Stacking NDAEs offer a layerwise unsupervised representation learning algorithm, which allows the model to learn complex relationships between different features.

3. Random Forest Classifier: Classification power of stacked autoencoder is relatively weak compared to other discriminative models such as RF, KNN, SVM. In this model, RF Classifier is trained using encoded representations learned by stacked NDAEs to classify network traffic into normal data and known attacks.

B. Mathematical Model

In this step, training data source (T) is normalized to be equipped for processing by using following steps:

$$T_{norm} = \{ \frac{T-\mu_T}{\sigma_T},\ \sigma_T \neq 0\, and T - \mu_T,\ \sigma_T = 0 \quad (1)$$

Where,

$T = \{x_{i,j} | i = 1,2,...,m\ and\ j = 1,2,3,...,n\}$ $\mu_T = \{\mu_j | j = 1,2,3,...,n\}$ $\sigma_T = \{\sigma_j | j = 1,2,3,...,n\}$

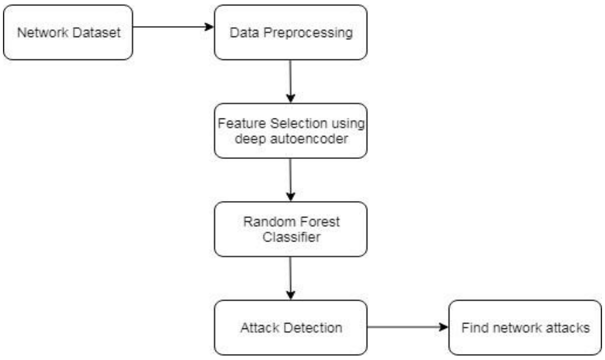

Fig. 1. Proposed System Architecture

T is m samples with n column attributes; x_{ij} is the jth column attribute in ith sample, $_T$ and $_T$ are $1 * n$ matrix which are the training data mean and standard deviation respectively for each of the n attributes. Test dataset (TS) which is used to measure detection accuracy is normalized using the same $_T$ and $_T$ as follows:

$$TS_{norm} = \frac{\sigma_T}{\sigma_T}(x)/\sigma_T,\ \sigma_T \neq 0\ and\ TS - \mu_T, \sigma_T = 0\ (2)$$

NDAE is an auto-encoder featuring non-symmetrical multiple hidden layers. The proposed NDAE takes an input vector $x \in R^d$ and step-by-step maps it to the latent representations $h_i \in R^d$ (here d represents the dimension of the vector) using a deterministic function shown in (3) below: $h_i = \sigma(W_i.h_{i-1} + b_i); i = 1,n,\bar{}$ (3)

Here, $h_0 = x$, σ is an activation function (in this work use sigmoid function $\sigma(t) = 1/(1 + e^{-t})$ and n is the number of hidden layers. Unlike a conventional Auto-Encoder and Deep Auto-Encoder, the proposed NDAE does not contain a Decoder and its output vector is calculated by a similar formula to (4) as the latent representation.

$$y = \sigma(Wn+1.hn + bn+1) \quad (4)$$

The estimator of the model $\theta = (W_i,b_i)$ can be obtained by minimizing the square reconstruction error over m training samples $(x^{(i)}, y^{(i)})_{i=1}^{m}$, as shown in (5).

$$E(\theta) = \sum_{i=1}^{m} (x^{(i)}, y^{(i)})^2 \quad (5)$$

C. Algorithms

1. Restricted Boltzamine Machine Algorithm x_1 is a sample from the training distribution for the RBM ∈ is a learning rate for the stochastic gradient descent in Contrastive Divergence W is the RBM weight matrix, of dimension (number of hidden units, numbr of inputs) b is the RBM offset vector for input units c is the RBM offset vector for hidden units

Notation: $Q(h_{2i} = 1|x_2)$ is the vector with elements $Q(h_{2i} = 1|x_2)$

Step 1: for all hidden units i do

Step 2: compute $Q(h_{1i} = 1|x_1)$ (for binomial units, $sigm(ci + \mathrm{P}j\ Wijx1j)$)

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, $sigm(bj + \mathrm{P}i\ Wijh1i)$)

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, $sigm(ci + \text{P}j\ Wijx2j)$) Step 11: end for
Step12: $W \longleftarrow W + \in (h_1 x_1' - Q(h_{2_i} = 1|x_2) x_2')$
Step 13: $b \leftarrow - b + \in (x_1 - x_2)$
Step 14: $c \leftarrow - c + \in (h_1 - Q(h_{2i} = 1|x_2))$
```

2. Deep Belief Network Algorithm

Train a DBN in a purely unsupervised way, with the greedy layer-wise procedure in which each added layer is trained as an RBM (e.g., by Contrastive Divergence).

$\hat{P}$ is the input training distribution for the network $\in$ is a learning rate for the RBM training l is the number of layers to train W^k is the weight matrix for level k, for k from 1 to l b^k is the visible units offset vector for RBM at level k, for k from 1 to l

c^k is the hidden units offset vector for RBM at level k, for k from 1 to l

$Mean\_field\_computation$ is a Boolean that is true iff training data at each additional level is obtained by a meanfield approximation instead of stochastic sampling

```
Step 1: for $k = 1$ to $l$ do
Step 2: initialize $W^k = 0, b^k = 0, c^k = 0$
Step 3: while not stopping criterion do
Step 4: sample $h^0 = x$ from $\hat{P}$
Step 5: for $i = 1$ to $k - 1$ do
Step 6: if mean _field computation then
Step 7: assign $h^i_j$ to $Q(h^i_j = 1|h^{i-1}$, for all elements $j$ of $h^i$
Step 8: else
Step 9: assign $h^i_j$ to $Q(h^i_j|h^{i-1}$, for all elements $j$ of $h^i$
Step 10:end if
Step 11:end for
Step12: $RBMupdate(h^{k-1}, \in, W^k, b^k, c^k)$ thus providing $Q(h^k|h^{k-1})$ for future use
Step 13: end while Step 14: end for
```

3. Random Forest Algorithm

Step 1: Let the number of training cases be N, and the number of variables in the classifier be M.

Step 2: The number m of input variables to be used to determine the decision at a node of the tree; m should be much less than M.

Step 3: For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

Step 4: Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction.

Results and Discussion

WSN-Trace is the wireless dataset for researchers. The WSN Trace dataset contains a total of 19 attributes. WSN Trace dataset is a real-time wireless dataset that gets information from the router which contains node details and packet information. In the training and testing dataset, there are 5 types of attacks which are subtypes of normal, probing, dos, u2r, and r2l attacks.

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: 4 GB.

Throughout this section, we will be using the metrics defined below:

1) TruePositive(TP) - Attack data that is correctly classified as an attack.

2) False Positive (FP) - Normal data that is incorrectly classified as an attack.

3) True Negative (TN) - Normal data that is correctly classified as normal.

4) False Negative (FN) - Attack data that is incorrectly classified as normal.

We will be using the following measures to evaluate the performance of our proposed solution:

Accuracy = TP + TN / TP + TN+ FP+ FN

The accuracy measures the proportion of the total number of correct classifications.

Precision = TP / TP + FP

The precision measures the number of correct classifications penalized by the number of incorrect classifications.

Recall = TP / TP + FN

The recall measures the number of correct classificationpenalized by the number of missed entries.

F-measure = 2· Precision·Recall / Precision + Recall

The F-measure the harmonic mean of precision and recall, which serves as a derived effectiveness measurement.

The performance of our proposed system, which is the combination of stacked NDAE (deep learning) and RF Classifier (machine learning) is given in the below table

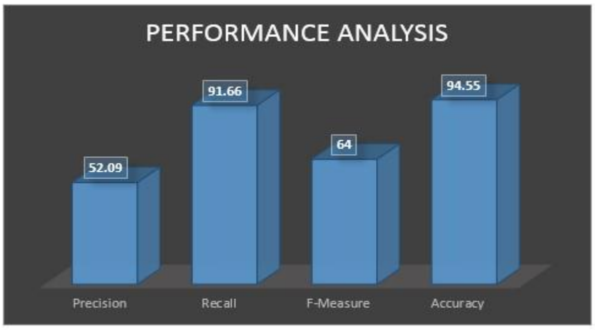

Fig. 2. Efficiency Graph

	Proposed System
Precision	52.09
Recall	91.66
F-Measure	64
Accuracy	94.55

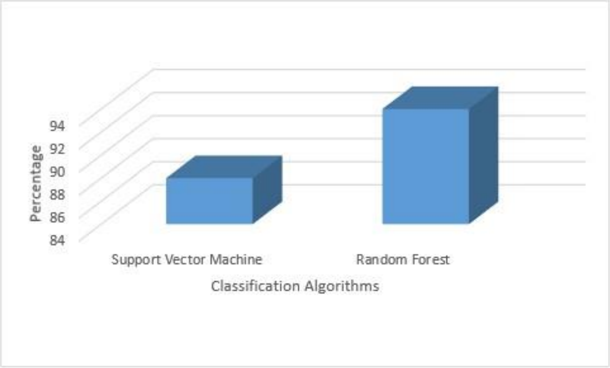

Fig. 3. Accuracy Graph

Conclusion

This paper mentioned the problems confronted by previous IDS techniques. In response to this proposed the novel NDAE approach for unsupervised feature learning. After then built upon this by proposing a novel classification model constructed from stacked NDAEs and the RF classification algorithm. The result shows that the given approach offers high levels of accuracy, precision and recall together with reduced training time. The proposed system has improved accuracy. Still, there is a need for further improvement of accuracy. And also further work on real-time network traffic.

References

[1]. R. Bace and P. Mell, "NIST special publication on intrusion detection systems," BOOZ-ALLEN AND HAMILTON INC MCLEAN VA, 2001. [2] A. Lazarevic, V. Kumar, and J. Srivastava, "Intrusion detection: A survey," in Managing Cyber Threats, Springer, 2005, pp. 19–78.

[2]. W. Stallings, "Cryptography and network security principles and practices," USA: Prentice Hall, 2006.

[3]. M. H. Aghdam and P. Kabiri, "Feature Selection for Intrusion Detection System Using Ant Colony Optimization," IJ Netw. Secur, vol. 18, no. 3, pp. 420–432, 2016.

[4]. C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," Expert Syst. Appl., vol. 36, no. 10, pp. 11994–12000, 2009.

[5]. B. Dong and X. Wang, "Comparison deep learning method to traditional methods using for network intrusion detection," in Proc. 8th IEEE Int. Conf. Commun. Softw. Netw., Beijing, China, Jun. 2016, pp. 581–585.

[6]. R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao, "Deep learning and its applications to machine health monitoring: A survey," Submitted to IEEE Trans. Neural Netw. Learn. Syst., 2016. [Online].

[7]. Available: http://arxiv.org/abs/1612.07640.

[8]. S. Pouyanfar et al., "A Survey on Deep Learning: Algorithms, Techniques, and Applications," ACM Comput. Surv. vol. 51, no. 5, p. 92, 2018.

[9]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning. nature 521 (7553): 436," Google Sch., 2015.

[10]. 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.

[11]. 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.

[12]. 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.

[13]. M. A. Salama, H. F. Eid, R. A. Ramadan, A. Darwish, and A. E. Hassanien, "Hybrid intelligent intrusion detection scheme," in Soft computing in industrial applications, Springer, 2011, pp. 293–303.

[14]. Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," methods, vol. 9, no. 5, 2015.

[15]. 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.

[16]. 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.

[17]. 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.

[18]. 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.

[19]. 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.

[20]. 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.

[21]. 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.

[22]. 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

[23]. (BigComp), 2017 IEEE International Conference on, 2017, pp. 313–316.

[24]. 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.