An anomaly-based Network Intrusion Detection System using Deep learning

Nguyen Thanh Van, Tran Ngoc Thinh, Le Thanh Sach

***Abstract*— Recently, anomaly-based intrusion detection techniques are valuable methodology to detect both known as well as unknown/new attacks, so they can cope with the diversity of the attacks and the constantly changing nature of network attacks. There are many problems need to be considered in anomaly-based network intrusion detection system (NIDS), such as ability to adapt to dynamic network environments, unavailability of labeled data, false positive rate. This paper, we use Deep learning techniques to implement an anomaly-based NIDS. These techniques show the sensitive power of generative models with good classification, capabilities to deduce part of its knowledge from incomplete data and the adaptability. Our experiments with KDDCup99 network traffic connections show that our work is effective to exact detect in anomaly-based NIDS and classify intrusions into five groups with the accuracy based on network data sources.**

***Keywords*— network intrusion, anomaly-based network intrusion detection, deep learning.**

I. INTRODUCTION

The Internet along with the enterprise networks plays a major role in creating and supporting new business avenues. This makes today's networks more vulnerable to intrusions and attacks. The diversity of the attacks and the constantly changing nature of network attacks makes the development of network security is oriented flexibly. Therefore, it is required a flexible defense system with capable of analyzing large amounts of network traffic to exact detect many variety attacks. In this context, anomaly-based intrusion detection techniques is valuable methodology to detect both known as well as unknown/new attacks in intrusion detection systems. Through continuous observation and modeling of normal behavior in the networks, anomaly detection offers a way to find possible threats via deviation from the normal model.

In anomaly detection, anomalies are very important because they indicate serious but rare events, for example, an unusual traffic pattern in a network could mean that a computer is attacked and data is transmitted to unauthorized destinations. A significant factor of anomaly detection is the nature of the anomaly. An anomaly can be categorized in the three ways [1], includes point anomaly, contextual anomaly and collective anomaly. These different types of anomaly have the relationship with the attacks in network security, includes DoS, Probe, U2R, R2L. The *DoS* attack characteristics match with the collective anomaly. *Probe* attacks are based on specific purpose to get information and reconnaissance, so they can be matched to contextual anomaly. *User to Root* (U2R) attacks are illegal access to the administrator account, exploiting one or several vulnerabilities. *Remote to local* (R2L) attacks are local access to get the privilege to send packets on the network, the attacker uses trial and error to guess the password. Both U2R and R2L attacks are condition specific and sophisticated. Initiation of these attacks are not similar as compared to others, therefore, these attacks are considered as point anomaly. All these types of anomaly need to be detected by network intrusion detection systems, then analyzed and classified into network attack types. Besides those known attacks, many unknown attacks are detected based on anomalous in network traffic. However, in many case, anomalous may be normal behaviors in the system that are haven't updated. Therefore, these anomaly-based NIDSs should adapt to dynamic network environments with new network protocols and updated behaviors. In the past years, several different techniques have been used in anomaly-based NIDSs such as statistical-based, knowledge-based, and machine learning-based. However, there are research challenges need to be scrutinized to improve performance and make them suitable with current network data characters.

In anomaly-based network intrusion detection, the system is trained with "normal" network traffic to generate *Model*. When the *Model* for the system is available, it is subsequently used to classify new events or objects or traffic as anomalous or not. To get an effective model, an expected characteristic is its ability to adapt to generalize its behavior and cope with dynamic network environments. That means a self-learning system is needed.

Another major issue in anomaly-based network intrusion detection is availability of labeled data for training and validation of models. Labels for normal behavior are usually available, while labels for intrusions are not. Therefore, semi-supervised and unsupervised anomaly detection techniques are preferred in this case.

Therefore, motivated by problems above, we propose an anomaly-based NIDS using deep learning. With deep learning, we expect to tackle issues in anomaly-based network intrusion detection, such as high intrusion detection rate, ability to adapt to dynamic network environments, unavailability of labeled data.

Deep learning a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for unsupervised feature learning and for pattern analysis or classification. Deep learning show the power of generative models with

Nguyen Thanh Van. Author is now with Ho Chi Minh City University of Technology and Education - Vietnam (HCMUTE), No 1 Vo Van Ngan Street, Linh Chieu Ward, Thu Duc District, HCM City (corresponding author, phone: +84 905131246; e-mail: vannt@fit.hcmute.edu.vn).

Tran Ngoc Thinh. Author is now with Ho Chi Minh City University of Technology - Vietnam (HCMUT) (e-mail: tnthinh@hcmut.edu.vn).

Le Thanh Sach. Author is now with Ho Chi Minh City University of Technology - Vietnam (HCMUT) (e-mail: ltsach@hcmut.edu.vn).

978-1-5386-3422-6/17/$31.00 ©2017 IEEE

high accuracy classification, capabilities to extract parts of its information from incomplete training data. Deep learning techniques have been successfully applied to fields like computer vision, automatic speech recognition, natural language processing, and music/ audio signal recognition. In intrusion detection field, there are few researches using deep learning but they have not exploited the power of deep learning techniques effectively. Our work exploited the strengths of self-learning, process with big data, analyzed and compared deep learning techniques in anomaly-based NIDS.

II. BACKGROUND AND RELATED WORKS

A. Anomaly-based NIDS

Intrusion detection systems are security systems that has ability collect and analyze information from various types of system and network sources to detect activity that may be an attack or illegal access on the system. Intrusion detection refers to detection of malicious activities such as break-ins, penetrations, and other forms of computer abuse in a host or network. An intrusion is different from the normal behavior of the system, and hence anomaly detection techniques are applicable in intrusion detection domain.

Depending on the information source considered, an IDS may be either host or network-based. A host-based IDS analyzes events such as process identifiers and system calls that related to OS information. The intrusions are in the form of anomalous subsequences of the traces. The anomalous subsequences translate to malicious programs, illegal behavior and policy harms. Anomaly detection techniques applied for host based intrusion detection are required to handle the sequential nature of data. The techniques have to either model the sequence data or compute similarity between sequences. There are two common types of technique to detect anomalies using system call traces: short sequence-based and frequency-based [2].

On the other hand, a network-based IDS analyzes network related events, such as traffic volume, IP addresses, service ports, protocol usage, etc. In network-based IDS, intrusions typically are referred as anomalous through continuous observation and modeling of normal behavior in the networks. The data is high dimensional typically with a mix of categorical as well as continuous attributes. Therefore, the anomaly detection techniques need to be computationally efficient to handle these large sized inputs. Moreover the data typically comes in a streaming fashion, thereby requiring on-line analysis. Labeled data corresponding to normal behavior is usually available, while labels for intrusions are not. Thus, semi-supervised and unsupervised anomaly detection techniques are preferred in this domain. Some anomaly detection techniques used in network-based IDS are classify into four major [3] which include classification, statistical, information theory and clustering.

Because of the large sized input and dynamic network environment, an issue is the false positive rate. To address this problem, an expected characteristic is ability to adapt to generalize its behavior and cope with dynamic network environments. That means a self-learning system is needed. In our work, we focus on researching and implementing the anomaly detection in network-based IDS, called anomaly-based NIDS. The technique used in anomaly-based NIDS based on Deep learning will solve some challenges above.

B. Deep learning techniques

The principle of deep learning is to compute hierarchical features or representations of the observational data, where the higher-level features or factors are defined from lower-level ones. The Deep learning techniques aim to learn a good feature representation from a large amount of unlabeled data, so the model can be pre-trained in a completely unsupervised fashion. The very limited labeled data can then be used to only slightly fine-tune the model for a specific task in the supervised classification. The strategy of layer-wise unsupervised training followed by supervised fine-tuning allows efficient training of deep networks and gives promising results for many challenging learning problems, substantially improving upon the current [4]. Initializing weights in an area where is near a good local minimum make increase to internal distributed representations that are high-level abstractions of the input, thus it brings a better generalization. There are many Deep learning techniques used to pre-training and they are chosen depending on different domains. In our system, we use two techniques, they are Restricted Boltzmann Machines (RBM) and Autoencoder.

1) Restricted Boltzmann Machines – RBM

An RBM defines a probability distribution **P** on data vectors **v** based on energy model [5]. The variable **v** is the input vector and the variable **h** corresponds to unobserved features that be called hidden layer. The probability model is defined:

$$P(v, h; \theta) = \frac{1}{Z(\theta)} \exp\left(-E(v, h; \theta)\right) \quad (1)$$

where θ = *(W, b, c), weight: W, bias: b, c.* $Z(\theta)$ is the partition function defined as follows:

$$Z(\theta) = \sum_{v,h \in \{0,1\}} \exp\left(-E(v, h; \theta)\right) \quad (2)$$

The energy of the joint state {v, h} is defined as follows:

$$E(v, h; \theta) = -\sum_{i,j} v_i w_{ij} h_j - \sum_j b_j h_j - \sum_i c_i v_i \quad (3)$$

The energy function E is constructed to make the conditional probabilities p(h|v) and p(v|h) tractable with binary data. They are shaped as follows:

$$p(\mathrm{v}|\mathrm{h}) \text{ with } p(v_i = 1|\mathrm{h}) = sig\Big(\sum_j w_{ij} h_j + b_i\Big) \quad (4)$$

$$p(\mathrm{h}|\mathrm{v}) \text{ with } p\big(h_j = 1|\mathrm{v}\big) = sig\Big(\sum_i w_{ij} v_i + c_j\Big) \quad (5)$$

where sig(x) = 1/ (1 + exp(−x)).

With continuous valued data, E can be appropriately modified to define the Gaussian-Bernoulli RBM as follows:

$$E(v,h;\theta) = \frac{1}{2\sigma^2}\sum_i (v_i - c_i)^2 - \frac{1}{\sigma}\sum_{i,j} v_i w_{ij} h_j - \sum_j b_j h_j \quad (6)$$

where σ_i represents the variance of the input variable v_i.

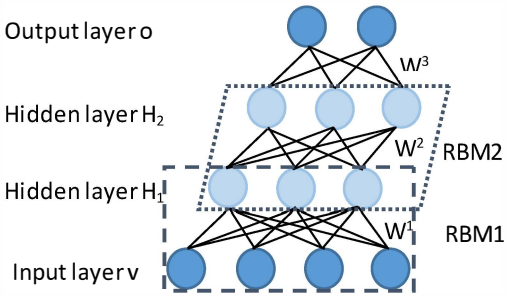

Figure 1. A stacked RBM

In RBM, the hidden variables are independent conditionally to the visible variables, but they are not statistically independent. Stacking RBMs aims at learning these dependencies with another RBM, as shown in Fig. 1. The visible layer of each RBM of the stack is set to the hidden layer of the previous RBM. A stacked RBMs architecture is a deep generative model. Patterns generated from the top RBM can be propagated back to the input layer using only the conditional probabilities as in a deep belief network [6].

In order to train RBMs as a probabilistic model, the natural criterion to maximize is the log-likelihood. It is estimated by Contrastive Divergence (CD) which is an approximation of an objective function with parameter θ and dataset, as shown follows:

$$L(\theta) = \sum_n L_n(\theta) = \sum_n \log P(v_n;\theta) \quad (7)$$

Basing on P defined above, the log-likelihood is modified as follows:

$$\frac{\partial \log P(\mathrm{v})}{\partial\theta} = E_{h\sim P(\mathrm{h}|\mathrm{v};\,\theta)}\left[-\frac{\partial}{\partial\theta}E(\mathrm{h},\mathrm{v})\right] - E_{\mathrm{v}',\mathrm{h}\sim P(\mathrm{v},\mathrm{h};\theta)}\left[-\frac{\partial}{\partial\theta}E(\mathrm{h}',\mathrm{v})\right] \quad (8)$$

Then, parameter θ will be deduced as follows:

$$\theta^{new} := \theta^{old} + \eta\frac{\partial \log P(\mathrm{v})}{\partial\theta} \quad (9)$$

2) Autoencoder

The most important advantage of deep learning is replacing manual features with better algorithms which can learn and extract hierarchical features by unsupervised or semi-supervised learning. Autoencoder is an example. The structure of Autoencoder is a two-layer neural network [5], as shown in Fig. 2. The first layer is the encoding layer and the second is the decoding layer. The goal of an Autoencoder is to balance between a code **z** of an input instance **x** and an **x** can be reconstructed $\hat{x}$. They are defined as follows:

$$z^{(1)} = h^{(1)}(W^{(1)}x + b^{(1)}) \quad (10)$$

And $$\hat{x} = h^{(2)}(W^{(2)}x + b^{(2)}) \quad (11)$$

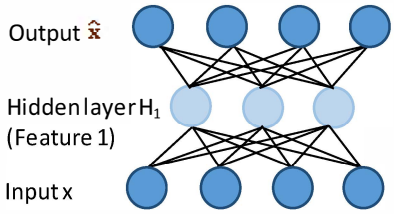

Figure 2. The structure of an Autoencoder

This models a two-stage approximation to the identity function: $f_{dec}(f_{enc}(x)) = f_{dec}(y) = \hat{x} \sim x$, with f_{enc} the function computed by the encoding layer and f_{dec} the function computed by the decoding layer.

Depending on the nature of the input data, the loss function can either be the squared error L_{SE} for continuous values or the cross-entropy L_{CE} for binary vectors:

$$L_{SE}(x,\hat{x}) = \sum(\hat{x}_i - x_i)^2, \quad (12)$$

$$L_{CE}(x,\hat{x}) = \sum [x_i \log \hat{x}_i + (1 - x_i)\log(1 - \hat{x}_i) \quad (13)$$

The process of training an Autoencoder is unsupervised basing on the optimization of a cost function. The Autoencoder training method approximates the CD method of the RBM.

A stacked Autoencoder (SAE) is a form of building a deep belief network DBN [5]. It is trained by an unsupervised greedy layer-wise pre-training before a fine-tuning supervised stage. SAE is possible to use different regularization rules with adding a sparsity constraint on the encoding unit activations. This leads to learning very different features in the intermediate layers.

C. Related works

In the past years, several different techniques have been used in anomaly-based NIDSs such as classification, statistical, information theory and clustering. However, there are the research challenges, such as network data size, unavailble labled data and features, and adaptation to dynamic network environments. There are few researchs using Deep learning and gain some preliminary results. Through reviewing literatures, we found that Deep learning could be well applied in the area of network intrusion detection [7, 8, 9], malicious code detection [10] and network traffic identification [11]. The two techniques used primarily in the pre-training phase are RBM [7, 8, 10] and Auto-Encoder [9, 10, 11] to reduce feature or feature extraction. However, no research has compared the effectiveness of these two techniques. For the data sets used, most of the above studies evaluate their systems using the KDDCUP99 [7, 8, 9, 10] data set. KDDCUP99 is supposed to be a sample data set for IDS with many types of attacks included, but the data are not realistic, obsolete and lacking the current network features. Some studies [8, 10] tested with small sample size, not enough comprehensive evaluate the system.

III. APPLYING DEEP LEARNING TO A-NIDS

Anomaly-based NIDS is an effective way to detect new attacks. However, its limitation is false positive when a normal behavior is not in the normal network data set. Therefore, how to model properly all the normal network data, whether they share a common range of values for a set of features, as well as correlation and sequencing between them. A recent study [12] assume that intrinsic network traffic is similar to itself, and is different from that of normal traffic. The idea is how to test whether there is a deep resemblance between normal behaviors. Hence, it is necessary to have a learning model that can describe all the problems of normal network traffic when compared to unseen network traffic. RBMs and AutoEncoders in Deep learning have a combination of demonstration capabilities of production models with good classification accuracy, so they are used as a tool to test this hypothesis. In addition, Deep learning techniques can address the challenges in anomaly-based NIDS such as network data size, unavailble labled data and features, and adaptation to dynamic network environments.

A. Designing Deep Belief Network (DBN) structure

Our work design a DBN struture for anomaly-based NIDS and train it using Stacked RBM and Stacked Autoencoder. Both techniques are used in the pre-training period of Deep learning with the models of multilayer architecture. However, they have differences in structure as shown in Fig. 3. In the finding the appropriate DBN architecture [13], in general, the more layers and more units in a layer an architecture has, the higher performance it is [14]. This makes network training more difficult and slower. Various points need to be scrutinized to avoid overfitting and underfitting problems. We calculated and tested with the DBN network structures to achieve the most suitable model for anomaly-based NIDS. An Autoencoder is composed of an input, a hidden and an output layer. The output layer is reconstructed by the input through the activation function of some of the hidden units. It provides a simple solution to reduce the dimension (similar to PCA). After an Autoencoder is trained, it can be used as input features for another Autoencoder. Therefore, compared to RBM, Autoencoder is intuitive and easy to understand and implement, namely easier to find the parameters for it. On other hand, RBM is made from an input and a hidden layer. An RBM is executed by defining a probability distribution **P** based on energy model sampling in the training, therefore, it is more complex computed. However, with ability to learn and extract hierarchical features, Autoencoder has much more calculations in training. When the training process is finished, we achieve dimensionality reduction. Features are mapped to new space. Redundant information is filtered as well.

After pre-training phase, a supervised fine-tuning phase with the multilayer network (the middle block is shown in Fig. 3) in will be performed by BackPropagation to obtain the best final weight set.

B. Implementation anomaly-based NIDS

We implement our system with two techniques, RBM and Autoencoder on KDDCup99 dataset [15]. KDDCup99 dataset consists of approximately 4,900,000 single connection vectors, each of which contains 41 features. The simulated attacks fall in one of the four categories: DoS, Probe, U2R, R2L. The dataset is preprocessed to match to the detection method of anomaly-based network intrusion detection system proposed. A statistical analysis is performed on the values of each feature based on the existing data from KDD99 dataset. We partition the dataset into three parts: training, validation and testing. Gradient descent on the objective function is performed with data in the training set, the validation set can be used during training to see how well it is performing in unseen data. The pre-training process allow neural network is constructed naturally from layer to layer. It also support for neural network initialization with parameters better than that with random parameters.

IV. EXPERIMENTS AND RESULTS

Our experiments with KDDCup99 network traffic connections have shown that using Deep learning techniques is effective to exact detect intrusions with low error rate. Our system not only detect attacks but also classify them in five groups (*Normal, DoS, Probe, U2R, R2L*) with the high accuracy of identifying and classifying network activity based on data sources.

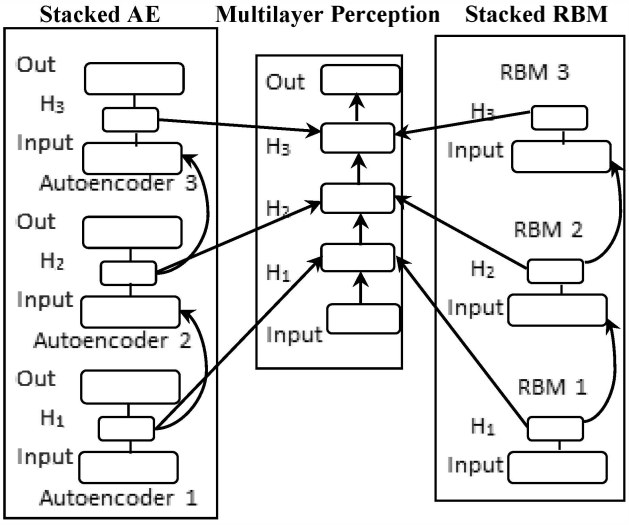

Figure 3 Comparison between SAE and SRBM

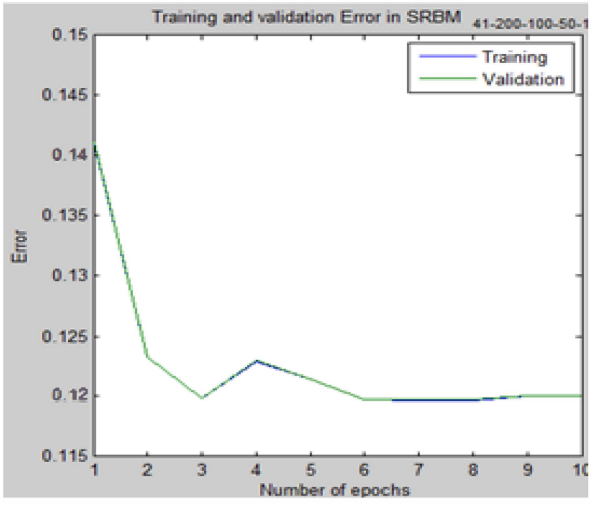

Figure 4. Error of training and validation with RBM

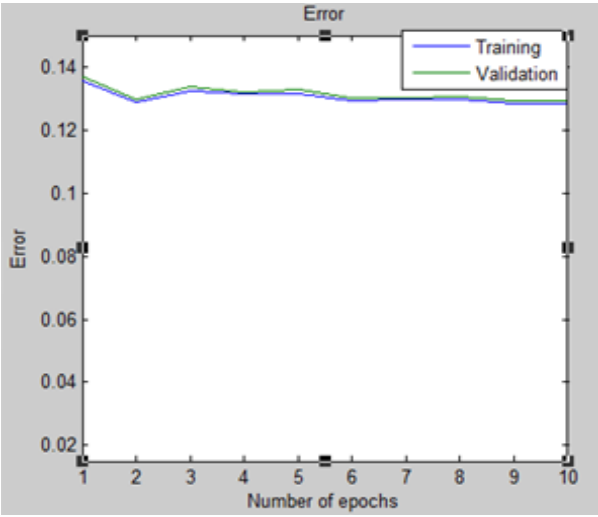

Figure 5. Error in classification of RBM with 4 attack groups

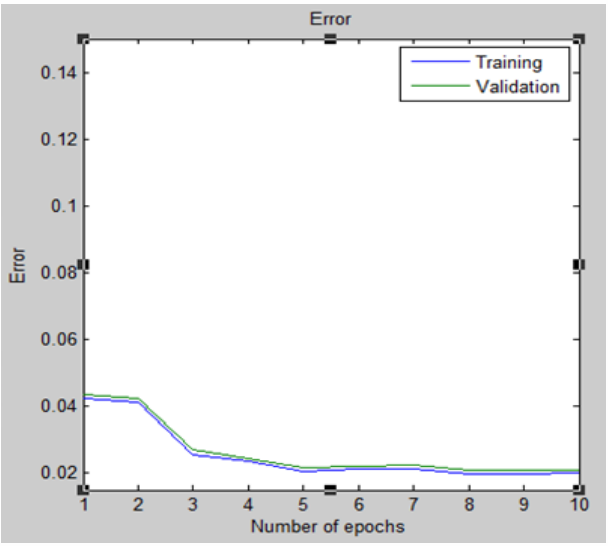

Figure 6. Error in classification of AE with 4 attack groups

We compare between anomaly and group classification in the same RBM technique, Fig. 4, Fig. 5 show that anomaly classification has lower error rate even though it has few oscillations in reducing the slope in the training process.

Comparison between two techniques RBM and Autoencoder, Fig. 5 and Fig. 6 show that Autoencoder is much better than RBM in classification into four groups of attacks and normal, respectively 0.02 and 0.13.

We also compare the execution time of Stacked RBM and Stacked Autoencoder in the same struture, Table 1 shows that the time taken by Stacked Autoencoder takes longer. Although the cost function of Autoencoder is simpler, much more calculations in training process are executed to achieve dimensionality reduction.

TABLE 1. EXECUTION TIME IN SRBM AND SAE

Structure	SRBM	SAE
41-200-100-50-1	161,659s	240,133s
41-200-100-50-4	240.870s	450.639s

V. CONCLUSION AND FUTURE WORKS

In this paper, we research, implement and evaluate of the application of Deep Learning to anomaly-based NIDS. The results show that our system detect intrusion and classify attacks in four groups with high accuracy using two deep learning techniques. Our experiments with two pre-training techniques of Deep learning, includes Stacked Autoencoder and Stacked RBM, show that the use of Stacked Autoencoder technique is better than that in RBM. The problem of the training time-consuming in SAE is much more than that in RBM due to much computations in SAE. Our future works will study implementation the system in parallel platforms to improve speed. Techniques are used in pre-training should also be considered in order to optimize and reduce oscillation in reducing the slope of the training error to get higher accuracy.

REFERENCES

[1] V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol. 41, no. 3, pp. 1-58, 2009.

[2] Forrest S., Hofmeyr S., Somayaji A., "The Evolution of System-Call monitoring," in Annual Computer Security Applications Conference, ACSAC, 2008.

[3] M. Ahmed, A.N. Mahmood, J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, p. 13, 2015.

[4] Hinton G, Deng L, Yu D, Dahl GE, Mohamed, "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Process, vol. 29, no. 6, p. 82–97, 2012.

[5] L. Arnold, S. Rebecchi, S. Chevallier, H. Paugam-Moisy, "An Introduction to Deep Learning," in European Symposium on Artificial Neural Networks, Bruges (Belgium), 2011.

[6] Hinton GE., S. Osindero and Yee-Whye Teh, "A fast learning algorithm for deep belief nets," Neural Computation, p. 15, 2006.

[7] U.Fiore, F.Palmieri, A.Castiglione, A.Santis, "Network anomaly detection with the RBM. Neurocomputing," Neurocomputing, Elsevier B.V, p. 11, 2013.

[8] Md. Zahangir Alom, V.Bontupalli, and Tarek M. Taha, "Intrusion Detection using DBN," in National Aerospace and Electronics Conference (NAECON), USA, 2015.

[9] Q. Niyaz, W. Sun, A. Javaid, and M. Alam, "A Deep Learning Approach for NIDS," in Bio-inspired Information and Communications Technologies, (BIONETICS), Brussels, Belgium, 2014.

[10] Y. Li, R. Ma and R. Jiao, "A Hybrid Malicious Code Detection method based on Deeplearning," International Journal of Security and Its Applications - IJSIA, vol. 9, no. 5, pp. 205-216, 2015.

[11] W. Zhanyi, "The Applications Of Deep Learning On Traffic Identification," Blackhat, USA, 2015.

[12] J.-S.R. Lee, H.-D.J. Jeong, D.C. McNickle, K. Pawlikowski, "Self-Similar Properties of Spam," in 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS, 2011.

[13] G. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, UTML TR 2010-003, University of Toronto, 2010.

[14] Elisseeff, A., and Paugam-Moisy, H., "Size of multilayer networks for exact learning: analytic approach," Advances in Neural Information Processing Systems 9, pp. 162-168, 1997.

[15] M. Lincoln, "Dataset, KDD. Kdd cup 1999," 1999.