Unsupervised Feature Learning using a Novel Non Symmetric Deep Auto-encoder (NDAE) For NIDPS Framework

Vinav S. Autkar[1], Prof. P. R. Chandre[2]
Department of Computer Engineering,
Smt. Kashibai Navale College of Engineering, Savitribai Phule, Pune University, Pune, India.

Abstract- Repetitive and in material highlights in data have caused a whole deal issue in system traffic classification. Lately, one of the fundamental concentrations inside (Network Intrusion Detection System) NIDS investigate has been the use of machine learning and shallow learning strategies. This paper proposes a novel deep learning model to empower NIDS activity inside present day systems. The model demonstrates a blend of deep and shallow learning, prepared to do accurately investigating a wide-scope of system traffic. The system approach proposes a Non-symmetric Deep Auto-Encoder (NDAE) for unsupervised feature learning. Also, furthermore proposes novel profound learning order show constructed using stacked NDAEs. Our proposed classifier has been executed in Graphics preparing unit (GPU)- engaged TensorFlow and surveyed using the benchmark utilizing KDD Cup '99 and NSL-KDD datasets. The execution assessed organizes interruption location examination datasets, especially KDD Cup 99 and NSL-KDD dataset. However the to cover-up the Limitation of KDD dataset in proposed system WSN-DS dataset has been used. The commitment work is to execute interruption counteractive action framework (IPS) contains IDS usefulness however progressively complex frameworks which are fit for making quick move so as to forestall or diminish the vindictive conduct.

Keywords - Deep learning; Anomaly detection; Auto-encoders; Neural Network; Network security.

I. INTRODUCTION

One of the real difficulties in system security is the arrangement of a powerful and successful Network Intrusion Detection System (NIDS). Regardless of the critical advances in NIDS innovation, most of arrangements still work utilizing less-able mark based strategies, rather than irregularity recognition methods. The present issues are the current systems prompts ineffectual and wrong discovery of assaults. There are three fundamental confinements like, volume of system information, inside and out observing and granularity required to enhance adequacy and precision lastly the quantity of various conventions and assorted variety of information crossing. The primary focus on developing NIDS has been the use of machine learning and shallow learning techniques. The underlying profound learning research has shown that its unrivaled layer-wise element learning can better or possibly coordinate the execution of shallow learning procedures. It is equipped for encouraging a more profound examination of system information and quicker recognizable proof of any peculiarities. In this paper, we propose a new deep learning model for NIDPS for present day systems.

Thus refer the papers [1],[2],[3] and [4] and study all the details about how to work Deep Belief Network (DBN) and Restricted Boltzmann Machine (RBM) for intrusion detection. The contribution of intrusion prevention system refers the paper [12] using for rule generation technique and according to take the action against it.

A. Motivation:

- A new NDAE technique for unsupervised feature learning, which unlike typical auto-encoder approaches provides non-symmetric data dimensionality reduction. Hence, our technique is able to facilitate improved classification results when compared with leading methods such as Deep Belief Networks (DBNs).
- A novel classifier model that utilizes stacked NDAEs and the RF classification algorithm. By combining both deep and shallow learning techniques to exploit their respective strengths and reduce analytical overheads. We are able to better or at least match results from similar research, whilst significantly reducing the training time.

II. RELATED WORK

The paper [1] focuses on deep learning methods which are inspired by the structure depth of human brain learn from lower level characteristic to higher levels concept. It is because of abstraction from multiple levels, the Deep Belief Network (DBN) helps to learn functions which are mapping from input to the output. The process of learning does not dependent on human-crafted features. DBN uses an unsupervised learning algorithm, a Restricted Boltzmann Machine (RBM) for each layer. Advantages are: Deep coding is its ability to adapt to changing contexts concerning data that ensures the technique conducts exhaustive data analysis. Detects abnormalities in the system that includes anomaly detection, traffic identification. Disadvantages are: Demand for faster and efficient data assessment.

The main purpose of [2] paper is to review and summarize the work of deep learning on machine health monitoring. The application of machine learning in health monitoring systems are reviewed mainly from the following aspects: Autoencoder (AE) and its variants, Restricted Boltzmann

Machines and its variants which includes Deep Belief Network (DBN) and Deep Boltzmann Machines (DBM), Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). Advantages are: DL-based MHMS do not require extensive human labor and expert knowledge. The applications of deep learning models are not restricted to specific kinds of machines. Disadvantages are: The performance of DL-based MHMS heavily depends on the scale and quality of datasets.

Proposes the use of a stacked denoising autoencoder (SdA), which is a deep learning algorithm, to establish an FDC model for simultaneous feature extraction and classification. The SdA model [3] can identify global and invariant features in the sensor signals for fault monitoring and is robust against measurement noise. An SdA is consisting of de-noising autoencoders that are stacked layer by layer. This multilayered architecture is capable of learning global features from complex input data, such as multivariate time-series datasets and high-resolution images. Advantages are: SdA model is useful in real applications. The SdA model proposes effectively learn normal and fault-related features from sensor signals without preprocessing. Disadvantages are: Need to investigate a trained SdA to identify the process parameters that most significantly impact the classification results.

Proposes a novel deep learning-based recurrent neural networks (RNNs)model [4] for automatic security audit of short messages from prisons, which can classify short messages(secure and non-insecure).In this paper, the feature of short messages is extracted by word2vec which captures word order information, and each sentence is mapped to a feature vector. In particular, words with similar meaning are mapped to a similar position in the vector space, and then classified by RNNs. Advantages are: The RNNs model achieves an average 92.7% accuracy which is higher than SVM. Taking advantage of ensemble frameworks for integrating different feature extraction and classification algorithms to boost the overall performance . Disadvantages are: It is apply on only short messages not large-scale messages.

Signature-based features technique as a deep convolutional neural network [5] in a cloud platform is proposed for plate localization, character detection and segmentation. Extracting significant features makes the LPRS to adequately recognize the license plate in a challenging situation such as i) congested traffic with multiple plates in the image ii) plate orientation towards brightness, iii) extra information on the plate, iv) distortion due to wear and tear and v) distortion about captured images in bad weather like as hazy images. Advantages are: The superiority of the proposed algorithm in the accuracy of recognizing LP rather than other traditional LPRS. Disadvantages are: There are some unrecognized or miss-detection images.

In [6] paper, a deep learning approach for anomaly detection using a Restricted Boltzmann Machine (RBM) and a deep belief network are implemented. This method uses a one-hidden layer RBM to perform unsupervised feature reduction. The resultant weights from this RBM are passed to another RBM producing a deep belief network. The pre-trained weights are passed into a fine tuning layer consisting of a Logistic Regression (LR) classifier with multi-class soft-max. Advantages are: Achieves 97.9% accuracy. It produces a low false negative rate of 2.47%. Disadvantages are: Need to improve the method to maximize the feature reduction process in the deep learning network and to improve the dataset.

The paper [7] proposes a deep learning based approach for developing an efficient and flexible NIDS. A sparse autoencoder and soft-max regression based NIDS was implemented. Uses Self-taught Learning (STL), a deep learning based technique, on NSL-KDD - a benchmark dataset for network intrusion. Advantages are: STL achieved a classification accuracy rate more than 98% for all types of classification. Disadvantages are: Need to implement a real-time NIDS for actual networks using deep learning technique.

In [8] paper choose multi-core CPU's as well as GPU's to evaluate the performance of the DNN based IDS to handle huge network data. The parallel computing capabilities of the neural network make the Deep Neural Network (DNN) to effectively look through the network traffic with an accelerated performance. Advantages are: The DNN based IDS is reliable and efficient in intrusion detection for identifying the specific attack classes with required number of samples for training. The multi-core CPU's was faster than the serial training mechanism. Disadvantages are: Need to improve the detection accuracies of DNN based IDS.

In [9] paper, proposes a mechanism for detecting large scale network-wide attacks using Replicator Neural Networks (RNNs) for creating anomaly detection models Our approach is unsupervised and requires no labeled data. It also accurately detects network-wide anomalies without presuming that the training data is completely free of attacks. Advantages are: The proposed methodology is able to successfully discover all prominent DDoS attacks and *SYN Port* scans injected. Proposed methodology is resilient against learning in the presence of attacks, something that related work lacks. Disadvantages are: Need to improve proposed methodology by using stacked autoencoder deep learning techniques.

Based on the flow-based nature of SDN, we propose a flow-based anomaly detection system using deep learning. In [10] paper, apply a deep learning approach for flow-based anomaly detection in an SDN environment. Advantages are :It finds an optimal hyper-parameter for DNN and confirms the detection rate and false alarm rate. The model gets the performance with accuracy of 75.75% which is quite reasonable from just using six basic network features. Disadvantages are: It will not work on real SDN environment.

III. OPEN ISSUES

The present system traffic information, which are regularly enormous in size, present a noteworthy test to IDSs These

"Big Data" back off the whole location process and may prompt unsuitable grouping precision because of the computational troubles in taking care of such information. Machine learning innovations have been normally utilized in IDS. In any case, a large portion of the conventional machine learning innovations allude to shallow learning; they can't viably understand the gigantic interruption information order issue that emerges despite a genuine system application condition. Also, shallow learning is contradictory to wise examination and the foreordained necessities of high-dimensional learning with colossal information. Disadvantage: Computer frameworks and web have turned into a noteworthy piece of the basic framework. The present system traffic information, which are regularly gigantic in size, present a noteworthy test to IDSs. These "Big Data" back off the whole recognition process and may prompt inadmissible grouping precision because of the computational challenges in dealing with such information. Ordering a colossal measure of information for the most part causes numerous numerical troubles which at that point lead to higher computational complexity.

IV. SYSTEM OVERVIEW

The paper [1] proposes a novel deep learning model to enable NIDS operation within modern networks. The model proposes is a combination of deep and shallow learning, capable of correctly analyzing a wide-range of network traffic. More specifically, combine the power of stacking our proposed Non-symmetric Deep Auto-Encoder (NDAE) (deep learning) and the accuracy and speed of Random Forest (RF) (shallow learning). This paper proposes NDAE, which is an auto-encoder featuring non-symmetrical multiple hidden layers. NDAE may be 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 is able to refine the model by prioritizing the most descriptive features.

Fig. 1 shows the proposed system architecture of Network Intrusion Detection and Prevention System (NIDPS). The input traffic data is uses for WSN-DS dataset with 19 features.

The training dataset contains data preprocessing which includes two steps: Data transformation and data normalization. After uses two NDAEs arranged in a stack, which uses for selecting number of features. After that apply the Random Forest Classifier [9] for attack detection. Intrusion Prevention Systems (IPS) contains IDS functionality but more sophisticated systems which are capable of taking immediate action in order to prevent or reduce the malicious behavior. The intrusion prevention system is implementing with the help of Rule Status Monitoring Algorithm [12]. There are 8 rule actions when the attack detected or not, the system will take the action using following :

- 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:

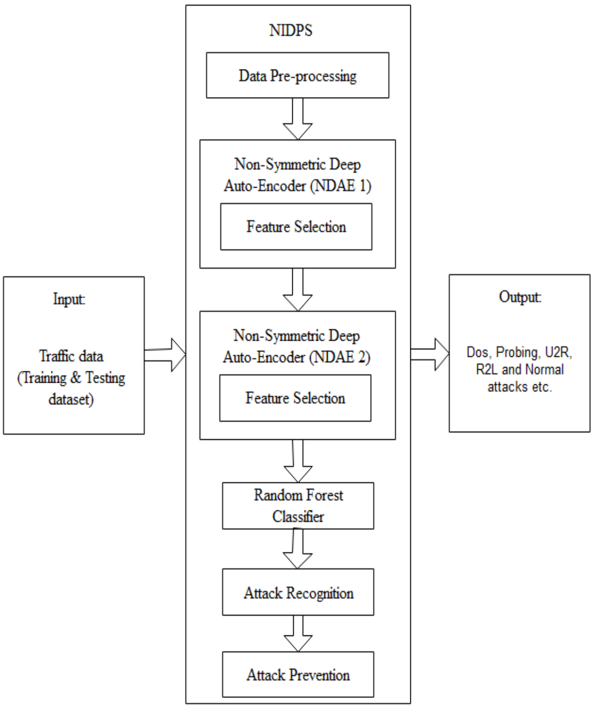
NIDPS
Data Pre-processing
Non-Symmetric Deep Auto-Encoder (NDAE 1)
Feature Selection
Input:
Traffic data (Training & Testing dataset)
Non-Symmetric Deep Auto-Encoder (NDAE 2)
Feature Selection
Output:
Dos, Probing, U2R, R2L and Normal attacks etc.
Random Forest Classifier
Attack Recognition
Attack Prevention

Figure 1: Proposed System Architecture

B. Mathematical Model:

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

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

Where,

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

T is m samples with n column attributes; x_{ij} is the jth column attribute in ith sample, μ_T and μ_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 μ_T and μ_T as follows:

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

ii). Feature Selection - NDAE is an auto-encoder featuring *non-symmetrical* multiple hidden layers. The proposed NDAE [1] 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:

$$\square_i = \sigma(W_i . \square_{i-1} + b_i); \; i = \overline{1,n}, \qquad (3)$$

Here, $\square_0 = x$, $\sigma$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.

$$y = \sigma(W_{n+1} . \square_n + b_{n+1}) \qquad (4)$$

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).

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

C. Algorithms

i). Restricted Boltzamine Machine Algorithm - $x_1$is a sample from the training distribution for the RBM

$\in$ 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: $Q(\square_{2_i} = 1 | x_2)$is the vector with elements $Q(\square_{2_i} = 1 | x_2)$

```
Step 1: for all hidden units i do
Step 2: compute $Q(\square_{1_i} = 1 | x_1)$(for binomial units, $sigm(c_i + \sum_j W_{ij} x_{1j})$)
Step 3: sample $\square_{1_i} \in \{0,1\}$from $Q(\square_{1_i} | x_1)$
Step 4: end for
Step 5: for all visible units j do
Step 6: compute $P(x_{2_j} = 1 | \square_1)$(for binomial units, $sigm(b_j + \sum_i W_{ij} \square_{1i})$)
Step 7: sample$x_{2_j} \in \{0,1\}$from $P(x_{2_j} = 1 | \square_1)$
Step 8: end for
Step 9: for all hidden units j do
Step 10: compute $Q(\square_{2_i} = 1 | x_2)$(for binomial units, $sigm(c_i + \sum_j W_{ij} x_{2j})$)
Step 11: end for
Step12: $W \longleftarrow W + \epsilon(\square_1 x_1' - Q(\square_{2_i} = 1 | x_2) x_2')$
Step 13: $b \longleftarrow b + \epsilon(x_1 - x_2)$
Step 14: $c \longleftarrow c + \epsilon(\square_1 - Q(\square_{2_i} = 1 | x_2))$
```

ii). 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 mean-field approximation instead of stochastic sampling

```
Step 1: for k = 1 to $\ell$do
Step 2: initialize $W^k = 0, b^k = 0, c^k = 0$
Step 3: while not stopping criterion do
Step 4: sample $\square^0 = x$ from $\hat{P}$
Step 5: for$i = 1$ to $k - 1$do
Step 6: if mean_field_computationthen
Step 7: assign $\square_j^i$ to $Q(\square_j^i = 1 | \square^{i-1}$, for all elements j of $\square^i$
Step 8: else
Step 9: assign $\square_j^i$ to $Q(\square_j^i | \square^{i-1}$, for all elements j of $\square^i$
Step 10:end if
Step 11:end for
Step 12: RBM update($\square^{k-1}, \in, W^k, b^k, c^k$) {thus providing $Q(\square^k | \square^{k-1})$for future use}
Step 13: end while
Step 14: end for
```

iii). Random Forest Classifier

1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
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*.
3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. 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.
5. 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.

iv). Rule Status Monitoring Algorithm:

Variables: The following variables are used: i, the rule set name index; j, the rule index; k, the rule action index; m, the number of rule sets; n, the number of rule actions; off_counter, counts of the number of rules that are off; on_counter, counts the number of rules that are off; ON, array containing the line numbers of rules that are on for a given action and rule set; OFF, array containing the line numbers of rules that are off for a given action and rule set.

Algorithm:

```
Step 1.  Obtain and store the name of each rule set into a file named rule sets
Step 2.  Determine the total number of rule sets, m, in the rule sets file
Step 3.  Assume that each rule is stored on a separate and single line index by j
Step 4.  Assume that there are n possible rule actions stored in array action
Step 5.  Read the rule sets file
Step 6.  For(k=1:n)
Step 7.  Select a rule action of interest
Step 8.  For(i=1:m)
Step 9.  Obtain the name for rule set, $S_i$, by reading line i of the rulesets file
Step 10. Read the file consisting of the rules for $S_i$,
Step 11. Initialize arrays ON and OFF to 0; to keep track of rules that are on and off, respectively
Step 12. Let j=1, on_counter = 0, off_counter = 0
Step 13. While (not end of file)
Step 14. Read rule[j]
Step 15. If (rule[j] uses action a[k])
Step 16. If( rule[j] is on)
Step 17. on_counter = on_counter + 1
Step 18. ON[on_counter] = j
Step 19. Else
Step 20. off_counter = off_counter + 1
Step 21. OFF[off_counter] = j
Step 22. j = j + 1
Step 23. End While
Step 24. If( on_counter != 0)
Step 25. Write ON array to file [‘on_rules’ + ruleset[i] + date + time]
Step 26. If(off_counter != 0)
Step 27. Write OFF array to file ‘off_rules’ + ruleset[i] + date + time
Step 28. End For
Step 29. End For
```

V. RESULT AND DISCUSSIONS

WSN-DS is the wireless dataset for researchers. The WSN-DS dataset contains total 19 attributes which are given below:

Table I WSN-DS Dataset Attributes

Total Attributes	
Id	SCH_R
time	Rank
Is_CH	DATA_S
who CH	DATA_R
Dist_To_CH	Data_Sent_To_BS
ADV_S	dist_CH_To_BS
ADV_R	send_code
JOIN_S	Consumed Energy
JOIN_R	Class
SCH_S	

For experimental set up, use Windows 7 operating system, Intel i5 processor, 4 GB RAM, 200GB Hard disk, Eclipse Luna JDK 8 tool and Tomcat server. To calculate the results, WSN-DS dataset is used. WSN-DS dataset is a real-time wireless dataset which gets information from router which contains node details and packet information. In training and testing dataset, there are 5types of attack which are subtypes of normal, probing, dos, u2r and r2l attacks.

The performance evaluation of network intrusion detection& prevention system (NIDPS) is held using different parameters. The Parameters are Detection Accuracy, False Positive Rate, and Detection Time.

WSN Trace Stacked Autoencoder + RF Classifier Result

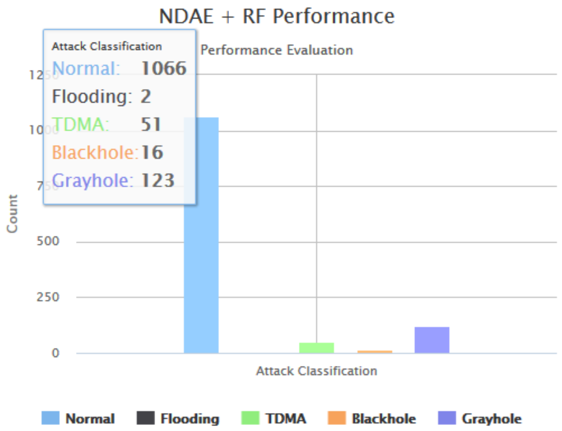

Figure 2: Performance analysis graph to count the attacks

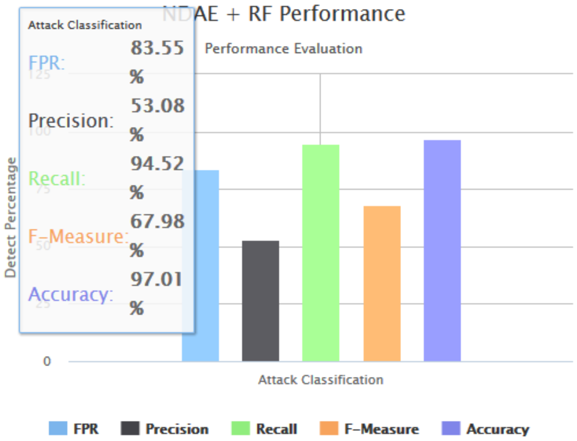

Figure 3: Performance graph of S-NDAE + RF classifier

VI. CONCLUSION

In this paper, mentioned the problems confronted by previous NIDS 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. Also implemented the Intrusion prevention system. 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 5% accuracy. So, there is need to further improvement of accuracy. And also further work on real-time network traffic and to handle zero-day attacks.

VII. REFERENCES

[1]. Nathan shone , trannguyenngoc, vu dinhphai , and qi sh, “a deep learning approach to network intrusion detection”,ieee transactions on emerging topics in computational intelligence, vol. 2, no. 1, february 2018

[2]. 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.

[3]. K. Alrawashdeh and C. Purdy, “Toward an online anomaly intrusion detection system based on deep learning,” in Proc. 15th IEEE Int. Conf. Mach. Learn. Appl., Anaheim, CA, USA, Dec. 2016, pp. 195–200.

[4]. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, “A deep learning approach for network intrusion detection system,” in Proc. 9th EAI Int.Conf. Bio-Inspired Inf. Commun. Technol., 2016, pp. 21–26. [Online]. Available: http://dx.doi.org/10.4108/eai.3-12-2015.2262516

[5]. S. Potluri and C. Diedrich, “Accelerated deep neural networks for enhanced intrusion detection system,” in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom.*, Berlin, Germany, Sep. 2016, pp. 1–8.

[6]. C. Garcia Cordero, S. Hauke, M. Muhlhauser, and M. Fischer, “Analyzing flow-based anomaly intrusion detection using replicator neural networks,” in *Proc. 14th Annu. Conf. Privacy, Security. Trust*, Auckland, New Zeland, Dec. 2016, pp. 317–324.

[7]. 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 *Proc. Int. Conf. Wireless Netw. Mobile Commun.*, Oct. 2016, pp. 258–263.

[8]. C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in *IEEE Access*, vol. 5, pp. 21954-21961, 2017.

[9]. Revathi, S &Malathi, A. (2013). A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection. International Journal of Engineering Research & Technology (IJERT). 2. 1848-1853.

[10]. N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, Feb. 2018.

[11]. Anomaly-based network intrusion detection: Techniques, systems and challenges Garcia-Teodoro P. Diaz-Verdejo J., Macia-Fernandez G., Vazquez E. (2009) *Computers and Security*, 28 (1-2), pp. 18-28.

[12]. Claude Turner*, Rolston Jeremiah, Dwight Richards, Anthony Joseph, “A Rule Status Monitoring Algorithm for Rule-Based Intrusion Detection and Prevention Systems”, Procedia Computer Science, ISSN: 1877-0509, Vol: 95, Page: 361-368, 2016

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

[14]. H. Lee, Y. Kim, and C. O. Kim, “A deep learning model for robust wafer fault monitoring with sensor measurement noise,” IEEE Trans. Semicond. Manuf., vol. 30, no. 1, pp. 23–31, Feb. 2017.

[15]. L. You, Y. Li, Y. Wang, J. Zhang, and Y. Yang, “A deep learning based RNNs model for automatic security audit of short messages,” in Proc. 16th Int. Symp. Commun. Inf. Technol., Qingdao, China, Sep. 2016, pp. 225–229.