International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-9 Issue-3 September 2020

Network Intrusion Detection using a Deep Learning Approach

V. V. Mandhare, D. R Pede, P. S. Vikhe

Abstract: ***At present situation network communication is at high risk for external and internal attacks due to large number of applications in various fields. The network traffic can be monitored to determine abnormality for software or hardware security mechanism in the network using Intrusion Detection System (IDS). 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 ID. Deep Learning (DL) is a subgroup of Machine Learning (ML) which is hinged on data description. The new model based on deep learning is presented in this research work to activate operation of IDS from modern networks. Model depicts combination of deep learning and machine learning, having capacity of wide range accurate analysis of traffic network. The new approach proposes non-symmetric deep auto encoder (NDAE) for learning the features in unsupervised manner. Furthermore, classification model is constructed using stacked NDAEs for classification. 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 consists IDS use, which are efficient in taking instant measures in order to stop or minimize the malicious actions.***

Key Words: Intrusion Detection System (IDS), Non- Symmetric Deep Auto-Encoder (NDAE), Deep Learning (DL), WSN Trace, Machine Learning (ML).

I. INTRODUCTION

Internets have a part in our daily life and is required weapon today. Internet has risen to multiple vices along with its boons, this lead in increased number of attacks. The organizations and individuals may be affected due to attacks. 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) [1] [2]. IDS identify interloper's activities that warn the integrity, availability, and confidentiality of resources. The distinct types of malicious network communications and computer systems usage can be detected by IDSs. However, performance of the task is not possible using traditional firewall. The detection of Intrusion is on the assumption of intruder's behavior and is different from a legal user [3]. In general, IDSs can diverge into two categories: 1) anomaly 2) signature detection, which are hinged on their detection methods [4]. In Anomaly detection, system classifies unknown behavior in traffic network studying normal behavior structures in traffic network. The Network traffic which is different in pattern from normal traffic is classified as an intrusion. In Signature detection, signatures attack is pre-installed in the IDS. The matching of pattern is carried out for traffic versus installed signatures to identify an intrusion which is present in network [5]. The current situation has reached to a conclusion that such approaches leads in false detection. In current years, the main focus within IDS research have been the appliance of machine learning methods like Decision Trees, Naive Bayes, Random Forest(RF) and Support Vector Machines (SVM) and many more [6]. The accuracy of detection have improved using these approaches. However, these approaches have some limitations, like expert knowledge is required to operate data; Interaction of high level of human expert is needed. Similarly, huge amount of data training is required for operation [7]. To locate the mentioned drawbacks, area of research recently switched towards deep learning. Deep learning is improved learning approach where 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]. The deep learning method has been utilized in this paper to activate NIDS operation within modern networks.

A. Motivation

- A novel NDAE method based on unsupervised learning feature, this is like auto-encoder technique gives non-symmetric dimensionality reduced data. This leads in better classification compare to Deep Belief Networks (DBNs) approach results.
- A new model classifier uses algorithms like stacked NDAEs and RF for classification. The combination of deep and machine learning methods allow to achieve their strengths and minimize analytical overheads.

Revised Manuscript Received on August 10, 2020.
***** Correspondence Author
V. V. Mandhare*, Associate Professor, Department of Computer Engineering, Pravara Rural Engineering College, Loni, Rahata, Ahmednagar, India.
D. R. Pede, Department of Computer Engineering, Pravara Rural Engineering College, Loni, Rahata, Ahmednagar, India.
P. S. Vikhe, Associate Professor, Department of Instrumentation and Control Engineering, Pravara Rural Engineering College, Loni, Rahata, Ahmednagar, India.

II. REVIEW OF LITERATURE

F. Farahnakian et al. [11] presented Deep Auto Encoder (DAE) model to train greedy layer-wise fashion, to avoid over fitting and local optima. In [11] Deep Auto Encoder hinged IDS (DAE-IDS) is suggested made up of four auto encoders, results AE at existing layer is used as AE input in following layer. Moreover, an AE at existing layer is trained prior the AE at following layer. After the 4 auto-encoders are trained, they have utilized a SoftMax layer for classifying the network traffic into normal data and attacks. 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. Ni GAO et al. [12] proposed a methodology which is hinged on multilayer DBN technique for identification of DoS attacks. DBN consists of large RBMs. The training of RBM is carried in process of advance learning. For learning RBM of next layer, trained features of current RBM are provided as input. The performance of DBN technique is tested using KDD CUP 1999 dataset. Detection precision of DBN method is better than ANN and SVM approach. S. Seo et al. [13] put forward comparison of intrusion detection rates of NIDS that utilize classification method and NIDS with trained data, noise and exception are eliminated using RBM. Noise and exception in KDD Cup '99 Data set are eliminated by assigning data to RBM and building new data. K. Alrawashdeh et al. [14, 15] suggested an approach of deep learning, to detect anomalies using RBM and deep belief network. One-hidden layer RBM is used for performance of unsupervised feature reduction. Resulting weights are passed from one RBM to other RBM which produces DBN. Pre-trained weights are proceeded to Logistic Regression (LR) classifier to classify the inputs into normal data and attacks. This model has performed better compare to previous methods of deep learning, performed by Li and Salama [14, 15] in terms of accuracy and speed detection. The detection rate achieved is 97.9% on total 10% KDD-CUP 1999 testing data-set. [16]. J. Kim et al. [17] developed the model for IDS using deep learning method. In [17] Long Short Term Memory (LSTM) framework was applied to an RNN and have trained IDS utilizing KDDCup-99 data-set. For training stage, data-set have been produced using extracted samples from KDDCup-99 data-set by analyzing them with another IDS classifiers; discovered attacks are efficiently detected via LSTM-RNN classifier. Since, it have better accuracy and detection rate, although rate of false alarms is little higher than others. The deep learning is sufficient for IDS based on performance tests. Y. Chuan-Long et al. [18, 19] design and implemented detection system hinged on recurrent NNs. Moreover, they have studied efficiency of model in binary and multi-class classifications. Furthermore, efficiency of multi-layer perception, Naives Bayes, SVMs and other methods of machine learning in multi-class classification on the benchmark KDD-Cup 1999 dataset were investigated. Y. Yu et al [20] suggested intrusion detection methodology which is hinged on Hybrid MLP/CNN. A hybrid MLP/CNN neural network is generated, to enhance the rate of detection in time-delayed attacks. A simulation test is organized utilizing DARPA 98 data-set. Hybrid MLP/CNN neural network have taken result from MLP as a chaotic neuron input. In a way chaotic neurons number have to be identical to number of output nodes. The result of distribution of input is tested using MLP, may be delivered and received by CNN coupled to output node of MLP. Since, hybrid NN have flexible time-delay criterion and capability; it can accomplish high rates of intrusion detection and low rate of false alarms. This method have high scalability and ability, to verify new patterns of attacks detection of BSM strings. K. Wu et al. [22] put forward NIDS method using CNNs. CNN automatically selects traffic features from raw dataset and apply cost function weight coefficient of particular class hinged on numbers, to deal with imbalanced dataset issue. The method not only minimize FAR, but also enhances accuracy of class with less numbers. To decrease computation cost raw traffic vector is converted to image format. In this utilizing original KDDCup-99 data-set to evaluate efficiency of suggested CNN model. Experimental results shows precision, FAR and computational cost of presented model have better performance compared to conventional standard algorithms. More improvements of detection accuracy of this work are possible, modifying CNN model structure for sake of achieving goal. In addition, detection time is also crucial to identify intrusion, it is essential to assure method is capable of meeting time requirements of IDS, enhancing accuracy of detection. J. Kim et al. [23] suggested DNN method for attack detection. The popular KDDCup 1999 data-set have been used for testing and training, to detect the intrusion. The testing data is generated via data pre-processing and extraction of samples, to meet aim of study. A DNN method, comprise of four hidden layers and hundred hidden units are used by suggested IDS of presented study as classification algorithm and ReLU function is used as activation function of hidden layers. In addition, they used adaptive moment (Adam) optimizer, a stochastic method of optimization for DNN learning. The results shows considerably high precision and detection rate of 99% and FAR 0.08% approximately.

T. A. Tang et al. [24] suggested deep learning technique for flow-based anomaly detection in an SDN environment. DNN method was constructed for IDS and trained using NSLKDD dataset. From experimentation, it has been discovered an optimal hyper-parameter for DNN and confirmed detection rates and false alarms. The method achieved efficiency with a precision of 75.75% approximately.

III. SYSTEM OVERVIEW

N. Shone et al. [10] presented new deep learning framework to allow NIDS operation in modern networks. This model is combination of deep and machine learning, able to perfectly inspect vast quantity of network traffic. Especially, combine the ability of stacked suggested Non-symmetric Deep Auto-Encoder (NDAE), which is the deep learning approach and the speed and accuracy of Random Forest (RF), which is machine learning method.

This research work deals with NDAE, is an auto-encoder presenting asymmetric diverse hidden layers. NDAE may use as hierarchical unsupervised feature extractor scales properly, to deal with excessive-dimensional inputs. It study's significant features by applying similar training method as compared to regular auto-encoder. Stacking NDAEs provide a layer-wise unsupervised feature learning, which allow suggested framework to study complicated relationships among distinct features.

Fig. 1 depicts suggested system architecture of Network Intrusion Detection. WSN trace dataset with 12 features are provided as input data to this framework. Training dataset contains data preprocessing which involve three steps: Data preprocessing, data normalization and transformation. This framework uses two NDAEs arranged in a stack, used, to select number of features. Then apply Random Forest Classifier for attack detection [25].

To prevent or reduce malicious behavior Intrusion Detection System (IDS) contains IDS functionality capable of taking immediate action. The IDS is implemented using Rule Status Monitoring Algorithm [26]. There are 8 rule actions for attack detected, system will take action using following list:

- ALERT - Generate alert using selected ALERT approach, and log packet.
- LOG - Log packet.
- PASS - Neglect packet.
- ACTIVATE - Alert and then set on other dynamic rule.
- DYNAMIC – Inactive till operated by an activate rule, then after appear as a log rule.
- DROP - Block and log packet
- REJECT - Block packet, log it, and if protocol is TCP then send TCP reset or if protocol is UDP then send an ICMP port unreachable message.
- SDROP – Do not log the packet but only Block the packet.

A. Architecture

The proposed system consists three steps as below.

1. Data Pre-processing: In this phase preprocessing, normalization and transformation is carried.

a) Preprocessing

Training and testing of neural network is carried out using only numeric values for classification. WSN trace dataset consist different data types. Hence, preprocessing stage is required to transform non-numeric values present in dataset to numeric values.

Two important work performed in pre-processing stage are:

1. Transforming non-numeric values of features, present in dataset to numeric values.
2. Convert attack types into its numeric categories.

b) Normalization

The features of WSN trace dataset have discrete or continuous values. The ranges value of feature is different and this makes them incomparable. Min-max normalization function is manipulated to plot all diverse values for each feature to [0, 1] range. Thus, In order to bring numeric values present in dataset in same range, normalization procedure is done.

c) Transformation

In this stage numeric normalized values of dataset are converted into its optimal form.

2. Feature Selection: Auto encoder is an unsupervised neural network-based feature extraction algorithm, applies back propagation in setting target value to be equal to input. The objective of auto encoder is to minimize reconstruction error between input and output. The proposed NDAE, is an auto encoder presenting asymmetric diverse hidden layers. The reason is, to decrease time and computational expenses with less effect on performance and accuracy. Stacking NDAEs provide layer-wise unsupervised feature learning, which allows proposed framework to study complicated relationships among distinct features.

3. Random Forest Classifier: Classification power of stacked auto encoder is poor in contrast to other discriminative frameworks like random forest, support vector machine, KNN, and many more. In this framework, RF classifier is trained utilizing encoded representations studied by stacked NDAEs, to distinguish network traffic into normal network data and attacks.

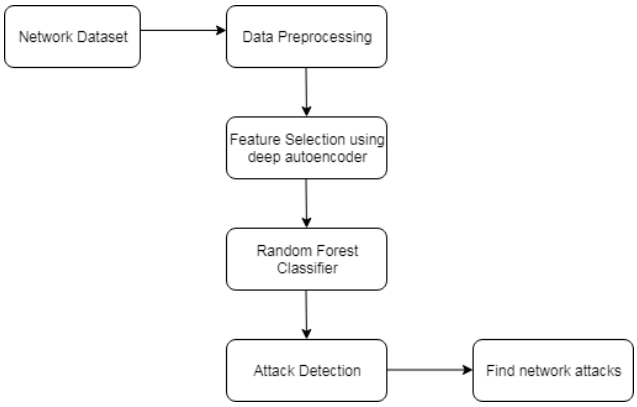

Fig. 1 Schematic representation of proposed system architecture

B. Mathematical Model

1. Preprocessing:

At this stage, data training source (F) is normalized, to implement processing by applying following steps:

$$F_{norm} = \left\{ \frac{F - \mu_F}{\sigma_F}, \sigma_F \neq 0 \; and \; F - \mu_F, \sigma_F = 0 \right\} \quad (1)$$

Where,

$$F = \{x_{m,n} | m = 1,2, \dots, i \; and \; n = 1,2,3, \dots, j\}$$

$$\mu_F = \{\mu_n | n = 1,2,3, \dots, j\}$$
$$\sigma_F = \{\sigma_n | n = 1,2,3, \dots, j\}$$

F has i samples with j column attributes; x_{mn} is the n^{th} column attribute in m^{th} sample, μT and σT are 1*j matrix which are the training data mean and standard deviation respectively for each of the j attributes. Test dataset (FS) which is used to determine detection accuracy is normalized by applying the same μ_F and σ_F as follows:

$$FS_{norm} = \frac{(FS - \mu_F)}{\sigma_F}, \sigma_F \neq 0 \; and \; FS - \mu_F, \sigma_F = 0 \quad (2)$$

2. Feature Selection:

NDAE is an auto-encoder presenting asymmetric diverse hidden layers. Suggested NDAE accepts input vector $l \in R^a$ and step-by-step plot it to latent representations $h_j \in R^a$ (Here a represents dimension of vector) applying a deterministic function shown below in Eq. 3:

$$h_j = \sigma(W_j . h_{j-1} + b_j); \; j = \overline{1, m,} \quad (3)$$

Here, $h_0 = l$, σ is an activation function (sigmoid function $\sigma(t) = 1/(1 + e^{-t})$)and m is number of hidden layers. In contrast to traditional auto-encoder and deep auto-encoder, suggested NDAE does not contain a decoder, output vector is estimated by applying similar formula as Eq. 4 as the latent representation.

$$k = \sigma(W_{m+1} . h_m + b_{m+1}) \quad (4)$$

The estimator of model $\theta = (W_j, b_j)$ is produced by minimizing square reconstruction error over n training samples $(l^{(j)}, k^{(j)})_{j=1}^{n}$, as given in Eq. 5.

$$E(\theta) = \sum_{j=1}^{n} (l^{(j)}, k^{(j)})^2 \quad (5)$$

C. Algorithms

1. Restricted Boltzmann Machine Algorithm

l_1 is a sample from the training distribution for the RBM.

ε is a learning rate for the stochastic gradient descent in Contrastive Divergence.

W is the RBM weight matrix, of dimension (number of hidden units, number of inputs).

d is the RBM offset vector for input units.

r is the RBM offset vector for hidden units.

Notation: $M\left(h_{2_j} = 1 \middle| l_2\right)$ is the vector with elements $M\left(h_{2_j} = 1 \middle| l_2\right)$

Step 1: ***for*** *all hidden units j* ***do***

Step2: compute $M\left(h_{1_j} = 1 \middle| l_1\right)$ *(for binomial units,*$sigm(r_j + \sum_i W_{ji}\, l_{1i})$*)*

Step 3: sample $h_{1_j} \varepsilon \{0,1\}$ *from* $M\left(h_{1_j} \middle| l_1\right)$

Step 4: ***end for***

Step 5: ***for*** *all visible units i* ***do***

Step 6: compute $N(l_{2_i} = 1 | h_1)$ *(for binomial units,*$sigm(d_i + \sum_j W_{ji}\, h_{1j})$*)*

Step 7: sample $l_{2_i} \varepsilon \{0,1\}$ *from* $N(l_{2_i} = 1 | h_1)$

Step 8: ***end for***

Step 9: ***for*** *all hidden units i* ***do***

Step 10: compute $M\left(h_{2_j} = 1 \middle| l_2\right)$ *(for binomial units,* $sigm(r_j + \sum_i W_{ji}\, l_{2i})$*)*

Step 11: ***end for***

Step12: $W \leftarrow W + \varepsilon \left(h_1 l_1' - M\left(h_{2_j} = 1 \middle| l_2\right) l_2'\right)$

Step 13: $d \leftarrow d + \varepsilon (l_1 - l_2)$

Step 14: $r \leftarrow r + \varepsilon (h_1 - M\left(h_{2_j} = 1 \middle| l_2\right))$

---

2. Deep Belief Network Algorithm

Train DBN in unsupervised manner follows greedy layer-wise procedure; each added layer is trained as an RBM (e.g., Contrastive Divergence).

$\hat{Z}$ is the input training distribution for the network.

ε is a learning rate for the RBM training.

q is the number of layers to train.

W^f is the weight matrix for level f, for f from 1 to q

d^f is the visible units offset vector for RBM at level f, for f from 1 to q

r^f is the hidden units offset vector for RBM at level f, for f from 1 to q

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*** *f = 1 to q* ***do***

Step 2: initialize $W^f = 0, d^f = 0, r^f = 0$

Step 3: ***while*** *not stopping criterion* ***do***

Step 4: sample $h^0 = l$ *from* $\hat{Z}$

Step 5: ***for*** *j=1 to f−1* ***do***

Step 6: ***if*** *mean_field_computation* ***then***

Step 7: assign $M(h_i^j = 1 | h^{j-1},$ *for all elements i of* h^j *)*

Step 8: ***else***

Step9: assign h_i^j *to* $M(h_i^j | h^{j-1},$ *for all elements i of* h^j*)*

Step 10: ***end if***

Step 11: ***end for***

Step 12: *RBMupdate* $(h^{f-1}, \varepsilon, W^f, d^f, r^f)$

{Thus providing $M(h^f | h^{f-1})$ *for future use}*

Step 13: ***end while***

Step 14: ***end for***

---

3. Random Forest Classifier

1. Assume number of variables in classifier be L and number of training cases be K.

2. The number of input variables be n, which regulate judgment of node of tree; n should be lesser than L.

3. Training set for this tree are chosen m times with substitution from all K available training cases. Apply rest of cases, to compute falsehood of tree, determine their classes.

4. For each node of tree, randomly choose n variables, on which judgment of that node is hinged. Estimate best split hinged on these n variables in training set.

5. Each tree is fully grown and not pruned (as may be done in building a normal tree classifier).

For prevision a new sample is pushed down tree. It is assigned label of training sample in terminal node it ends up in. This process is repeated for all trees in concert, and average poll of all trees is noted as random forest prevision.

IV. RESULT AND DISCUSSIONS

The evaluation were performed using windows 7 operating system, Intel i5 processor, 4 GB RAM, 200GB Hard disk, Eclipse Luna JDK 8 tool and Tomcat server. To perform evaluations WSN-trace dataset is used. WSN-trace is wireless dataset for researchers. WSN trace dataset contains total 19 attributes given below:

Table I WSN Trace 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	

WSN trace dataset is real-time wireless dataset gets information from router contains node details and packet information. The network traffic includes normal and different types of attacks like DoS, Probing, user-to-root (U2R), remote-to-local (R2L).

Throughout this work metrics defined below are used:

1) True Positive (TP) - Attack precisely distinguished as attack.

2) False Positive (FP) - Normal network data wrongly distinguished as attack.

3) True Negative (TN) - Normal network data precisely distinguished as normal data.

4) False Negative (FN) - Attack wrongly distinguished as normal data.

The following measures are used to evaluate performance of suggested solution:

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

The accuracy measures, fraction of total number of precise division.

Precision = TP / TP + FP

The precision measures, number of precise division condemn by number of wrong division.

Recall = TP / TP + FN

Recall measures, number of precise division condemn by number of missed entries.

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

F-measure, measures harmonic mean of precision and recall, serves as derived effectiveness measurement.

Performance of proposed system is combination of stacked NDAE (deep learning) and RF classifier (machine learning) using WSN trace dataset given in table below.

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

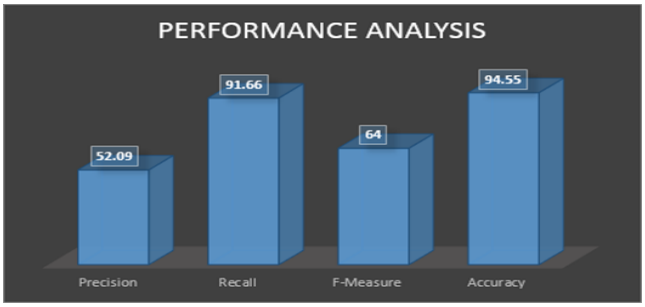

Fig. 2 Performance graph of stacked NDAEs + RF classifier using WSN Trace dataset

To find best performance different dataset are used to evaluate, which dataset could have best accuracy. Accuracy is chosen as essential interpretation standard. The results can be notice in Fig. 3, shows accuracy for different dataset on different network traffic. The fact have been determine from WSN trace dataset nearly have same accuracy, classifying normal traffic as well as detecting DoS attack and probing attack. The WSN trace dataset have better accuracy, classifying R2L attack and U2R attack as compared to NSL -KDD dataset.

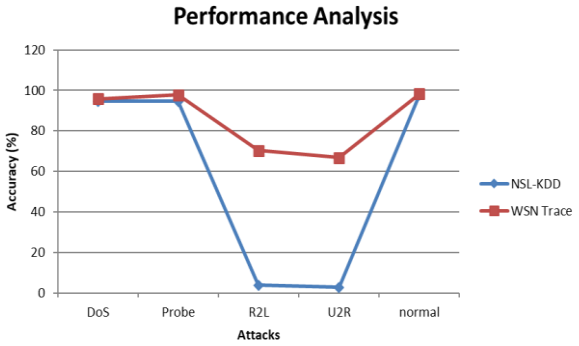

Fig. 3 Accuracy of different dataset used to recognize normal traffic and attacks.

V. CONCLUSION

The obtained result depicts given approach which is combination of stacked NDAE (deep learning) and RF classifier (machine learning) using WSN trace dataset provides high levels accuracy of 94.55 %, precision of 52.05% and recall of 91.66% along with moderate training time. The suggested NIDS system has enhance 4% accuracy. Since, still there is scope for more accuracy enhancement. In future scope, the work will be extended to design a real-time NIDS for real networks by manipulating deep advanced approach. Moreover, on-the-go feature learning on raw network trace headers rather than derived features utilizing raw headers can be immense effect research in this area.

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.
3. W. Stallings, "Cryptography and network security principles and practices," USA: Prentice Hall, 2006.
4. 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.
5. 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.
6. 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.
7. 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.

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. 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. 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.
12. 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.
13. 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
14. 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.
15. Y. Li, R. Ma, and R. Jiao, “A hybrid malicious code detection method based on deep learning,” methods, vol. 9, no. 5, 2015.
16. 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.
17. 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.
18. 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.
19. 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.
20. 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.
21. 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.
22. 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.
23. J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, “Method of intrusion detection using deep neural network,” in Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on, 2017, pp. 313–316.
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.
25. 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.
26. Claude Turner*, Rolston Jeremiah, Dwight Richards, Anthony Joseph, “A Rule Status Monitoring Algorithm for Rule-Based Intrusion Detection and Detection Systems”, Procedia Computer Science, ISSN: 1877-0509, Vol: 95, Page: 361-368, 2016

AUTHORS PROFILE

Vaishali V. Mandhare received the Ph. D in Information Technology from S. G. G. S. Institute of Engineering and Technology, affiliated to S. R. T.M University, Nanded and had received her B.E and M. Tech in Information Technology and Computer Science Engineering from Shivaji University, Kolhapur and Dr. B.A.T.U, Lonere in 2005 and 2009 respectively. She is working as Associate Professor in Department of Computer Engineering at Pravara Rural Engineering College, Loni, affiliated to Savitribai Phule Pune University. Her research interests include issues related to Wireless networking, Image processing Mobile Communication, Networking etc. She has published paper in national, international conference and journals. She is life member of ISTE and IAENG.

Deepika R. Pede, she have completed her B.E in Computer Engineering from Bharati Vidyapeeth College of Engineering for Women, Pune in 2013 affiliated to Savitribai Phule Pune University and pursuing her M.E. in Computer Engineering from Pravara Rural Engineering College , Loni , Tal: Rahata , District Ahmednagar affiliated to Savitribai Phule Pune University.

Pratap S. Vikhe received the M. Tech and Ph. D degree in Instrumentation engineering from SGGS Institute of Engineering and Technology Nanded affiliated to University of Shri Ramanand Teerth Marathwada University, Nanded in 2009 and 2018 respectively. He has completed his Bachelor’s degree in Instrumentation engineering from PDVVP, Ahmednagar affiliated to University of Pune in 2003. He works as an Associate Professor in the Department of Instrumentation and Control Engineering, at Pravara Rural Engineering College, Loni, Tal: Rahata, District Ahmednagar affiliated to University of Pune. His research interests include biomedical signal and image processing. He is author of few research papers published at national and international journals, conference proceedings. He is life member of ISTE, IAENG and IARA.