Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 45 No. 9
September 2024

Development of Node Cognitive Model for Identification of Malicious Nodes

Madhavi Dhingra[1], S C Jain[2] and Rakesh Singh Jadon[3]
[1]Amity University Madhya Pradesh, Maharajpura Dang, Gwalior (MP)
madhavi.dhingra@gmail.com,
[2]Amity University Manesar,(UP)
scjain555@gmail.com
[3]MITS, Gwalior
rsjadon@gmail.com

Abstract
Malicious nodes have been discovered in both wireless and wired networks. The examination of node activity is critical for securing the network's information. The anomalous intrusion detection mechanism examines the node activity and take appropriate action. The cognitives of the node are used to analyse malevolent nodes in this research. The neural network structure is used to analyse node behaviour. It identifies the node class using euclidean distance as a crucial element.. For attack nodes and attack-free nodes, the euclidean distance is calculated. The node behaviour is developed by analysing the node's characteristics. The node dynamics can be efficiently mapped with multilayer neural network that receives node attributes as input and computes the node class. The backpropagation procedure is performed by the multilayer-feed-forward neural network, which provides results for binary and multi-class output experiments.

Keywords: Wireless Networks, Cognitive Model, Information Security, Malicious Node, Intrusion Detection

1. Introduction

Any malicious node can have a significant effect on the communication of the entire computer network. A legitimate node may pose as a malicious node in a number of attacks, such as the Blackhole and Wormhole attacks, which compromise network performance. There is a dearth of study on identifying and interpreting node behavior in networks, despite the fact that many studies have been conducted to detect attacks in online systems. Therefore, it is crucial to identify the node's benign and malevolent behavior.

Nodes are the building blocks of wireless networks. The clustering technique helps create groups according to the categorization model when nodes fit into different categories, like harmful or normal. The performance of the network is lowered when a malicious node breaks security guidelines and negatively impacts the network.To construct an intrusion detection model for malicious attacks, a significant amount of testing and training time is usually required. A lot of research also show a significant false alert rate. These shortcomings suggest that an intelligent and secure method is required to accurately detect and evaluate the node's activities. The data in the form of packets or messages travelling over the network can be in normal form, or can be suspicious. These data can itself affect the network nodes intentionally or unintentionally and can degrade the performance of the network. The condition can worsen due to network crash down because of such harmful data. Such malicious contents must be identified in order to stop them from further transmission in the network. The data that is transmitted over the network is associated with some attributes or features. These features can help in detection of faulty or malicious data. The computer network is dependent on the nodes making the network. Each node is responsible for the status of the network. Thus, it is equally important in the network to identify the status of each and every node for safety of the network and its data. The identification of the malicious node is absolutely essential and must be accurate enough. Sometimes, the false alarm

detection rate is very high and therefore, it must be reduced during the process of localisation[1].

The malicious traffic started by the source node goes via many nodes, making some of them evil as well. Unsupervised mechanisms of machine learning like clustering is able to distinguish between the pure traffic and attack traffic instances in the network. The cluster's regular nodes and malicious nodes are separated by a predetermined distance. The traffic records that are not under assault are included in the pure traffic dataset. The network traffic must be analysed in order to investigate the behaviour of the nodes in the network In general, the nodes of the same cluster behave and structure in a similar way. The cognitive study of node activity is critical for determining the node's dynamic composition.

2. Background work

An IDS examines the incoming as well as outgoing traffic in order to detect any suspicious activities and report any aberrant behaviour[1]. Misuse detection and anomaly detection are two crucial types of intrusion detection. For those cases where previous information is present in the datasets that may be utilised to train the learning system, misuse detection is possible[2]. Anomaly detection is one such technology that adjusts to the network's usual operation and uses computational statistical statistics to specific activities to assess if an occurrence is regular or not[3].

Node behavior is a major area of research these days[4,5]. This distance is a crucial factor in identifying the various node behaviours. There are other distance functions, but Euclidean distance is the one that is most frequently used. Since the data is dense, Euclidean distance is utilised [6]. It helps in identifying the category of node by using the clustered dataset's features[7]. The node's behaviour is identified by its characteristics. Under certain circumstances, the malicious node can turn normal and the normal node can turn malicious. The node activity need to be dynamically analysed using the right approach.

The majority of current algorithms use the routing path to determine each node's reputation. They may not be false in practice, but they mostly rely on the supposition that all nodes on the same routing path have the same reputation, which can lead to erroneous detection outcomes. We approach this topic as a multivariate multiple linear regression problem and employ the K-means classification algorithm to identify harmful nodes in order to solve it[8].

Based on game theory and reputation, a method is put out for identifying the nefarious and self-centered nodes. The three stages of the suggested approach are setup and clustering, data transmission and multiplayer gaming, and updating and malicious node detection[9].

To prevent malicious nodes like interrupt attack and selective forwarding attack nodes, a trust and reputation-based approach for identifying malicious nodes is introduced along with environmental parameters (TRS&EP). Environmental parameters can be solved by integrating the energy of nodes, data volume, number of neighboring nodes, node sparsity, and other deterministic parameters with machine learning's linear regression[10].

Additionally, a fuzzy trust model and artificial bee colony algorithm (ABC) are the foundations of a malicious node detection technique that is proposed (FTM-ABC). In order to determine indirect trust, the fuzzy trust model (FTM) is presented. The ABC method is then used to optimize the trust model in order to identify dishonest recommendation assaults. Additionally, to increase effectiveness, the fitness function adds the recommended deviation and the interaction index deviation[11].

In decentralized networking, blockchain—a shared distributed ledger—can foster cooperation amongst transaction processing and interaction devices as well as aid in the creation of an adaptable, scalable, private, secure, and dependable automotive networking system. This study proposes a malicious node detection mechanism based on block networks. A security plan that can guarantee smooth communication between network vehicles might be created by utilizing blockchain technology in an automobile network for the malicious node identification algorithm[12].

Additionally, a correlation theory-based malicious-node identification technique that thwarts fault data injection assaults is put forth. First, using time correlation, abnormalities between comparable kinds of sensor data are found. Second, spatial

correlation is used to identify malicious nodes. Third, event correlation is used to validate the malicious nodes that have been detected[13].

Every registered node in the network has its trust factor determined by the suggested model. The process of node authentication and the identification of malicious nodes within the network both make use of the trust factor. A dynamic authentication system built into the proposed primary security module enables existing nodes to authenticate newly arriving nodes. This process creates secure links and spreads authentication across nearby nodes. The authentication method stops malicious external nodes from entering the system. This study proposes a Trusted Node Feedback based Clustering model for Malicious Node Detection (TNFC-MND) to identify and eliminate malicious nodes from the network[14].

3. Computation of Euclidean Distance from Clustered Dataset

The wireless network contains n number of nodes. When nodes are of different cate- gory like malicious or normal, then clustering process helps in making the clusters according to the category. The malicious traffic initiated by the source node travels through different nodes and make some other nodes also malicious. The normal nodes and the malicious nodes present in the cluster are separated by some specific distance. This distance is important parameter so as to identify different behavior of nodes. The distance function are of several types, of which Euclidean distance is the most commonly used. Euclidean distance is used as the data is dense. The euclidean distance is calculated between the attack node and the attack free node from the clustered dataset by using its features[15].

In our case, the attacking node identified from the clustered dataset start with IP ad- dress 175.45.176.p where p = 0,1,2,3. The attack free node identified from the clustered dataset start with IP address 59.166.0.q, where q = 0,1,2,3,4,5,6,7,8,9. There are some other IP addresses also, but only first instance of both dataset are required to compute the distance value. Let dij denote the distance between two nodes i and j, we assume i = attack free or normal node and j = attack node or malicious node.

The euclidian distance is calculated by the formula [16]-

$$Dij = \sqrt{(xj - xi)2 + (yj - yi)2 + \quad + (z\ j\ -\ zi\ )2}$$

Equation 1

where x, y and z represents the features.

Equation 1 is used for comparing two “objects” across a range of variables and this will determine the similarity or dissimilarity of the objects. The features used from the clustered dataset are scrip, dstip, sbytes, sttl, dttl, tcprtt, synack, ackdat, and ctstate_ttl, total 9 features. Thus the equation 1 is transformed to equation 2 after incorporating all the features of the clusters.

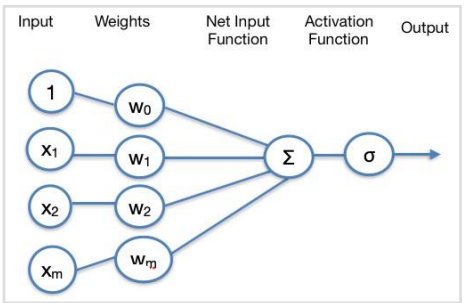
$$Dij = (sj - si)2 + (dj - di)2 + \ldots$$

Equation 2

where sj is scrip feature of non-attack traffic record and si is scrip feature of attack traffic record.

4. Generation of Node Behavior Pattern

The node features identify the behavior of the node. The malicious node may become normal and the normal node may become malicious under specific conditions. The node behavior is dynamic and must be analysed by suitable methodology. To generate the node behavior, Artificial Neural Networks(ANN) are used[17,18] and the different features of the traffic records of the network are provided as input. This network may be used to train the network in order to detect malicious nodes. The general organisation of the ANN is depicted in Figure 1. The network is used to create patterns between features and to analyse how



those patterns are employed.

Fig. 1. Simple Neural Network

For calculation of the neuron weights (wi), euclidean distance is used. Traffic current trends are provided as input (xi).-

y = Σ Xi wi, say networkA where

y - standing for the neural network's output. The input vector is indicated by Xi in terms of nodes.

j stands for the quantity of hidden neurons.

The weights collected are indicated by wi.

Weighted distance is used to identify malicious nodes. Let n1, n2, n3, n4, and nk be the k-nearest neighbours of node x, where w1, w2, w3, w4, and wk are the weights associated with each node's IP address.

The weighted distance for object x is defined as w1d1, w2d2, w3d3, w4d4,..... wkdk, where d1,d2,d3,d4.....dk are the normal euclidean distances, d (x,si) = wi di.

The average weighted distance = 1/k Σ wi di/ Σ wi

Malicious nodes are those whose distance is larger than the average weighted distance.

5. Cognition Analysis using Multilayer FeedForward Neural Network (MFNN)

MFNN analyses the patterns created for each node. The network's learning process affects the identifying process. Three layers make up the MFNN; the first - input layer contains the node's features, there are some hidden layers and the output depicts the category of the nodes. Figure 2 depicts the network.

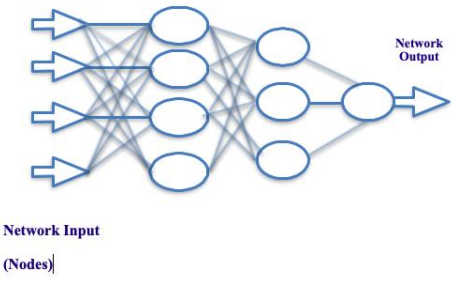


Fig. 2. Multilayer-feed-forward Neural Network

Initially, the input data is sent to the middle-hidden layer of neurons together along with the weights. Based on the supplied input value, each neuron calculates the output. The third-output layer is generated from previous input neuron. From left to right, information travels. The membership function of the artificial neural network is created as a sigmoidal function, and the output is calculated from the inputs' weighted sum. Backpropagation algorithm is used by MFNN [19].

The node from the normal category present in the cluster may move to the attack category cluster, and the similarly the node can migrate from the attack category to normal one. All of these modifications are dependent on the node's activity, which is dynamic in a network without wires. As a result, node behaviour might be described as "fuzzy" since it lies between two distinct regions, and a node's values determine where it is in the cluster.

In this case, the membership function (u), which categorises nodes into normal and malicious categories. The tanh sigmoidal function depicted in Figure 3 defines the neural network activation function. The tanh function's range is from (-1 to 1). Tanh is sigmoidal as well (s - shaped).

f(x) = tanh(x) = 2/(1 + e-2x) - 1

The function is mapped on the neural network as f(networkA) = {+1, if networkA >= 0, -1, if networkA < 0}

Threshold 'Θ' is a factor is used to determine the activations of the specified network.f(networkA) = {+1, if networkA >= Θ, -1, if networkA < Θ}

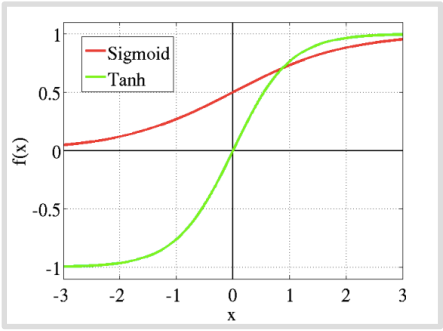


Fig. 3. tanh function (sigmoid another version) [20]

During the training phase, the neural network is trained using the EFS-selected traffic features as input[21]. The distance between each of the nodes is calculated by the neural network and analysed with average weighted distance. Nodes are identified as abnormal or malevolent which have greater distance than the calculated weighted distance.

During Testing, identification of new nodes is done via neural network learning. The output is classified as either malicious or normal.

The performance of the neural network is evaluated on the basis of the three important factors namely, Number of Neurons present in the input layer, Learning Rate and Number of

iterations. The effective behavioural pattern can be generated by having less learning rate[22].

The total number of neurons shown in the first-input layer, learning rate, and total number of epochs (iterations) are three critical metrics used to evaluate the neural network's success. A slower pace of learning can result in a more successful behavioural pattern[23].

6. Cognition Analysis using Multilayer FeedForward Neural Network (MFNN)

Initially, the input data is sent to the middle-hidden layer of neurons together along with the weights. Base

6.1 Dataset with Two Classes

The UNSW-NB15 dataset has been processed by EFS algorithm and k-means algo- rithm by taking class as label (normal or malicious). The clustered dataset received as output is taken as input dataset. The resulting dataset has formed two clusters one for normal class and another for malicious class. The ensemble feature selection algorithm[21] and the k-means method are used to process the UNSW-NB15 dataset[24]giving output as normal or malicious. The clustered dataset is taken as input dataset. The resulted clustered dataset has two clusters: one for the normal node and one for the malicious node. The clusters generated in the dataset are shown in figures 4(a) and (b).

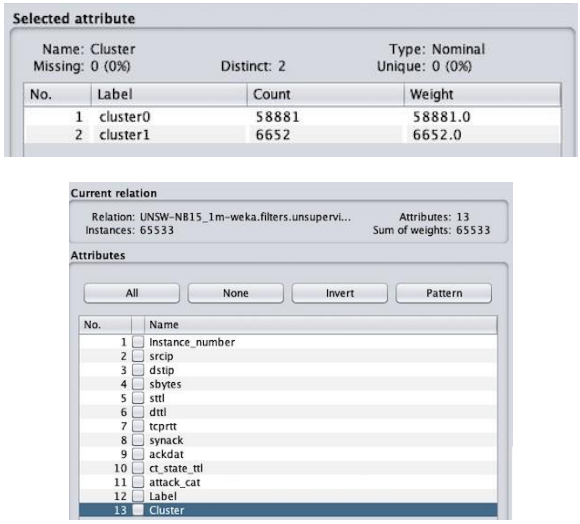


Fig. 4 (a) Two Class Clustered Dataset attributes
(b) Two Class Clusters

MultiLayerPerceptron is used to classify the dataset, and several configurations are tested. Neurons denoting instances are used as the input, and the neural network's performance is assessed using various learning rates and iterations. The dataset has 65533 records with 13 features. Percentage split is used on the dataset where 66% of the total is taken as training dataset and remaining as testing dataset.

Variable hidden layers and epochs - The default setting value of momentum and learning rate are 0.2 and 0.3, respectively. The total number of hidden layers to be added in the neural network is calculated using the formula a = (total attributes + classes)/2. With various numbers of iterations and hidden layers, a 99 percent accuracy rate is reached.

Variable learning rate and momentum - The letter "a" represent the hidden layers, and the epochs number are set to 100 by default. The accuracy rate is 99 percent with default settings of learning rate and momentum values, but when the learning rate and momentum are increased, the accuracy rate begins to decline.

Table 1: Results of MLP with different iterations and hidden layer in two clusters

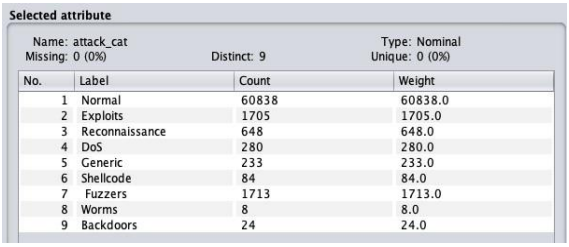
Epochs (Number of Iterations)	Number of Hidden Layers	Accuracy rate	RMSE
100	a	99.99	0.0054
10	a	99.99	0.0048
100	1	99.99	0.0068
10	1	99.99	0.0068
200	a	99.99	0.0053

Table 2: Results of MLP with different learning rate and momentum in two clusters

Learning Rate	Momentum	Accuracy rate	RMSE
0.3	0.2	99.99	0.0054
0.5	0.4	99.99	0.0062
0.7	0.6	99.99	0.0063
0.9	0.8	89.32	0.3268

6.2 Dataset with Nine Classes

Additionally, the EFS method and the k-means algorithm, using the class attack have been used to process the UNSW-NB15 dataset (nine attack categories)[21]. 13 features, including the class attack will be part of the adjusted dataset. Figure 5 displays the nine attack categories.

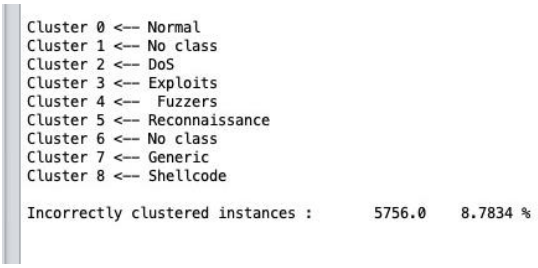

Selected attribute

Name: attack_cat Type: Nominal
Missing: 0 (0%) Distinct: 9 Unique: 0 (0%)

No.	Label	Count	Weight
1	Normal	60838	60838.0
2	Exploits	1705	1705.0
3	Reconnaissance	648	648.0
4	DoS	280	280.0
5	Generic	233	233.0
6	Shellcode	84	84.0
7	Fuzzers	1713	1713.0
8	Worms	8	8.0
9	Backdoors	24	24.0

Fig. 5. Dataset with Nine different attacks

After applying KMeans clustering algorithm, the results indicate that 91.22 percent of the instances are correctly classified. The attacks are divided into nine categories of clusters, including "normal," "exploits," "reconnaissance," "doS," "generic," shell code, buzzers, worms, and backdoors. The results are displayed in Figures 6.


```
Cluster 0 <-- Normal
Cluster 1 <-- No class
Cluster 2 <-- DoS
Cluster 3 <-- Exploits
Cluster 4 <--  Fuzzers
Cluster 5 <-- Reconnaissance
Cluster 6 <-- No class
Cluster 7 <-- Generic
Cluster 8 <-- Shellcode

Incorrectly clustered instances :       5756.0    8.7834 %
```

Fig. 6. Output of KMeans Clustering on EFS applied Nine class Dataset

The clustered dataset is archived and used as Multilayer Perceptron's input. The number of instances define the neurons present in the first-input layer. The performance is tested by setting the values of the learning rate and iterations. There are 65533 records having 14 features. The percentage split concept having 66 % of dataset is taken as training-dataset and the remaining is taken as testing dataset.

Variable epochs and hidden layers - The learning rate is 0.3 and the momentum is 0.2 in the initial stage.With more number of hidden layers specified as “a”, 99 percent accuracy rate is attained. However, with fewer layers, a low accuracy of 91 percent is attainable.

Variable learning rate and momentum - The default set values for the hidden layers and epochs are "a" and 100, respectively. With the default values of learning rate and momentum, 99 percent accuracy rate is reached. However, when the values are increased, a poor accuracy of 88 percent is reached.

Table 3 Results of MLP with different iterations and hidden layer in nine clusters

Epochs (Number of Iterations)	Number of Hidden Layers	Accuracy rate	RMSE
10	1	91.459	0.1028
100	1	91.459	0.1026
10	a	99.98	0.0072
100	a	99.99	0.0032
200	a	99.99	0.003

Table 4 Results of MLP with different learning rate and momentum in nine clusters

Learning Rate	Momentum	Accuracy rate	RMSE
0.3	0.2	99.99	0.0032
0.5	0.4	100	0.0015
0.7	0.6	88.74	0.1546
0.9	0.8	88.74	0.1582

7. Conclusion

The node behavior is very dynamic. It can vary depending on the environmental attributes. Keeping this principle in mind, the cognitive analysis is required so as to determine it. This research has studied the dynamic behavior of network by using neural network structure. It employs euclidean distance as a major parameter for determining the class of node. The euclidean distance is computed for attack nodes and the attack- free nodes. The node behavior is generated by analysing the features of the node. The neural network structure

is described by Multilayer feed forward network that takes the node features as input and gives the class of node as output. The membership function of the subsequent network structure is also described. The MFNN performs the backpropagation algorithm and provide the results. Using a cognitive model of node activity, malicious nodes are determined from the network traffic dataset. The approach uses a multilayer-feed-forward neural network with a back-propagation method to classify network nodes by processing, clustering, training and testing the dataset. The processed clustered dataset is taken with two classes (normal and malicious) and nine classes in the experimental study. The deployment of neural networks for the analysis of node activity and the identification of malicious traffic is actually justified by the results obtained. Artificial intelligence techniques have wider scope of research in information security. This paper has used Multilayer-feed-forward Neural Network for identification of Cognitive behavior of node. The datasets can be experimented with multiple configuration of classifiers in future.

References

1. Mishra, A., Nadkarni, K. & Patcha, A. (2004), Intrusion detection in wireless ad hoc networks", IEEE wireless communications.
2. Bahrololum, M., Salahi, E., & Khaleghi, M. (2009), Machine Learning Techniques for Feature Reduction in Intrusion Detection Systems: Comparison, 2009 Fourth Int. Conf. Comput. Sci. Converg. Inf. Technol.
3. Hossein, M. S. (2009), Anomaly Intrusion Detection System Using Information Theory, K-NN and KMC Algorithms, Australian Journal of Basic and Applied Sciences, vol 3(3), pp. 2581-2597.
4. Lal, N., Kumar, S., Saxena, A., Chaurasiya & V. K. (2015), Detection of Malicious Node Behaviour Via I-Watchdog Protocol in Mobile Ad hoc Network With DSDV Routing Scheme, International conference on Advances in Computing, Communication and Control, vol 49, pp. 264-273.
5. Mamatha, S. & Damodaram, A. (2014), Intrusion Detection System for Mobile Adhoc Networks Based on the Behaviour of Nodes, International Journal of Grid Distribution Computing, vol 7, issue 6, pp. 241-256.
6. Tuliao, K.D.L. (2005), Euclidean Distance – Raw, Normalised, and Double-Scaled Coefficients, The Technical Whitepaper series, [Online]. Available: https://www.pbarrett.net/techpapers/euclid.pdf.
7. Kalid, A., Marsal, A., Azni, H. & Ridzuan, F. (2019), Formulation of 3D Euclidean Distance for Network Clustering in Wireless Sensor Network, Vol.8, Issue 9, COMPUSOFT, International Journal of Advanced Computer Technology, pp. 3368-3373.
8. Liang Liu, Jingxiu Yang, Weizhi Meng, (2019), Detecting malicious nodes via gradient descent and support vector machine in Internet of Things, Computers & Electrical Engineering, Volume 77, Pages 339-353, ISSN 0045-7906.
9. Nobahary, S., Garakani, H.G., Khademzadeh, A. et al.(2019), Selfish node detection based on hierarchical game theory in IoT. J Wireless Com Network 2019, 255.
10. Z. Teng, B. Pang, C. Du and Z. Li, (2020), Malicious Node Identification Strategy With Environmental Parameters, in IEEE Access, vol. 8, pp. 149522-149530, doi: 10.1109/ACCESS.2020.3013840.
11. B. Pang, Z. Teng, H. Sun, C. Du, M. Li and W. Zhu, (2021), A Malicious Node Detection Strategy Based on Fuzzy Trust Model and the ABC Algorithm in Wireless Sensor Network, in IEEE Wireless Communications Letters, vol. 10, no. 8, pp. 1613-1617, doi: 10.1109/LWC.2021.3070630.
12. Chen, J.; Li, T.; Zhu, R. (2022), Analysis of Malicious Node Identification Algorithm of Internet of Vehicles under Blockchain Technology: A Case Study of Intelligent Technology in Automotive Engineering. Appl. Sci. 12, 8362.
13. Yingxu Lai, Liyao Tong, Jing Liu, Yipeng Wang, Tong Tang, Zijian Zhao, Hua Qin, (2022),Identifying malicious nodes in wireless sensor networks based on correlation detection, Computers & Security, Volume 113, 102540, ISSN 0167-4048.
14. Madhuri, Simhadri & Lakshmi, Sanapala. (2023). A TRUSTED NODE FEEDBACK BASED

CLUSTERING MODEL FOR DETECTION OF MALICIOUS NODES IN THE NETWORK. Journal of Theoretical and Applied Information Technology. 101. 2686-2697.

15. Chahar, Vijay & Chhabra, Jitender & Kumar, Dinesh. (2014). Impact of Distance Measures on the Performance of Clustering Algorithms. 10.1007/978-81-322-1665-0_17.
16. Rosso, Gianluca. (2014). Outliers Emphasis on Cluster Analysis - The use of squared Euclidean distance and fuzzy clustering to detect outliers in a dataset.
17. Liu, Y. , Tian, D. & Wang, A. (2003), ANNIDS: intrusion detection system based on artificial neural network, In Proceedings of the Second International Conference on Machine Learning and Cybernetics. vol. 3. IEEE, Amsterdam, pp. 2-5.
18. Linda, O., Vollmer, T. & Manic, M. (2009), Neural network based intrusion detection system for critical infrastructures, In Proceedings of IEEE International Joint Conference on Neural Networks, Georgia. IEEE, Amsterdam, pp.102-109.
19. Saurabh. (2019), Backpropagation – Algorithm For Training A Neural Network, [Online].Available: https://www.edureka.co/blog/backpropagation.
20. Sharma, S. (2017), Activation Functions in Neural Networks, Sigmoid, tanh, Softmax, ReLU, Leaky ReLU EXPLAINED, [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.
21. Dhingra M, Jain, S C and Jadon, R S, Thesis online at Shodhganga(2021), available online at : http://shodhganga.inflibnet.ac.in:8080/jspui/bitstream/10603/343140/8/madhavi_thesis_chapter%205.pdf.
22. Jabez, J., Gowri, S., Vigneshwari, S., Albert, Mayan J., Srinivasulu, S. (2019), Anomaly Detection by Using CFS Subset and Neural Network with WEKA Tools, In: Satapathy S., Joshi A, Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 107. Springer, Singapore.
23. Moustafa, N. & Slay, J. (2015), The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set, Information Security Journal: A Global Perspective.
24. Moustafa, N. & Slay, J. (2015), The Significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems, BADGERS@RAID, pp. 25-31.