Quest Journals
Journal of Software Engineering and Simulation
Volume 10 ~ Issue 6 (2024) pp: 54-61
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Optimization Accuracy for Network Attack Detection based on Deep Learning Techniques

Rahul Sharma[1] and Dr. Amit Asthana[2]
Department of Computer Science and Engineering, Shree Guru Gobind Singh Tricentenary University Gurugram,India
rahul941999@gmail.com, amitasthana_feat@sgtuniversity.org

Abstract- *With the fast-growing network scale, network intrusions are becoming more and more frequent, volatile and advanced. The problem of how to capture the intrusions in such a large-scale network is critical and challenging. Intrusion detection has become one of the challenging fields of research in every networked environment. The IDS models should be able to handle the huge volume and velocity associated with network communications. Current researches in areas of intrusion detection have tended towards network-based systems and how to improve on their intrusion detection. However, both host-based and network-based systems should be involved to effectively detect attacks from insider as well as outsider users. Deep learning (DL) based solution framework is developed consisting of approach. The approach is Long-Short Term Memory (LSTM) and gate recurrent unit (GRU) with seven optimizer functions The model is evaluated on CIS IDS dataset and classified multi attack classification. The model has outperformed with adamax optimizer in terms of accuracy and loss. The results of LSTM-GRU are compared with existing shallow machine and deep learning models in terms of accuracy. The proposed method provides an accuracy of 99.46%. Clearly, the proposed method is a 1.07% improvement accuracy compared to previous technique.*

Keywords- Long-Short Term Memory (LSTM), Intrusion Detection, Deep Learning

R*eceived 14 June, 2024;* ***R****evised 25 June, 2024;* ***A****ccepted 28 June, 2024 © The author(s) 2024.*
P*ublished with open access at* ***www.questjournals.org***

I. INTRODUCTION

With the advent of digital technology, the size of the data being generated every second has been crossing the boundary of gigabytes and even into terabytes. Companies from different domains are gaining profit by managing their resources and transactions over the network. Thus, big data security remains a key challenge for all the solutions because data value is not worth it if we compromise on data security and privacy. One of the most important security challenges over network traffic data is to detect and prevent network intrusions with high accuracy and less prediction time. These intrusions affect the confidentiality, integrity, and availability of big data resources and offered services. Some providers of big data services use the firewall to find a solution for the above issues [1, 2].

A firewall is treated as the first line of defence, can only sniff the packets at the border of a network (outsider intruders), insider intruders cannot be detected. Several intruders are too complex to detect using a traditional firewall. Thus, the traditional firewall is not an efficient solution to block all intrusions. To define an efficient solution for such problems, a high speed intrusion detection system should be capable to work in big data environment and process huge data of network traffic at the same time, since it acts as an additional preventive layer of security [3, 4]. The main reasons for Intrusion Detection System (IDS) popularity is that it can provide security on the host as well as on the entire network such as network monitoring, maintaining the consistency, availability, reliability, and the integrity of the data and services hosted on the host or network. Intrusion detection systems are either the hardware or software or both that provide security to the data and the services on the host or the network according to the defined policies. If any violation of the security policies is caught, the system administrator is alerted [5].

Therefore, the domain of intrusion detection systems based on data mining is the best encapsulation of the technical framework to overcome the range of intrusion of network traffic. The entire of these attacks have heavy consequences in an environment. For that reason, it is better to classify these attacks using the most common classification techniques at the beginning step so that the attacker could be blocked. This is can be

achieved using an efficient intrusion detection system, which can identify the intrusions earlier than the attack can take place and can give a notification that it is possible to have an attack [6]. The efficient machine learning algorithms are taken into consideration efficiently in predicament domains with altering parameters and continuity. The approaches employed in general for malicious entity identification are dominant, such as Support vector machine, Bayesian belief network, decision tree, and artificial neural network. Data mining becomes a key factor in the intrusion detection system. Due to the massive volume of existing and newly appearing network data, advanced processing is required [7, 8]. Machine Learning (ML) methods are predominantly employed to classify the intrusions from these big network data. However, these algorithms are not compatible with the modern network data with new attack classes and result in inadequate training, high false-positive rates, and high complexity. Deep Learning (DL) is an advanced class of ML algorithms that overcomes the problems of inadequate training and reduced computational capabilities. Accuracy and prediction time of the network traffic is the main importance in the networks for preventing an intruder from attacking the network and avoiding network crashes and failures.

II. PROPOSED METHODOLOGY

The LSTM architecture is formed by connecting key units called memory blocks. Memory block contains memory cells with self- feedback connections to store the data and a special type of logic gate to manage the flow of data. Input flow is controlled by organized memory cell with input data and activation function.

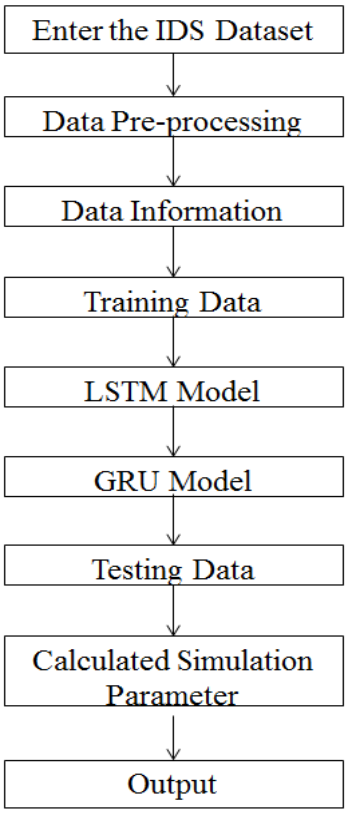

Figure 1: Flow Chart of Proposed Methodology

Similarly, the output gate is used to control the data flow by controlling cell along with the activation function (figure 1). Data is controlled with the help of gates at both entry and exit, i.e., while entering to the block, as well as at output point and whenever is about forget by the solution model. Function of theses gates are explained in the next sub sections.

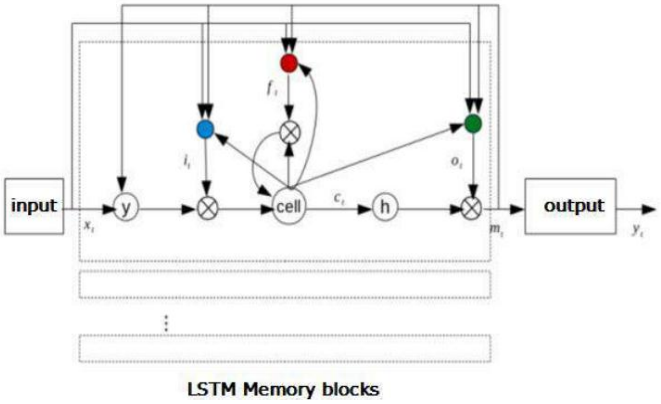

Figure 2: LSTM Node

Forget Gate: This gate determines whether data should be saved or discarded in the cell state. Combine the current time input (x) with the past time output ($ht-1$) to decide whether to save or reject the data. If the output is binary 0, the data should be discarded. If the output is 1, the data should be stored as:

$$f_t = \sigma(W_f . x_t + W_f . h_{t-1} + b_f)$$

The possible values of f_t are always in the range 0 to 1. Multiplication between f_t and h_{t-1} is a bitwise operation. Based on the results, some of the values of h_{t-1} are considered and some elements are removed.

Input Gate: The general principle of input gate operation is defined based on the results of two stages of operation. In the first phase, the values to be updated are determined and new cell state values are generated by:

$$i_t = \sigma(W_i . x_t + W_i . h_{t-1})$$

$$g_t = tanh(W_g.x_t + Wg.h_{t-1} + b_g)$$

The new value of the cell state is determined by a combination of oblivion and oblivion input gate. Clearing old data and saving the resulting value is handled by the input gate. Writing critical data to the cell state is done by input gates:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes g_t$$

Which part of the current cell state (C_t) is considered for reading and redirection Decide what should be done. Exit is controlled through an exit gate.
Oblivion gates are used to reduce internal data before connecting to the next node energetically. Adjust or rearrange cell memory. In order to know the exact timing of results or outputs, the newly introduced LSTM architecture forms a peephole network from the inner cell to various gates of similar cells. To store data for a long period of time, we need to store and retrieve data through gates that are multiplicative in nature. A sigmoid function (0-1) and this type of gate help the LSTM to save or clear the cell state data.

Algorithm:
CIS IDS dataset using Enhanced LSTM
Input: CIS IDS data set with class labels
Output: Classification of CIS IDS whether IDS implies Fwd Pkts, Bwd Pkts, Pkt Len Max, Pkt Len Min and Pkt Len Mean

Step 1: Pre-processed CIS IDS Dataset taken in the form of .csv file, data set is loaded

Step 2: Tagging the CIS IDS Dataset

Step 3: Tagged CIS IDS Dataset converted into vectors (word2vector conversion)

Step 4: Apply Evolutionary algorithm on the vectors to select the best feature set

Step 5: Enhanced-LSTM performs training only on the best features set selected by Evolutionary Algorithm and obtains a Model

Step 6: Testing data set is supplied to the Model obtained by Enhanced LSTM

Step 7: Evaluate the performance of this model based on some parameters

III. SIMULATION RESULT

3.1 Simulation Parameter

Accuracy: Accuracy is the proportion of number of precisely detected anomaly records and normal records to all records.

$$\text{Accuracy} = TP+TN/\ (TP+TN+FP+F\text{N})$$

4.2 Simulation Result

Steps as follows proposed work

1. Import important libraries
2. Download and access CIS IDS dataset
3. Show data in pandas data frame

[ ] df_dataset

	Unnamed: 0	Dst Port	Protocol	Flow Duration	Tot Fwd Pkts	Tot Bwd Pkts	TotLen Fwd Pkts	TotLen Bwd Pkts	Fwd Pkt Len Max	Fwd Pkt Len Min	...	Fwd Seg Size Min	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	146275	3389	6	5206015.0	9	11	1213	1948	661	0	...	20	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1
1	512109	443	6	60991608.0	14	14	1211	3386	875	0	...	20	116504.6667	117401.7629	358150	68519	10002387.33	1.665183e+04	10009249.0	9968397	1
2	807703	53	17	27292.0	1	1	45	163	45	45	...	8	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1
3	491912	443	6	70343424.0	17	20	1118	5653	258	0	...	20	527883.5000	388311.5264	800847	254520	34570235.00	3.406089e+07	58654821.0	10485549	1
4	858679	53	17	137267.0	2	2	94	214	47	47	...	8	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1048337	543790	53	17	224.0	1	1	43	59	43	43	...	8	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1
1048338	745477	53	17	1392.0	1	1	35	87	35	35	...	8	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1
1048339	871781	53	17	219.0	1	1	40	56	40	40	...	8	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1
1048340	10521	80	6	11976.0	3	4	314	935	314	0	...	20	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	4
1048341	821802	80	6	96.0	2	0	0	0	0	0	...	20	0.0000	0.0000	0	0	0.00	0.000000e+00	0.0	0	1

1048342 rows × 79 columns

1. Data information

```
df_dataset.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048342 entries, 0 to 1048341
Data columns (total 79 columns):
 #   Column              Non-Null Count    Dtype
---  ------              --------------    -----
 0   Unnamed: 0          1048342 non-null  int64
 1   Dst Port            1048342 non-null  int64
 2   Protocol            1048342 non-null  int64
 3   Flow Duration       1048342 non-null  float64
 4   Tot Fwd Pkts        1048342 non-null  int64
 5   Tot Bwd Pkts        1048342 non-null  int64
 6   TotLen Fwd Pkts     1048342 non-null  int64
 7   TotLen Bwd Pkts     1048342 non-null  int64
 8   Fwd Pkt Len Max     1048342 non-null  int64
 9   Fwd Pkt Len Min     1048342 non-null  int64
 10  Fwd Pkt Len Mean    1048342 non-null  float64
 11  Fwd Pkt Len Std     1048342 non-null  float64
 12  Bwd Pkt Len Max     1048342 non-null  int64
 13  Bwd Pkt Len Min     1048342 non-null  int64
 14  Bwd Pkt Len Mean    1048342 non-null  float64
 15  Bwd Pkt Len Std     1048342 non-null  float64
 16  Flow Byts/s         1048342 non-null  float64
 17  Flow Pkts/s         1048342 non-null  float64
 18  Flow IAT Mean       1048342 non-null  float64
 19  Flow IAT Std        1048342 non-null  float64
 20  Flow IAT Max        1048342 non-null  float64
 21  Flow IAT Min        1048342 non-null  int64
 22  Fwd IAT Tot         1048342 non-null  int64
 23  Fwd IAT Mean        1048342 non-null  float64
```

2. Preprocessing

- replace +ve and -ve infinity with NaN
- drop missing values
- Implementing binary classification
- Balancing the data (Under sampling)

3. Perform Exploratory Data Analysis

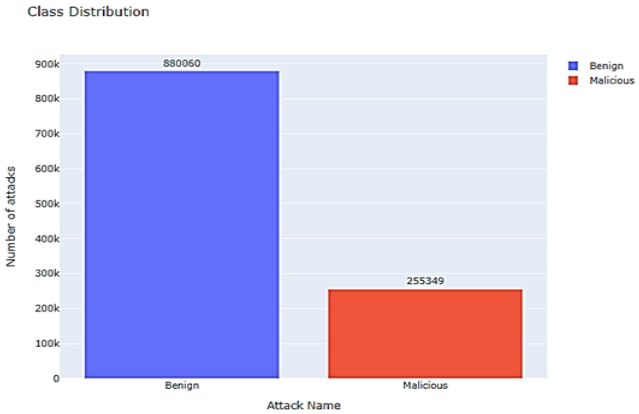

Figure 3: Number of Attacks

4. Data splitting

Data splits into 80:20 ratio 80% use for training and 20% use for testing.

Table 1: Hyper parameters Detail

Model	LSTM ,GRU
Activation	Relu
Epochs	20
Batch size	32
Metrics	Accuracy, Loss,
Units	64
Optimizer	Adam

Table-2 Performance Evaluation of DL Algorithm

Model	Loss	Accuracy
LSTM+GRU	0.0	99.46%

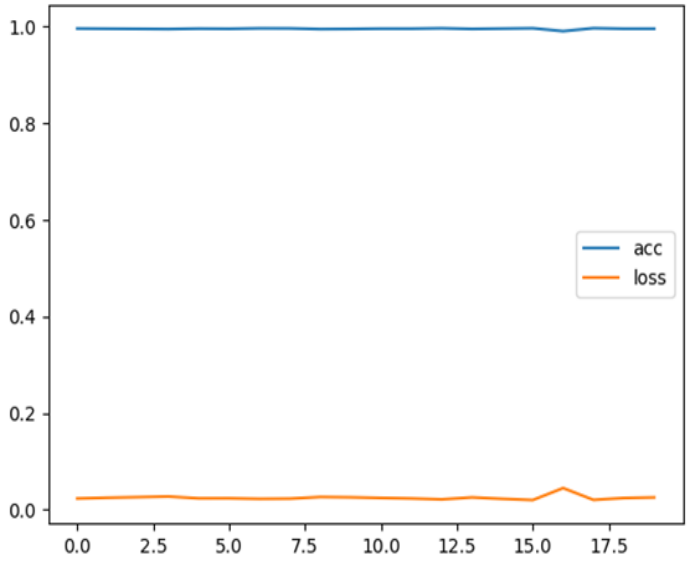

Figure 4: Accuracy and Loss of CIS IDS Dataset

Table 3 represents the method of KNN, Fuzzy, FNN, GRU, multilayered S. K. B Kezhou Ren et al. [1] and proposed method in terms of accuracy, recall and precision. Kezhou Ren et al. [1] shows an accuracy of 93.6%, a precision 98%, a recall 91% for KNN, an accuracy of 91.0% for fuzzy, an accuracy of 97.4%, a precision 92.5%, a recall 86.9% for FNN, an accuracy of 97.1%, a precision 95.8%, a recall 98.7 for GRU, an accuracy of 98.1%, a precision 99.8%, a recall 99.8 for multilayered. The proposed method shows an accuracy of 99.0%, an recall 99.89% and precision 99.87%. Clearly, the proposed method is a 1.07% improvement

accuracy compared to Kezhou Ren et al. [1]. Fig. 5.6, 5.7 and 5.8 shows the graphical representation of the comparison result.

Table 3: Compared Results

Method	Accuracy	Precision	Recall
Existing KNN	93.6%	98%	91%
Existing Fuzzy	91%	-	-
Existing FNN	97.4%	92.5%	86.9%
Existing GRU	97.1%	95.8%	98.7%
Existing Multilayered	98.02%	99.8%	99.8%
Proposed Method	99.46%	99.87%	99.89%

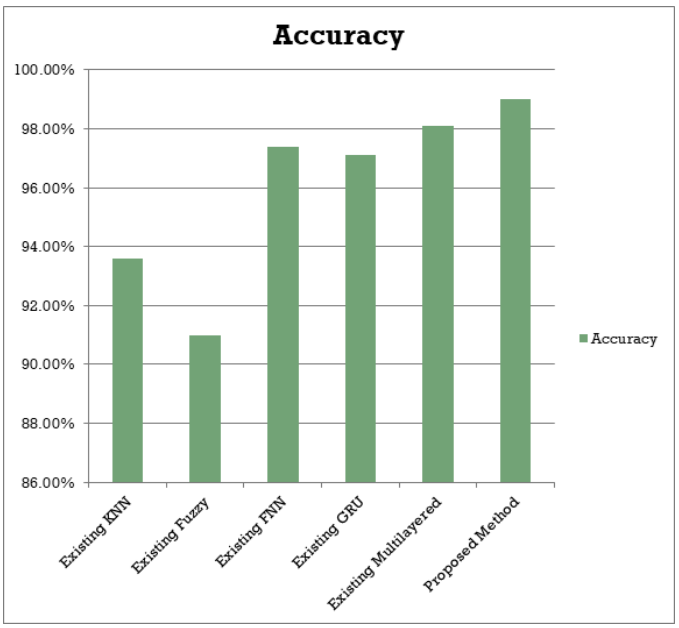

Figure 5: Graphical Accuracy

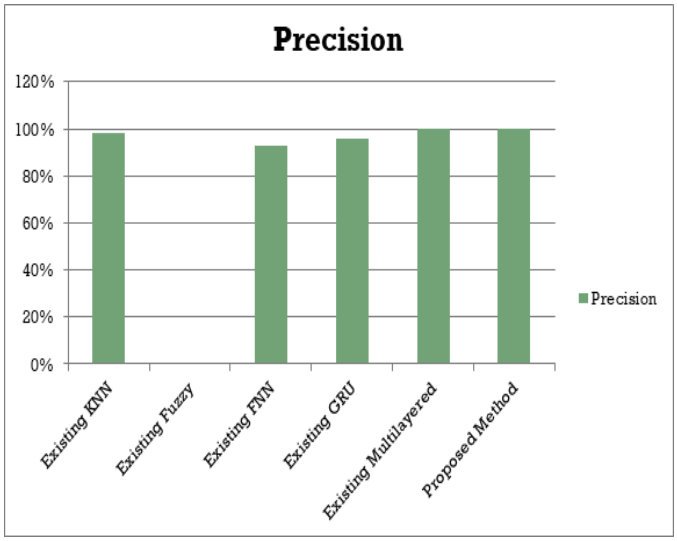

Figure 6: Graphical Precision

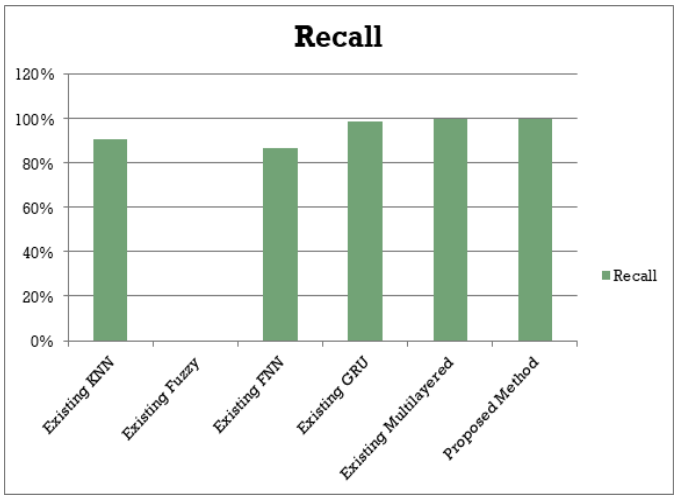

Figure 7: Graphical Recall

IV. CONCLUSION

Nowadays people are computer technology dependent which leads to vulnerabilities. To make data secure, we need a secure computer network system. All computers and networks are at risk of unauthorized access and insecurity of private and sensitive information. In the previous few decades, internet technology has extended its application space in many various domains in our life like banking operations, online auctions, electronic commerce applications, social networking, and online application/registration, etc. The models are evaluated on datasets individually and reached promising results. Results of this research are shown that MLP, RNN, LSTM and LSTM-GRU models produced good accuracy and loss. The proposed method is a 1.07% improvement accuracy compared to Kezhou Ren et al. [1].

REFERENCES

[1] Kezhou Ren, Maohuan Wang, Yifan Zeng and Yingchao Zhang, "An Unmanned Network Intrusion Detection Model Based on Deep Reinforcement Learning", IEEE International Conference on Unmanned Systems (ICUS), IEEE 2022.

[2] R. Ahsan, W. Shi, X. Ma, and W. L. Croft, "A comparative analysis of CGAN-based oversampling for anomaly detection," *IET* Cyberphysical Systems: Theory & Applications, vol. 7, no. 1, pp. 40–50, Mar. 2022.

[3] S. Dong, Y. Xia, and T. Peng, "Network Abnormal Traffic Detection Model Based on Semi-Supervised Deep Reinforcement Learning," IEEE Transactions On Network And Service Management, vol. 18, no. 4, pp. 4197–4212, Dec. 2021.

[4] Lan Liu, Pengcheng Wang , Jun Lin, and Langzhou Liu, "Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning", IEEE Access 2020.

[5] A. Raghavan, F. D. Troia, and M. Stamp, ``Hidden Markov models with random restarts versus boosting for malware detection," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 2, pp. 97107, Jun. 2019.

[6] Zhiyou Zhang and Peishang Pan "A hybrid intrusion detection method based on improved fuzzy C-Means and SVM", IEEE International Conference on Communication Information System and Computer Engineer (CISCE), pp. no. 210-214, Haikou, China 2019.

[7] Afreen Bhumgara and Anand Pitale, "Detection of Network Intrusion Using Hybrid Intelligent System", IEEE International Conferences on Advances in Information Technology, pp. no. 167-172, Chikmagalur, India 2019.

[8] Ritumbhira Uikey and Dr. Manari Cyanchandani "Survey on Classification Techniques Applied to Intrusion Detection System and its Comparative Analysis", IEEE 4th International Conference on Communication $ Electronics System (ICCES), pp. no. 459-466, Coimbatore, India 2019.

[9] Aditya Phadke, Mohit Kulkarni, Pranav Bhawalkar and Rashmi Bhattad "A Review of Machine Learning Methodologies for Network Intrusion Detection", IEEE 3rd National Conference on Computing Methodologies and Communication (ICCMC), pp. no. 703-709, Erode, India 2019.

[10] S. Sivantham, R.Abirami and R.Gowsalya "Comapring in Anomaly Based Intrusion Detection System for Networks", IEEE International conference on Vision towards Emerging Trends in Communication and Networking (ViTECon), pp. no. 289-293, Coimbatore, India 2019.

[11] Azar Abid Salih and Maiwan Bahjat Abdulrazaq "Combining Best Features selection Using Three Classifiers in Intrusion Detection System", IEEE International Conference on Advanced science and Engineering (ICOASE), pp. no. 453-459, Zakho - Duhok, Iraq 2019.

[12] Lukman Hakim and Rahilla Fatma Novriandi "Influence Analysis of Feature Selection to Network Intrusion Detection System Performance Using NSL-KDD Dataset", IEEE International Conference on Computer Science, Information Technology, and Electrical Engineering (ICOMITEE), pp. no. 330-336, Jember, Indonesia 2019.

[13] T. Sree Kala and A. Christy, "An Intrusion Detection System Using Opposition Based Particle Swayam Optimization Algorithm and PNN", IEEE International Conference on Machine Learning, Big Data, Cloud and Parallel Computing, pp. no. 564-569, Coimbatore, India 2019.

[14] Xiaoyan Wang and Hanwen Wang "A High Performance Intrusion Detection Method Based on Combining Supervised and Unsupervised Learning", IEEE Smart World, Ubiquitous Intelligence $ Computing Advanced $ Trusted Computing, Scalable Computing, Internet of People and Smart City Innovations, pp. no. 889-897, Guangzhou, China 2018.

[15] P. Singh and M. Venkatesan, "Hybrid Approach for Intrusion Detection System", IEEE International Conference on Current Trends Towards Converging Technologies (ICCTCT), pp. no. 654-659, Coimbatore, India 2018.

[16] M. Tavallaee, E. Bagheri, W, Lu and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 dataset", IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. no. 892-899, Ottawa, India 2018.
[17] Karuna S. Bhosale and Assoc. prof. Maria, "Data Mining Based Advanced Algorithm for Intrusion Detection in Communication Networks", IEEE International Conference on Computational Techniques, Electronics & Mechanical System (CTEMS), Belgaum, India 2018.