Network Intrusion Detection for Cyber Security using Unsupervised Deep Learning Approaches

Md Zahangir Alom and Tarek M. Taha
Department of Electrical and Computer Engineering
University of Dayton
Dayton, OH 45469, USA
Email: alomm1@udayton.edu,ttaha1@udayton.edu

Abstract— **In the paper, we demonstrate novel approach for network Intrusion Detection System (IDS) for cyber security using unsupervised Deep Learning (DL) techniques. Very often, the supervised learning and rules based approach like SNORT fetch problem to identify new type of attacks. In this implementation, the input samples are numerical encoded and applied un-supervised deep learning techniques called Auto Encoder (AE) and Restricted Boltzmann Machine (RBM) for feature extraction and dimensionality reduction. Then iterative k-means clustering is applied for clustering on lower dimension space with only 3 features. In addition, Unsupervised Extreme Learning Machine (UELM) is used for network intrusion detection in this implementation. We have experimented on KDD-99 dataset, the experimental results show around 91.86% and 92.12% detection accuracy using unsupervised deep learning technique AE and RBM with K-means respectively. The experimental results also demonstrate, the proposed approach shows around 4.4% and 2.95% improvement of detection accuracy using RBM with K-means against only K-mean clustering and Unsupervised Extreme Learning Machine (USELM) respectively.**

I. Introduction

Nowadays, cyber security is a challenging issue in the cyber space and it has been increasing dramatically depending on computerization on different application domains including finances, industry, medical, and many other important areas. There is a strong demand for effective Intrusion Detection System (IDS) that is designed to interpret intrusion attempts of incoming network traffic efficiently, intelligently and energy effectively. In order to protect cyber-attack, awareness of an attack is essential to being able to react or defend against attackers. For instant, it is important to know immediately for additional precaution or possibly take law enforcement or legal actions, if information of credit card data has already been stolen. Intrusion detection can also be applied beyond detecting cyber-attack in noticing abnormal system behavior to identify accidents or unexpected conditions. For examples, IDS can be informed anomalies where a human error or malfunction is causing customer credit card numbers to be incorrectly changed several times. There are two type of intrusion based security system: misuse detection or classification and anomaly detection where unknown attacks are also recognized [1], [2]. In this implementation, we have explored misuse detection for specific type of cyber-attack using unsupervised deep learning approaches including AE and RBM. There are many researches have been already conducted in this area with unsupervised algorithm for network intrusion detection. The two tier architecture is used for clustering the packet payloads and correcting anomalies in the packet stream using Self Organizing Maps (SOM)[3].

Nowadays intrusion detection includes analysis of big data, which is defined as one of the challenging research topic where conventional computing technologies cannot deal with the huge volume of data [4], [5]. In addition, according to Nassar et a. merely 1 Gbps of nonstop network traffic can be caused big data challenges for Intrusion Detection using deep packet inspection[4]. Another issue arise for the big companies which deals with an incredible amount of host log event data. However, it is noted that for solving this type of big data problem with effective deep approaches is very essential for detecting cyber-attack and for undertaking future challenge in the field of cyber security. Deep learning on the other hand, is showing huge success in different application domains and achieves human-level performance on different benchmarks in particular for recognition tasks. In the last decade, the neural network based object recognition approaches attracting more attention to the researcher due to the revolution of new arena of neural network technique called deep learning (DL). DL approaches are more efficient for learning because of the combination of feature extraction and classification layers and shown better accuracy compare to shallow learning approach[6], [7]. Recently Convolutional Neural Networks (CNN) has shown a tremendous contribution in the field of machine learning and computer vision which is already proved potential on different benchmarks and provided the state-of-the-art accuracies of object recognition [8], [9], [10], [11], [12]. For instance: CNN shown better performance compare to handcraft features after training on 1.2 million images from ImageNet by a significant margin in classifying objects into 1000 categories [13]. However, there are some works have been proposed for intrusion detection for cyber security using supervised deep learning approach where Deep Belief Network (DBN) is used [14]. In addition, the ELM technique is used for cyber security which is compare with different deep learning algorithms [15].

Extreme Learning Machine (ELM) has got some attention of the community in the field of machine learning due to its higher regularization performance at a much faster speed (Huang, 2006) [16], [17] There are a lot of extension has been made on this particular algorithm in the last few years [18], [19], [20]. In 2014, Huang et al. Proposed semi-supervised ELM (SS-ELM) and un-supervised ELM which is applied in different application domains including object recognition, USPS clustering, and so on [21]. This is first time, we have

978-1-5386-3200-0/17/$31.00 ©2017 IEEE

applied the USELM technique for network intrusion detection for cyber security.The main contributions of this work can be summarized as:

- AE and RBM are used to extract the features and iterative k-means clustering is used for final detection of attacks.
- Evaluation of the performance of network intrusion detection and classification for KDD-Cup 99.
- The accuracy of proposed approaches are compared against other machine learning techniques of K-means, USELM.

The remainder of this paper is organized as follows: proposed intrusion detection system for cyber security is discussed in Section II. The details discussion of algorithms and implementation details are given in Section III. Section IV describes experimental results of network intrusion detection. Finally, conclusions are drawn in Section V.

II. Proposed cyber security system

Since the SNORT, which is rule based approach network intrusion detection and provides almost 100% accuracy based on the rules that already included in the dictionary for intrusion detection problem. However, this type of rules based approach does not work properly for new type of attack which is required new rules to identify. A single attack on the other hand can be caused of billions of dollars damage, for example: slammer worm which took place in January, 2003, as a result a lot of airline flights have been canceled, US 911 system has affected, around 13,000 Bank of America ATMs were failed. According to the U.S Federal Bureau of Investigation (FBI), this attack causes around 1 billion or equivalent dollars of productive loss. This is where the deep learning based unsupervised intrusion detection can ensure around 92% detection for cyber security. According to the proposed model, when SNORT fails to predict new type of attack, the system receives the packets with new type of attack is called zaro-day attach. After receiving the packets, the preprocessing techniques are applied on the input samples. Then the deep learning techniques including AE and RBM are used to proper data encoding with dimensionality reduction. Auto-encoder is a fundamental deep learning approach which is aim to transform the inputs data into outputs with the minimum possible amount of distortion. The concept of auto-encoder first introduced by Hinton [22], [23] emphasizing the problem of Backpropagation without teacher which is the fundamental paradigms of unsupervised deep learning.

Then iterative k-means clustering is used to finally label the data with respect to the normal behavior of sample from exiting dataset. Based on the outcomes from the clustering approach, the new label has been assigned with respect to the labels of attack and non-attack (normal packets). After assigning the label of new attack, the supervised deep neural network is applied for training the model again to update the model. Thus the system can automatically detect same type of attack in the future. Finally, the security alerts will be provided to the end user. To deploy this type of cyber security system, the online learning or life-long learning system is required to label data automatically. The cyber security is one of the problem, which is required online learning for ensuring or updating the security system with respective to the time. Since in every second, there are different types of new attacks have been taking place on the cyber world. Our motivation is to design such a system which can learn and update the security system automatically. As a part of that, here we present only the clustering part for dealing with unknown type of attacks which is the most challenging part for this proposed network security system. Overall diagram of the proposed intrusion detection and classification system is given in Fig. 1.

III. Implementation details

There are different type of recently developed deep learning and machine learning approach are implemented for this proposed intrusion detection system. The technical details are discussed in the following sections.

A. Auto-Encoder (AE)

An auto encoder is a deep neural network approach used for unsupervised feature learning with efficient coding. The main objective of auto encoder to learn and representation (encoding) of data, typically for the purpose of data dimensionality reduction. This auto encoder technique is consist with two parts: the encoder and the decoder. In the encoding phase, the inputs samples maps usually in the lower dimension of features space with very constructive features. This approach can be continued to reach to the desired feature dimensional space. Whereas in the decoding phase, we regenerates actual feature from lower feature dimension with reverser processing [22, 23]. The conceptual diagram of auto-encoder is shown in Fig. 2. The encoder and decoder transition can be represented with ϕ and φ

$$\phi : \chi \rightarrow F \tag{1}$$

$$\varphi : \mathcal{F} \rightarrow \chi \tag{2}$$

$$\phi, \varphi = argmin_{\phi,\varphi} \|X - (\phi, \varphi\|^2 \tag{3}$$

If we considered a simplest auto encoder with one hidden layer, where the input is $xR^d = X$which is mapped onto $XR^p = F$, then the expression can be written for the forward operation as follows:

$$z = \sigma_1(W * x + b) \tag{4}$$

Where W is the weight matrix and b is bias. σ_1 represents a element wise activation function such as sigmoid or a rectified linear unit (RLU). Let is considered z is again mapped or reconstructed onto $x^{'}$ which is the same dimension of x. The reconstruction can be express as

$$x^{'} = \sigma_2(W^{'} * z + b^{'}) \tag{5}$$

This techniques can be also train with minimize the reconstruction errors:

$$\mathcal{L} = \|x - x^{'}\|^2 == \|x - \sigma_2(W^{'} * \sigma_1(W * x + b) + b^{'})\|^2 \tag{6}$$

Usually the feature space of $\mathcal{F}$ has the lower dimension of input feature space $\mathcal{X}$ which can stated as the compress representation of input sample. In case of multilayer auto encoder, the same operation will be incorporated as required with in the encoding and decoding phases.

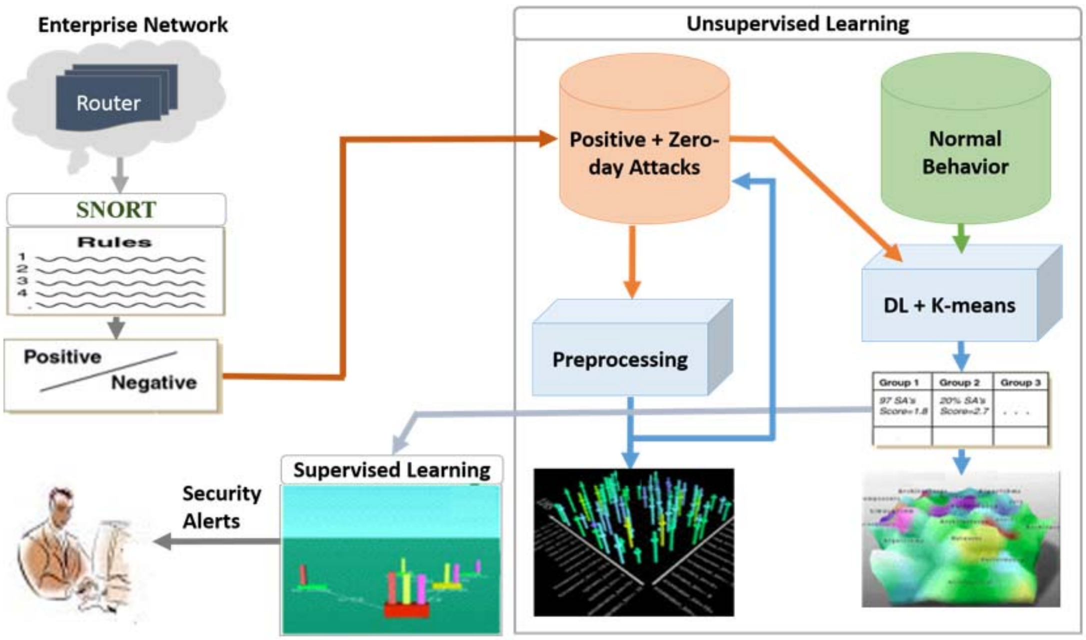


Fig. 1. Overall proposed intrusion detection system for cyber security using deep learning.

B. Restricted Boltzmann Machine (RBM)

The training set can be modeled using a two-layer network called a Restricted Boltzmann Machine in which, binary pixels are connected stochastically, binary feature detectors use symmetric weighted connections[24], [25]. RBM is an energy-based undirected generative model that uses a layer of hidden variables to model a distribution over visible variables. The undirected model for the interactions between the hidden and visible variables are used to ensure that the contribution of the likelihood term to the posterior over the hidden variables is approximately factorial which greatly facilitates inference [26]. The conceptual diagram of RBM is shown in Fig. 3.

Energy-based model means that the probability distribution over the variables of interest is defined through an energy function. It is composed from a set of observable variables s $V = \{v_i\}$ and a set of hidden variables $H = \{h_i\}$ where i is node in the visible layer, j node in the hidden layer. It is restricted in the sense that there are no visible-visible or hidden-hidden connections. The values correspond to visible units of the RBM because their states are observed; the feature detectors correspond to hidden units. A joint configuration energy of the visible and hidden units has an energy (Hopfield, 1982) given by:

$$E(v,h;\theta) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i h_j w_{ij} \quad (7)$$

Where v_i h_j are the binary states of visible unit i and hidden unit j, a_i, b_j are their biases and w_{ij} is the weight between them. The network assigns a likelihood to every possible pair of a visible and a hidden vector via this energy function:

$$p(v,h) = \frac{1}{Z} e^{-E(v,h;\theta)} \quad (8)$$

Where Z is the partition function, which is given by summing over all possible pairs of visible and hidden vectors:

$$Z = \sum_{v,h} e^{-E(v,h)} \quad (9)$$

The probability for the visible layer given by the summation over all possible hidden vectors:

$$p(v) = \frac{1}{Z} \sum_h e^{-E(v,h;\theta)} \quad (10)$$

The probability of the network assigns to a training input sample can be elevated by changing the weights and biases to lower the energy of that sample and to raise the energy of other input samples, especially those have low energies and that make a big role to the partition function. The derivative of the log likelihood of a training vector with respect to a weight can be written as

$$\frac{\partial \log p(v)}{\partial w_{ij}} = \langle v_j h_j \rangle_d - \langle v_j h_j \rangle_m \quad (11)$$

where the angle brackets represents expectations under the distribution specified by the subscript that follows. This leads to a very simple learning rule for performing stochastic steepest ascent in the log probability of the training data:

$$w_{ij} = \epsilon \frac{\partial \log p(v)}{\partial w_{ij}} \quad (12)$$

where ϵ is a learning rate because there are no direct connections between hidden units in an RBM, it is very easy to get an unbiased sample of $\langle v_j h_j \rangle_d$ data. Given a randomly selected training image,v, the binary state, h_j ,of each hidden unit, , j is set to 1 with probability

$$p(h_j=1|v)=\sigma\left(b_j+\sum_i v_i w_{ij}\right) \tag{13}$$

where $\sigma(\cdot)$ is the logistic sigmoid function. Due to no have direct connections between visible units in an RBM, visible and hidden layer are then an unbiased sample. An unbiased sample of the state of a visible unit, given a hidden vector can be calculated by the following equation.

$$p(v_i=1|h)=\sigma\left(a_i+\sum_j h_j w_{ij}\right) \tag{14}$$

Getting an unbiased sample of$\langle v_j h_j\rangle_m$ model, however, is much more difficult. It can be done by starting at any random state of the visible units and performing alternating Gibbs sampling for a very long time. One iteration of alternating Gibbs sampling consists of updating all of the hidden units in parallel using equation 13 followed by updating all of the visible units in parallel using equation 14. A much faster learning procedure was proposed by Hinton in 2002. This starts by setting the states of the visible units to a training vector. Then the binary states of the hidden units are all computed in parallel using equation 13. Once binary states have been chosen for the hidden units, a reconstruction is produced by setting each v_i to 1 with a probability given by equation 14. The change in a weight is then given by

$$\triangle w_{ij}=\epsilon\left(\langle v_j h_j\rangle_d-\langle v_j h_j\rangle_r\right) \tag{15}$$

In this work, RBM are used to feature extraction and dimensionality reduction on KDD-99 dataset.

C. Unsupervised Extreme Learning Machine

The Extreme Learning Machine (ELM) is a supervised learning approach which consists with a single layer of perceptron. The weight and bias of input to hidden layer is assigned randomly instead of learning parameters with iterative process. The single layer feedforward network generalized with L hidden nodes of the function of ELM which can be express by the following equation [16], [17].

$$f_L(x)=\sum_{i=1}^{L}\beta_i g_i(x)=\sum_{i=1}^{L}\beta_i G(a_i,b_i,x), x\varepsilon R^d, \beta_i\varepsilon R^m \tag{16}$$

Where the weight matrix connecting with input is represented with $a_i=[a_{i1},a_{i2},,a_{in}]^T$ to the ith hidden node, b_i is the $i^t h$ bias of the hidden node , output function is defined with g_i, i.e., activation function $G(a_i,b_i,x)$ of the ith hidden node, and the weight matrix which connect with hidden to output are represented with $\beta_i=[\beta_{i1},\beta_{i2},,\beta_{im}]^T$. In short, we can define ELM with the following equations:

$$H\beta=T \tag{17}$$

$$\beta^2=H^{+}T \tag{18}$$

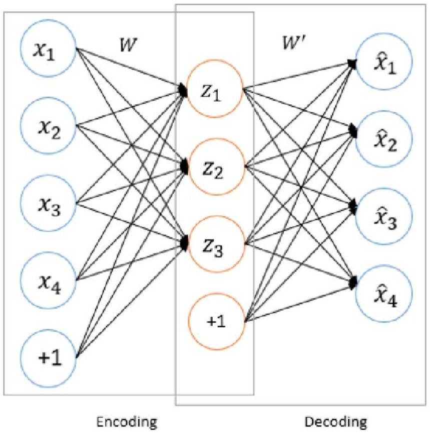


Fig. 2. Diagram for Auto encoder with encoding and decoding phases.

H is the hidden layer output matrix of the SLFN, the output weight matrix between hidden and output layer. Where T is the target and H^+ is the Moore-Penrose generalized inverse of matrix H. The unsupervised learning with ELM (US-ELM) is proposed by Huang et al. makes use of least square method to obtain the embedded matrix which can be used to cluster [21]. The formulation of US-ELM is from semi-supervised ELM (SS-ELM). The substitution of the constrains with objective function can be define as follows:

$$min_{\beta\varepsilon R^{n_h\times n_0}}\frac{1}{2}\|\beta\|^2+\frac{1}{2}\|C^{1/2}(\hat{Y}-H\beta)\|^2+\frac{\lambda}{2}T_r(\beta^T H^T LH\beta) \tag{19}$$

Where Y is the training target, C is a constant diagonal matrix. The above equation can be simplified

$$min_{\beta\varepsilon R^{n_h\times n_0}}\|\beta\|^2+\lambda T_r(\beta^T H^T LH\beta) \tag{20}$$

The formulation of US-ELM is given by Haung. The theoretical details is available in [21].The above equation can be simplified and written as :

$$st.(\beta H^T LH\beta=I_{n_0}) \tag{21}$$

D. Iterative K-mean clustering

K-means clustering is a very popular and classical clustering algorithm [27]. After randomly assigning the cluster center at the initial stage for example to k clusters, the center of cluster are updated and examples are assigned to the clusters with the closest centers. The process is repeated iteratively until the cluster centers do not reach significantly change. Once the cluster centers are fixed, the mean distance of an example to cluster centers is used as the scores. Based on the score, the new input samples are clustered. In this implementation, k is the free parameter.

IV. Result and discussion

In the experiment, we have used different deep learning techniques including AE ad RBM for feature extraction and dimensionality reduction. The KDD-99 dataset is used for the experimental evaluation of the proposed system [28]. The database details is given in the following section. The entire experiments have been conducted on a desktop computer with an Intel Core 2 Duo CPU E86 @ 3.33 GHz processor and 12GB of RAM to evaluate the processing time in MATLAB (R2015a).

A. *Database*

The NSL-KDD intrusion detection database is based on the 1998 DARPA initiative to provide designers of intrusion detection systems (IDS) for cyber security benchmark. The NSL-KDD intrusion detection dataset is published by DARPA in 1998 for very first time which is very popular benchmark for network intrusion detection for cyber security.

However, there are different version of this dataset is released after that. In this implementation, we have used KDD-99 which is published in 1999. . In the database, each NSL-KDD sample contains 41 features (e.g., protocol type, service, and flag) which contains attack and no attack samples. The example training sample is shown in Fig. 4. There are four specific types of attacks are defined depending on the sample of dataset which is shown the Fig.5. This table also shows the data distribution of the KDD dataset. These attacks fall into one of the five categories listed below:

- Normal: data with no attack.
- Denial of Service (DoS): which tries to prevent actual users from using a service.
- Probe: this type of attacker tries to gain information about the target host.
- Remote to Local (R2L): tries to gain access where attacker does not have an account on the victim machine.
- User to Root (U2R): attacker has local access to the victim machine and tries to gain root user privileges.

However, in this implementation, we have considered all of the attack as single attack category.

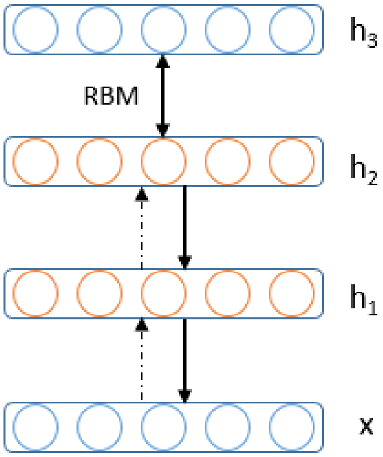
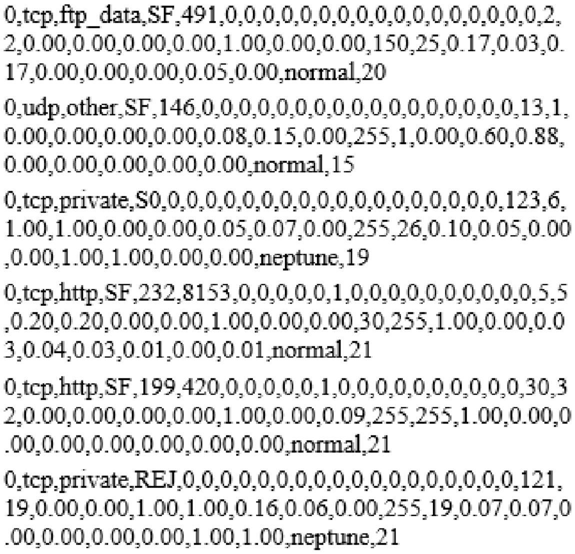


Fig. 3. Block diagram for stacked of RBMs.

There are about 25000 sample are used during the training for this proposed technique. We have used numerically encoded and normalized data samples for feature extraction with different AE and RBM.

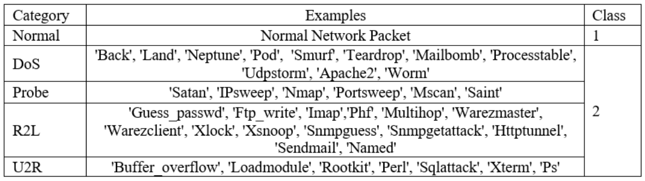


```
0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,
2,0.00,0.00,0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.
17,0.00,0.00,0.00,0.05,0.00,normal,20
0,udp,other,SF,146,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13,1,
0.00,0.00,0.00,0.00,0.08,0.15,0.00,255,1,0.00,0.60,0.88,
0.00,0.00,0.00,0.00,0.00,normal,15
0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,123,6,
1.00,1.00,0.00,0.00,0.05,0.07,0.00,255,26,0.10,0.05,0.00
,0.00,1.00,1.00,0.00,0.00,neptune,19
0,tcp,http,SF,232,8153,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,5,5
,0.20,0.20,0.00,0.00,1.00,0.00,0.00,30,255,1.00,0.00,0.0
3,0.04,0.03,0.01,0.00,0.01,normal,21
0,tcp,http,SF,199,420,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,30,3
2,0.00,0.00,0.00,0.00,1.00,0.00,0.09,255,255,1.00,0.00,0
.00,0.00,0.00,0.00,0.00,0.00,normal,21
0,tcp,private,REJ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,121,
19,0.00,0.00,1.00,1.00,0.16,0.06,0.00,255,19,0.07,0.07,0
.00,0.00,0.00,0.00,1.00,1.00,neptune,21
```

Fig. 4. Samples packets from the NSL-KDD dataset.

B. *Preprocessing*

The most of the fields of the input sample including network protocols, services, flag, etc. are represented with strings. The string are encoded with numerical values which is shown in Table I as an example. The table represents the corresponding values for the individual strings of network protocol.



Category	Examples	Class
Normal	Normal Network Packet	1
DoS	'Back', 'Land', 'Neptune', 'Pod', 'Smurf', 'Teardrop', 'Mailbomb', 'Processtable', 'Udpstorm', 'Apache2', 'Worm'	2
Probe	'Satan', 'IPsweep', 'Nmap', 'Portsweep', 'Mscan', 'Saint'	
R2L	'Guess_passwd', 'Ftp_write', 'Imap','Phf', 'Multihop', 'Warezmaster', 'Warezclient', 'Xlock', 'Xsnoop', 'Snmpguess', 'Snmpgetattack', 'Httptunnel', 'Sendmail', 'Named'	
U2R	'Buffer_overflow', 'Loadmodule', 'Rootkit', 'Perl', 'Sqlattack', 'Xterm', 'Ps'	

Fig. 5. Categories of network attacks in this implementation.

All the remaining fields with string are encoded in a similar way. The input samples are then normalized using min-max normalization approach. After this transformation, some input data features end up being encoded as all zeros. The column with zeros elements are dropped from the input samples, thus the final feature dimension of input samples set from 41 to 39. Therefore, only 39 features are used as inputs to the unsupervised deep learning approach. The network structure is consisted with 39 10050259 or 3. The output of this network is encoded to three or nine dimensional feature space which is very low dimensional feature space compare to the input dimension of 39.

The advantages of feature extraction with un-supervised deep learning approaches is not only that it ensures the faster calculation but also it provide better representation of

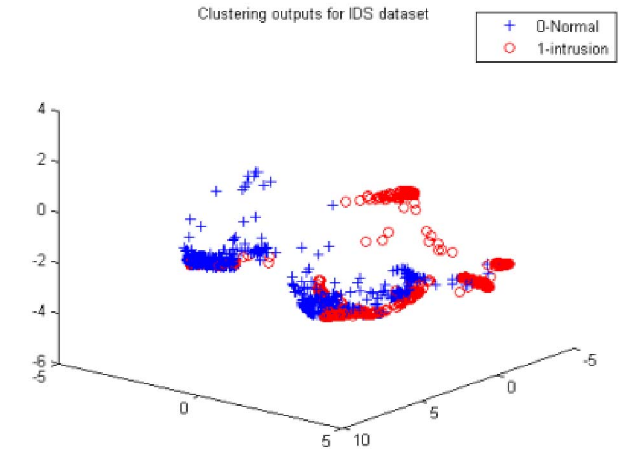


Fig. 6. Experimental clustering results for intrusion detection with RBM and K-means.

TABLE I. PROTOCOL NAME WITH CORRESPONDING VALUES

Protocol name	TCP	UDP	ICMP	Other protocols
Value	1	2	3	0

TABLE II. DETECTION ACCURACY FOR DIFFERENT DEEP AND MACHINE LEARNING TECHNIQUES

Methods	Number of features	Detection accuracy
K-means	39	87.72%
US-ELM	39	89.17%
AE+K-means	3	90.86%
AE+K-means	9	91.86%
RBM+eans	3	90.86%
RBM+K-means	9	92.12%

feature learning. As a result, it provides better recognition accuracy. The experiment are conducted for 50 iterations for binary classification with only 3 and 9 features as inputs. The following Fig.6 shows the experimental outputs with RBM and iterative k-means clustering for only 3 input features. From the figure, it can be clearly observed that the attacks and non-attack are clustered properly.

In this experiment, we have evaluated the iterative k-means clustering, deep learning approaches with k-means, and Un-Supervised ELM. The overall recognition accuracy is shown in Table III. The main significant of this proposed approach is that when the SNORT fail to detect the new type of cyber-attack, this approach can at least ensure around 93% security for detecting new type of cyber-attack with un-supervised deep leaning technique with k-means clustering.

V. CONCLUSION

In this paper, we proposed a new approach for the first time implementation of network intrusion detection using unsupervised deep learning approaches with iterative K-means clustering. In addition, we have tested on unsupervised ELM, and only K-means clustering approaches. From empirical evaluation on KDD-Cup 99 benchmark, it is observed that the deep learning approach of RBM and AE with k-means clustering show around 92.12% and 91.86% accuracy for network intrusion detection respectively. RBM with K-means clustering provides around 4.4% and 2.95% better detection accuracy compare to K-means and USELM techniques respectively. In the future, we would like to deploy complete proposed system for network intrusion detection for cyber security with online learning approach.

REFERENCES

[1] 1. Lee, Wenke, and Salvatore J. Stolfo. "A framework for constructing features and models for intrusion detection systems." ACM transactions on Information and system security (TiSSEC) 3.4 (2000): 227-261.

[2] 2. Zanero, Stefano, and Sergio M. Savaresi. "Unsupervised learning techniques for an intrusion detection system." Proceedings of the 2004 ACM symposium on Applied computing. ACM, 2004.

[3] 3. Zanero, Stefano, and Giuseppe Serazzi. "Unsupervised learning algorithms for intrusion detection." Network Operations and Management Symposium, 2008. NOMS 2008. IEEE. IEEE, 2008.

[4] 1. Nassar M, al Bouna B, Malluhi Q(2013) Secure outsourcing of network flow data analysis, In : Big Data (BigData Congress), 2013 IEEE International Congress on IEEE, Santa Clara, CA, USA. Pp 431-432.

[5] 5. Zuech, Richard, Taghi M. Khoshgoftaar, and Randall Wald. "Intrusion detection and big heterogeneous data: A survey." Journal of Big Data 2.1 (2015): 1-41.

[6] 6. Y. Bengio, Learning deep architectures for AI, Foundations and Trends in Machine Learning, vol. 2, iss. 1, pp. 1-127, 2009.

[7] 7. G. E. Hinton, S. Osindero, Y. Teh, A fast learning algorithm for deep belief nets. Neural Computation vol. 18, no. 7, pp. 1527-1554, 2006.

[8] 8. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (NIPS), pages 10971105, 2012.

[9] 9. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. arXiv preprint arXiv:1409.4842, 2014.

[10] 10. M. Lin, Q. Chen, and S. Yan. Network in network. In International Conference on Learning Representations (ICLR), 2014.

[11] 11. A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2014.

[12] 12. K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In British Machine Vision Conference, 2014.

[13] 13. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei- Fei. Imagenet: A large-scale hierarchical image database. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 248255, 2009

[14] 14. Md Zahangir Alom, VenkataRamesh Bontupalli, and Tarek M. Taha. "Intrusion detection using Deep Belief Networks." Aerospace and Electronics Conference (NAECON), 2015 National. IEEE, 2015

[15] 15. Zahangir Alom, Venkata Ramesh Bontupalli, and Tarek M. Taha. "Intrusion Detection Using Deep Belief Network and Extreme Learning Machine." Artificial Intelligence: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications (2016): 357.

[16] 16. Huang, G. B., Zhu, Q. Y., and Siew, C. K. (2006). Extreme learning machine: theory and applications. Neurocomputing, 70(1), 489-501.

[17] 17. Huang, G. B., Chen, L., and Siew, C. K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. Neural Networks, IEEE Transactions on, 17(4), 879-892.

[18] 18. Huang, G. B., Wang, D. H., and Lan, Y. (2011). Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics, 2(2), 107-122.

[19] 19. Md. Zahangir Aalom, Paheding Sidike, Tarek M. Taha, Vijayan K. Asar State Preserving Extreme Learning Machine for Face Recognition, in Proc. of IEEE International Joing Conference on Neural Network (IJCNN), pp. 1-7.

[20] 20. Md. Zahangir Aalom, Paheding Sidike, Tarek M. Taha, Vijayan K. Asar State preserving extreme learning machine: A monotonically increasing learning approach Neural processing letters, vol. 45, issue 2, pp. 703-725, April, 2017

[21] 1. Huang, Gao, et al. ”Semi-supervised and unsupervised extreme learning machines.” IEEE transactions on cybernetics 44.12 (2014): 2405-2417.

[22] 22. Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. ”Reducing the dimensionality of data with neural networks.” science 313.5786 (2006): 504-507

[23] 23. D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In Parallel Distributed Processing. Vol 1: Foundations. MIT Press, Cambridge, MA, 1986

[24] 24. Freund, Yoav, and David Haussler. ”Unsupervised learning of distributions of binary vectors using two layer networks.” (1994).

[25] 25. Smolensky, Paul. Information processing in dynamical systems: Foundations of harmony theory. No. CU-CS-321-86. Colorado University at Boulder, Dept. of Computer Science, 1986.

[26] 26. Larochelle, Hugo, and Yoshua Bengio. ”Classification using discriminative restricted Boltzmann machines.” Proceedings of the 25th international conference on Machine learning. ACM, 2008.

[27] 27. Duda, R., P.E.Hart, D.G.Stork: Pattern classication. second edn. John Wiley and Sons (2001)

[28] 28. KDD’99 dataset (2010), http://kdd.ics.uci.edu/databases, Irvine, CA, USA.