Proceedings of the Seventh International Conference on Communication and Electronics Systems (ICCES 2022)
IEEE Xplore Part Number: CFP22AWO-ART; ISBN: 978-1-6654-9634-6

Feature Extraction using LSTM Autoencoder in Network Intrusion Detection System

Gayatri Ketepalli
Department of Information Technology
Vignan's Foundation for
Science,Technology and Research
(Deemed to be University)
Guntur,India
gk_it@vignan.ac.in
0000-0002-0549-3620

Premamayudu Bulla
Department of Information Technology
Vignan's Foundation for
Science,Technology and Research
(Deemed to be University))
Guntur,India
drbpm_it@vignan.ac.in

2022 7th International Conference on Communication and Electronics Systems (ICCES) | 978-1-6654-9634-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCES54183.2022.9835788

Abstract— **IDS (intrusion detection systems) use analysis of network traffic patterns to detect incidents of hacking. It is essential to do feature extraction in order to minimize the computational cost associated with processing raw data in the IDS. Feature extraction decreases the number of features, which decreases the time it takes to train and increases accuracy. This research employs a simple LSTM autoencoder and a Random Forest to recognize intrusion attempts by IDSs. By activating and disabling various characteristics, the extent to which this feature extraction function can enhance accuracy is examined. To find out if detection algorithms are effective after feature extraction, the NSL-KDD dataset has been employed. Autoencoder hyperparameters contain the two activation functions. The loss and activation functions of the ReLU and the SoftMax have the greatest accuracy rating of any function. The use of a Long Short-Term Memory Autoencoder (LSTMAE) and a Random Forest (RF) for identifying the best features is a goal of this study. According to preliminary experimental data, classifiers that employ these variables have a prediction rate of 94.74 percent.**

Keywords—Intrusion Detection System, Machine Learning, Feature Extraction, Random Forest, Principal Component Analysis, Long Short Term Memory-Auto encoder, Deep Learning

I. INTRODUCTION

Threats are evolving continuously, and the seriousness of cyber assaults increases because the government, military, and commercial organizations have placed too much trust in the internet for day-to-day operations. Therefore, IDSs are of essential and fundamental importance to the security infrastructures that are in place to ensure the security of a system. Advance computer security. A lot of machine learning research has been done in the past several years. A wide range of models has been created and examined. Categorization algorithms, such as Decision tree, SVM, K-nearest, and feature selection, are used by various organizations. Essentially all IDS-based Using shallow learning on machine learning algorithms (single feed forwards networks.),

Feature engineering mechanisms that depend on a feature engineer's skills are being used to build features in the ML model. As a result of all the extensive data inputs, it is difficult to process. For this reason, models that include deep root concepts are more effective in encouraging learning, such as PCA, Random Forest, and LSTM autoencoder LSTMAE).

In the last several years, LSTM (long short-term memory) models have been growing in popularity. Additionally, there is the number of retrieved characteristics from raw network data. IDS must determine, even for a small network, is very big. This also means that most of the retrieved data are irrelevant and noisy, impairs the classifier's effectiveness. PCA (Principal component analysis) and Mutual information are related because they use dimensionality reduction to change the number of variables that a data set has selecting useful data is critical.

The process of reducing the number of available attributes is known as feature extraction. Research by Srivastava et al, 2015. Jiang and colleagues, 2020 In addition to speech recognition developed by Jaeger & Haas in 2004, (T. Kim et al., 2017). The learning process may improve accuracy even after PCA and LDA have been performed on the data. PCA and LDA are two examples of linear discriminant analysis.

MDS, ISOMAP, LLE and Laplacian Eigenmaps are some of the most utilized mathematical methodologies in feature extraction research (Papamartzivanos et al., 2019). (LE). Some other problems have recommended reducing the data down to a lower dimension to extract specific features all at once. However, this method's effectiveness degrades drastically when extracting a significant amount of data. Feature extraction is a machine learning method that uses an autoencoder (Shone et al., 2018).

An artificial neural network, called AE, utilizes the artificial neural network notion of depreciation of the reconstruction layer to decrease the dimensionality of the hidden layer. The Autoencoder's capability to extract important characteristics was greatly demonstrated by the Sparse Autoencoder and its related variants, such as the

Variational Autoencoder, Denoising Autoencoder, and Relational Autoencoder.

We explore automated feature extraction as a means of analyzing intrusion detection systems (IDS). We use Autoencoder to extract features from the data automatically. The research is focused on finding out how the hyperparameters of the activation and loss autoencoders may be used to boost attack detection accuracy. Furthermore, to evaluate the correctness of a score, RF is utilized (Kabir et al., 2018).

II. Related Work

Aldwairi et al. (2018), Various models have been previously developed to assist researchers in overcoming the anomalies in intrusion detection system's limitations. The following methods will be discussed in this section for analyzing IDS logs. Feature extraction, also known as feature selection, is often used as a first network traffic processing step for intrusion detection.

Feature reduction, which is performed using the decision tree classifier, is calculated based on the overall value of each class. Twenty-two criteria were used to arrive at the most accurate possible degree of detail. The feature extraction study focuses on reducing the dimensions of information while preserving its content.

As shown by Xin et al. (2018), a PCA-based feature reduction and classification model was developed based on PCA and classification anomalies (PCA). This procedure allows for the recovery of 22 attributes from a total of 41. In addition, the time it takes to compute the characteristics is reduced with this method. Although these deficiencies exist, the overall component selection still does not meet expectations since it analyzes just a subset of attributes.

For PCA and Linear Discriminant Analysis reduction, do an investigation (LDA). Decreasing the dataset size is used to categorize after using back-propagation. PCA was better able to classify smaller datasets than LDA.

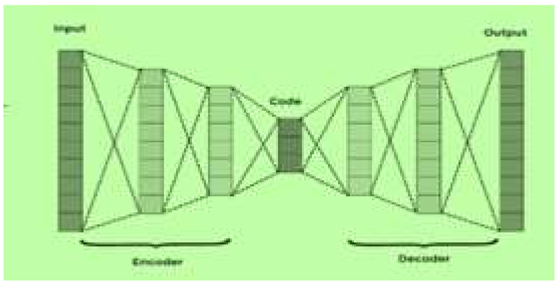
Aggarwal and Kumar (2015) state that in the case of network intrusion detection, Kernel Principal Component Analysis (KPCA) and Random Forest (RF) should be combined. Classifier models are used to identify assaults that have taken place in a multi-layered manner. They created an intrusion detection model in 2015 using principal component analysis (PCA) and Random Forest (RF). Reduced training and testing were achieved using RF-optimized RBF kernel parameters and cross-validation. In preventing non-significant U2R and R2L assaults, it is more precise.

D-FES uses layered feature extraction and weighted feature selection to detect Wi-Fi impersonation attempts. It is possible that future research on hyper parameter auto encoders and their impact on accuracy could still be advantageous.

III. Methods and Design

A. *Auto encoder*

The Auto encoder is a type of neural network that can learn a compressed version of raw input. An auto encoder is constructed from two sub-models: an encoder and a decoder. While compressing the input, the encoder does the work; when reconstructing the input, the decoder uses the compressed input. The encoder model is retained, but the decoder is removed once the model has been trained. When the encoder is used to prepare raw data for training another machine learning model, it is referred to as data preparation (Zhang et al., 2019). An auto encoder is a sort of neural network used to learn a condensed representation of the input. Auto encoders are neural networks that have been trained to try to recreate their input.



B. *LSTM Auto encoder*

LSTM Encoder-Decoder Auto encoder (J. Kim et al., 2016). Once the encoder is installed, it may encode or compress sequences of data for use in data visualizations or in training a supervised learning model. The following algorithm uses an encoder-decoder LSTM to convert input sequences into an output sequence that can be replicated for any dataset of sequences. The model's ability to accurately predict the input sequence is what defines its performance. The encoder model may be left as long as the model can perform the task of reproducing the sequence. Following that, this approach may be used to generate fixed-length vectors from input sequences. the sequence input to another supervised learning model may be used to compress vectors that may be further employed for many reasons, some of which includes creating a compressed representation of the sequence

Fig 1(a) and 1(b) depict the RNN-LSTM auto encoder (Le et al., 2017), which consists of recurrent hidden layers and LSTM memory blocks in the hidden layer, respectively (b). This image depicts the overall LSTM architectural structure, as shown in figure 2.

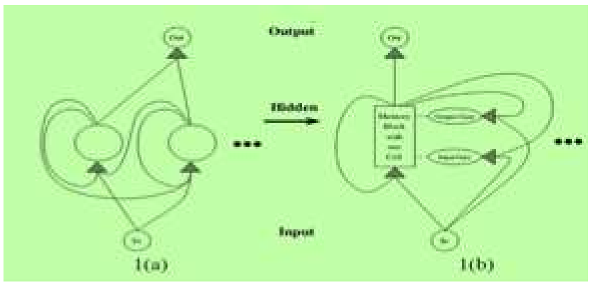


Fig. 1. (a) RNN with one fully recurrent hidden layer, 1(b) LSTM with memory blocks in the hidden layer

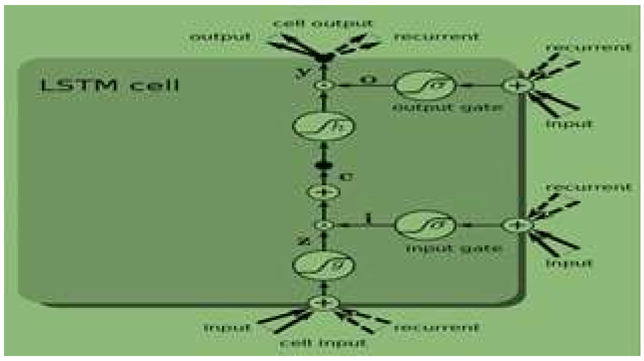


Fig. 2. LSTM architecture

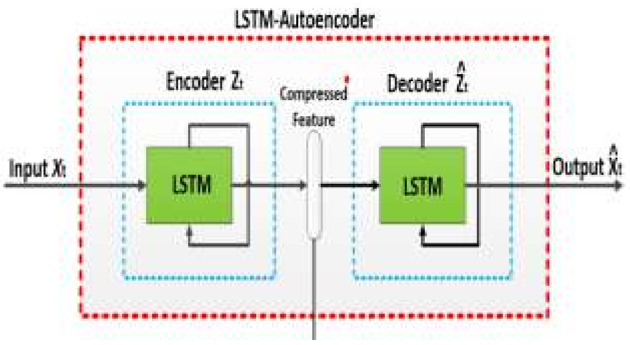


Fig.3. LSTM Autoencoder

The mathematical model of LSTM Autoencoder is derived as follows
Equation (1) and (2) represents the LSTM input

$$b_z(t) = W_z x(t) + S_z y(t-1) \qquad (1)$$

$$z(t) = g\big(b_z(t)\big) \qquad (2)$$

Equation (3) and (4) derives the data for the LSTM input gate

$$b_{in}(t) = W_{in} x(t) + S_{in} y(t-1) \qquad (3)$$

$$i(t) = \sigma\big(b_{in}(t)\big) \qquad (4)$$

Equation (5) derives constant error carousel of LSTM

$$c(t) = z(t) \odot i(t) + c(t-1) \qquad (5)$$

Equation (6), (7), and (8) derives constant error carousel of LSTM

$$b_{out}(t) = W_{out} x(t) + S_{out} y(t-1) \qquad (6)$$
$$o(t) = \sigma\big(b_{out}(t)\big) \qquad (7)$$
$$y(t) = h\big(c(t)\big) \odot o(t) \qquad (8)$$

LSTM Forward Pass: The cell state c is updated based on its current state and three inputs: b_z , b_{in} , b_{out}

$$b_z(t) = W_z x(t) + S\big(y(t-1)\big),\ z(t) = g\big(b_z(t)\big) \qquad (9)$$
$$b_{in}(t) = W_{in} x(t) + S_{in} y(t-1),\ i(t) = \sigma\big(b_{in}(t)\big) \qquad (10)$$
$$b_{out}(t) = W_{out} x(t) + S_{out} y(t-1)\ ,\ o(t) = \sigma\big(b_{out}(t)\big)$$
(11)
$$y(t) = h\big(c(t)\big) \odot o(t) \qquad (12)$$

LSTM Activation function

σ is the activation function of both gates:

$$\sigma(x) = \frac{1}{1+exp(-x)} \qquad (13)$$

g is the activation function of the input:

$$g(x) = \frac{2}{1+\exp(-x)} - 1 \qquad (14)$$

h is the activation function of the output:

$$h(x) = \frac{4}{1+\exp(-x)} - 2 \qquad (15)$$

LSTM Backward pass:

$$\delta_y(t) = \frac{\partial E_t}{\partial y(t)} + S_z^T \delta_z(t+1) + S_i^T \delta_i(t+1) + S_o^T \delta_o(t+1) \ (16)$$

The overall amount of data we're dealing with has grown significantly over the last several decades, including images, videos, and genetic information. Duplicated and unneeded information in high-dimensional data may lead to poor performance and increased computing costs. On the other hand, reducing the data's dimensionality with the extraction of features may lead to discoveries like pattern recognition, computer vision, and other helpful activities.

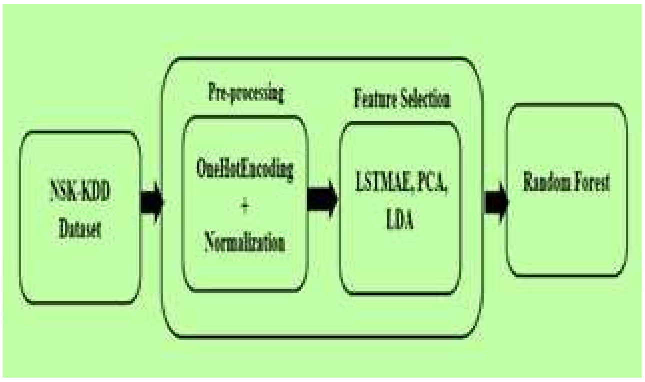


Fig. 4. Proposed system architecture

In the pre-processing step of the Deep Learning process for attack detection, the model suggested in the picture displayed in Figure 3 is used. The Pre-processing feature selection of 42 features sets off the whole feature extraction function. One Hot Encoding is used in this research. This is our method of choice. It produces a better outcome than the usage of ordinal codes. We take care of everything, even non-numerical characteristics. Numerical data is collected in the dataset, and non-numerical data is obtained from the dataset (especially on features 2, 3, and 4 of protocol type, service, and flag). The variable in binary form takes the value of each level on the map. For the following step, a set of features will be utilized with 120 features and only one feature label.

C. Feature selection

The performance of machine learning algorithms can be significantly impacted when working in a real-time situation with multiple dimensions in the feature space. The performance of machine learning algorithms can be significantly impacted when working in a real-time situation with multiple dimensions in the feature space. When working with large datasets, numerous approaches for dimensionality reduction and feature selection are employed. As a result, the machine learning algorithm is able to learn more quickly, while also simplifying the model and making it easier to understand [18]. Long short-term memory auto encoder (LSTMAE) and linear discriminant analysis are three approaches we utilise to reduce dataset dimensionality in our studies (LDA). There are two linear transformation methods: LDA and PCA: PCA ignores class labels whereas LDA uses supervision.

For features to not overwhelm one other, all features must be scaled. Feature scaling is the term used to describe this procedure. Various options, including Standardization, Scaling, and Normalization [19], are utilized to accomplish feature scaling [20]. Z Score Normalization was utilized in this study. SVM, logistic regression, and neural networks are all examples of machine learning methods that use it [21]. First, the mean and standard deviation of each attribute are calculated [24].

Feature selection by using the Autoencoder model is the next step. The preliminary research utilized an autoencoder with three layers, one of which included a hidden layer. The One Hot Coding Model is the framework we utilize for pre-processing. The input neurons of the Autoencoder have 120 units in them. In the hidden layer, we repeated X 100, 80, 60, 40, 22, 20, 15, 10, and 8 times to get the highest accuracy value (see Figure3). When the data have reduced dimensions of the characteristics, they have been extracted. Finally, training and testing may be used to process the output.

D. Datasets

Research involving network assaults frequently makes use of the NSL-KDD dataset. This dataset may not apply to modern networks, as noted by Mc Huge [26], who argues that it is out-of-date and faulty. NIDS studies, on the other hand, typically use it as a starting point. 24 attack types and 42 attribute values are included in the NSL-KDD dataset. There are four basic categories of assaults: DOS, R2L, U2R, and probing, which are used to define them. Data correction was applied to 311,029 test records utilized for testing in the NSL-KDD test set, which totaled 494 021.

There has been an improvement in this data set compared to KDD cup99 [15]. An effective intrusion detection system is the goal of several studies on the NSL-KDD dataset which have been conducted using a variety of approaches and tools. Machine learning techniques have been used extensively to analyze the NSL-KDD data set by the WEKA tool [16]. The NSL-KDD data collection [17] is used by the K-means clustering technique to train and test several existing and new threats. Using an Artificial Neural Network termed the Self Organization Map (SOM), the NSL-KDD data set was compared to the KDD99 cup data set. There are a variety of data mining-based machine learning techniques employed, such as support vector machines (SVMs), decision trees (DTs), K-nearest neighbors (KNs), K-means (KMs), and fuzzy C-means clustering algorithms (FCMs).

TABLE I. ATTACKS IN TESTING DATASET

Attacks in Dataset	Attack Type (37)
DOS	Back,Land,Neptune,Pod,Smurf, Teardrop,Mailbomb,Processtable,Udpstorm,Apache2,Worm
Probe	Satan,IPsweep,Nmap,Portsweep,Mscan,Saint
R2L	Guess_password,Ftp_write,Imap,Phf,Multihop,Warezmaster,Xlock,Xsnoop,Snmpguess,Snmpgetattack,Httptunnel,Sendmail, Named
U2R	Buffer_overflow,Loadmodule,Rootkit,Perl ,Sqlattack,Xterm,Ps

E. Classifier

For the categorization of 5 classes, we utilized the Random Forest method. Based on numerous experiments, researchers found that RF-based IDS categorization is both effective and suitable. In the beginning, we utilize the training-testing split on NSL-KDD [22, 24] training data to observe activation and loss in the autoencoder process. The ratio of the train to test is often utilized for the train-test split. Activation and optimization functions give the best accuracy values here. Epoch, batch, and activation functions provided the highest accuracy in cross-validation (cv = 10). Cross-validation is to estimate the model values better. In machine learning, the value of k = 10 is a frequent number. In the following step, NSL-KDD data is used to conduct a five-class classification test, where NSL-KDD data is tested using testing data (corrected data).

Additionally, the model is put through testing and training using 5 class classification from NSL KDD data. In Tables 1

TABLE II. THE PERFORMANCES AE-RF IDS FOR NSL KDD DATASET

Methods	Activation Function	Loss Function	Number of features	Accuracy (%)
RF (All features)	--	--	120	99.934
RF-LSTMAE	Linear	MSE	22	99.867
RF-LSTMAE	Linear	Entropy	22	99.969
RF-LSTMAE	Sigmoid	MSE	22	99.987
RF-LSTMAE	Sigmoid	Entropy	22	99.956
RF-LSTMAE	ReLU	MSE	22	99.949
RF-LSTMAE	ReLU	Entropy	22	99.948
RF-LSTMAE	SoftMax	MSE	22	99.891
RF-LSTMAE	SoftMax	Entropy	22	99.892
RF-LSTMAE	Softplus	MSE	22	99.941
RF-LSTMAE	Softplus	Entropy	22	99.943
RF-LSTMAE	Tanh	MSE	22	99.942
RF-LSTMAE	Tanh	Entropy	22	99.925

and 2, the training and testing datasets are shown. Finally

, using kernel RBF, we evaluate our RF classifier model with a linear and a polynomial kernel. The remaining

parameters, such as gamma, degree, and others, in scikit, learn to utilize the default values. We will fine-tune additional settings in the following phase.

IV. RESULTS AND DISCUSSION

Functional loss (MSE) and cross-entropy are both described in this article. To test for the best possible IDS activation function, a subset of activation functions, such as linear, sigmoid, ReLU, SoftMax, Soft Plus, and tanh, are applied to the hidden layer to find which activation provides the best fit. The PCA is our starting point. To evaluate the classification accuracy and processing time for feature extraction, feature extraction output was examined using RF. In addition, we studied the applicability of machine learning classification in the experiment without the use of automated feature extraction. It was important to us to ascertain if critical information could be retrieved from the original characteristics. These NSL-KDD dataset findings are presented in Table 2. More than a dozen investigations have shown that there are as many as 20 characteristics in the NSL-KDD. Feature extraction's automatic functionality has the added benefit of preserving critical data about features before they are reduced. Function activation and function loss perform better than RF models without any feature extraction.

Table 3 shows that the ReLU activation and entropy loss feature extraction model is the best. Without automatic feature extraction, the machine learning model has an accuracy rate of more than 99.947 percent (increased 0.024 percent). Other activation functions, such as soft plus, loss entropy activation, and activation linear, are more effective in delivering results (increased 0.016 percent).It takes the largest amount of time to train using softmax activation models. The features that show the cross-entropy provide a more accurate loss function outcome for the automated extraction process. It is easier to obtain a local optimum using the loss cross-entropy function than it is using the MSE. We utilized tenfold cross-validation on the dataset to ensure that the model was accurate. The NSL-KDD dataset yielded a 99.464 +/- 0.019-point structured model. Round-off error for NSL-KDD is 99.447 +/- 0.048 on average.

TABLE III. NSL-KDD 5 CLASS PERFORMANCE

Attack Class	No. Training	No. Testing	Accuracy (%)			Precision (%)			F-Score (%)		
			RFPCA	RF	RFLSTMAE	RFPCA	RF	RFLSTMAE	RFPCA	RF	RFLSTMAE
DoS	45928	5740	56.253	99.128	97.81	64.124	97.133	99.454	59.812	98.16	98.659
Probe	11655	1107	37.081	91.59	88.075	13.595	60.219	78.119	20.884	72.69	82.798
R2L	994	2200	0.055	11.64	12.879	103	92.765	97.489	0.103	20.685	22.583
U2R	53	38	0	24.335	8.128	0	69.242	51	0	37	13.767
normal	67342	9710	83.626	91.847	97.457	76.434	81.518	82.588	81.314	85.432	88.835
Total	125972	18795	177.015	318.54	304.349	257.153	400.877	408.65	162.113	313.967	306.642

V. CONCLUSION

This first study utilizes the automated feature extraction method to research IDS attributes. When it comes to using Automatic feature extraction, a technique that combines compression with dimensionality reduction has shown excellent results. The linear activation accuracy value with a cross-entropy loss function is demonstrated by the high value of ReLU activation accuracy. When optimizing parameters, we discovered that the best hyperparameter model for the Auto encoder uses a ReLU activation function with cross-entropy as the loss function. This model can be processed with an incredibly high level of accuracy at a fast pace compared to other functions. The study's shortcomings stem from the smaller dataset that was utilized. In the current context, the dataset is outdated due to changes in network technology and attack types. New and more comprehensive datasets will be analyzed in the future for future researchers. To facilitate a new hybrid clustering model that employs an intricate autoencoder process on the IDS system, we're planning to develop a new hybrid clustering model.

REFERENCES

[1] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019a). Features dimensionality reduction approaches for machine learning based network intrusion detection. Electronics (Switzerland), 8(3), 322. https://doi.org/10.3390/electronics8030322

[2] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., & Abuzneid, A. (2019b). Features Dimensionality Reduction Approaches for Machine Learning Based Network Intrusion Detection. Electronics, 8(3), 322. https://doi.org/10.3390/electronics8030322

[3] Aggarwal, P., & Kumar, S. (2015). Analysis of KDD Dataset Attributes - Class wise For Intrusion Detection. Procedia - Procedia Computer Science, 57, 842–851. https://doi.org/10.1016/j.procs.2015.07.490

[4] Aldwairi, T., Perera, D., & Novotny, M. A. (2018). An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection. Computer Networks. https://doi.org/10.1016/j.comnet.2018.07.025

[5] Aminanto, M. E., Choi, R., Tanuwidjaja, H. C., Yoo, P. D., & Kim, K. (2017). Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. IEEE Transactions on Information Forensics and Security, 13(3), 621–636. https://doi.org/10.1109/TIFS.2017.2762828

[6] Jaeger, H., & Haas, H. (2004). Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. Science. https://doi.org/10.1126/science.1091277

[7] Jiang, F., Fu, Y., Gupta, B. B., Liang, Y., Rho, S., Lou, F., Meng, F., & Tian, Z. (2020). Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security. IEEE Transactions on Sustainable Computing. https://doi.org/10.1109/TSUSC.2018.2793284

[8] Kabir, E., Hu, J., Wang, H., & Zhuo, G. (2018). A novel statistical technique for intrusion detection systems. Future Generation Computer Systems, 79. https://doi.org/10.1016/j.future.2017.01.029

[9] Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016). Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection. 2016 International Conference on Platform Technology and Service, PlatCon 2016 - Proceedings. https://doi.org/10.1109/PlatCon.2016.7456805

[10] Kim, T., Cha, M., Kim, H., Lee, J. K., & Kim, J. (2017). Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. http://arxiv.org/abs/1703.05192

[11] Kumar, S., & Yadav, A. (2015). Increasing performance of intrusion detection system using neural network. Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014, 978, 546–550. https://doi.org/10.1109/ICACCCT.2014.7019145

[12] Le, T. T. H., Kim, J., & Kim, H. (2017). An Effective Intrusion Detection Classifier Using Long Short-Term Memory with Gradient Descent Optimization. 2017 International Conference on Platform Technology and Service, PlatCon 2017 - Proceedings. https://doi.org/10.1109/PlatCon.2017.7883684

[13] Papamartzivanos, D., Gomez Marmol, F., & Kambourakis, G. (2019). Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2893871

[14] Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A Deep Learning Approach to Network Intrusion Detection. IEEE Transactions on Emerging Topics in Computational Intelligence. https://doi.org/10.1109/TETCI.2017.2772792

[15] Srivastava, N., Mansimov, E., & Salakhutdinov, R. (2015). Unsupervised Learning of Video Representations using LSTMs. 32nd International Conference on Machine Learning, ICML 2015, 1, 843–852. http://arxiv.org/abs/1502.04681

[16] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., & Wang, C. (2018). Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access, 6(c), 35365–35381. https://doi.org/10.1109/ACCESS.2018.2836950

[17] Zhang, Y., Chen, X., Jin, L., Wang, X., & Guo, D. (2019). Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. IEEE Access. https://doi.org/10.1109/ACCESS.2019.2905041

[18] Srikanth Yadav Moraboena, Gayatri Ketepalli, Padmaja Ragam(2020) A Deep Learning Approach to Network Intrusion Detection Using Deep Autoencoder Revue d'Intelligence Artificielle ,Vol. 34, No. 4, August 2020,pp. 457-463, ISSN: 1958-5748

[19] Srikanth Yadav.M, K. Sushma, Gayatri.K, Enhanced Network Intrusion Detection Using LSTM RNN ,International Journal of Advanced Science and Technology, Vol. 29 No. 5, May 2020, 7210,7220, ISSN: 2005-4238.

[20] Gayatri Ketepalli, Premamayudu Bulla (2020) Review on Generative deep models and datasets for Intrusion detection Systems, Revue d'Intelligence Artificielle, Vol,34, No.2, PP-215-226, ISSN: 1958-5748

[21] K. Gayatri, B. Premamayudu, and M. Srikanth Yadav (2020) A Two-Level Hybrid Intrusion Detection Learning Method , Machine Intelligence and Soft Computing, Advances in Intelligent Systems and Computing 1280, https://doi.org/10.1007/978-981-15-9516-5_21.

[22] Srikanth Yadav M and Kalpana R, “A Survey on Network Intrusion Detection Using Deep Generative Networks for Cyber-Physical Systems,” in Artificial Intelligence Paradigms for Smart Cyber-Physical Systems.Hershey PA, USA: IGI Global, 2020, ch. 07, pp. 137–159. DOI: 10.4018/978-1-7998-5101-1.ch007

[23] M. Srikanth Yadav. and R. Kalpana., "Data Preprocessing for Intrusion Detection System Using Encoding and Normalization Approaches," 2019 11th International Conference on Advanced Computing (ICoAC), Chennai, India, 2019, pp. 265-269, doi: 10.1109/ICoAC48765.2019.246851.

[24] Srikanth Yadav M., Kalpana R. (2022) Effective Dimensionality Reduction Techniques for Network Intrusion Detection System Based on Deep Learning. In: Jacob I.J., Kolandapalayam Shanmugam S., Bestak R. (eds) Data Intelligence and Cognitive Informatics. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-6460-1_39