An Approach to Improving Intrusion Detection System Performance Against Low Frequent Attacks

Yasir A. Mohamed [1,*], Dina A. Salih [2], and Akbar Khanan [1]

[1] A'Sharqiyah University, CoBA, Ibra, Oman; Email: akbar.khanan@asu.edu.om (A.K.)
[2] Faculty of Mathematical and Computer Sciences, University of Gezira, Medani, Sudan;
Email: oimana606@yahoo.com (D.A.S.)
*Correspondence: Yasir.abdulgadir@asu.edu.om (Y.A.M.)

***Abstract*—Network security is crucial in contemporary company. Hackers and invaders have regularly disrupted huge company networks and online services. Intrusion Detection Systems (IDS) monitor and report on harmful computer or network activities. Intrusion detection aims to detect, prevent, and react to computer intrusions. Researchers have suggested the fuzzy clustering-artificial neural network to improve intrusion detection systems. A hybrid Artificial Neural Network technique combines fuzzy clustering and neural networks to increase intrusion detection systems' accuracy, precision, and resilience. We built fuzzy clustering modified artificial neural networks to increase low-frequency attack detection and training time. This approach can be improved in terms of training duration and low-frequency attack accuracy. Our novel technique, Fuzzy Clustering-Artificial Neural Network-modified, beats the fuzzy clustering-artificial neural network algorithm by 39.4% in identifying low-frequent assaults and decreases the projected training time by 99.7%.**

***Keywords*—Intrusion Detection Systems (IDS), low frequent attack, fuzzy clustering-artificial neural network**

I. Introduction

Web applications like online shopping, auctions, and banking must communicate data and resources securely over a network. This data must be protected against abuse and theft. Firewalls, antivirus software, and Intrusion Detection Systems (IDSs) have the same aim. An IDS helps resolve network vulnerabilities. Data records from network processes are sifted for cyberattacks. IDS detection accuracy and consistency are key metrics [1]. Many investigations have improved detection accuracy and stability. Early-stage research focuses on expert systems and statistics. Rule-based expert systems and statistical approaches degrade with bigger datasets. Many data mining approaches have been developed [2]. Artificial Neural Networks (ANNs) are a form of machine learning method used to categorize data; they're effective when the task is too difficult to design by hand. Instead, the neural network is taught to emphasize class-specific characteristics [3]. Neural networks are effective in detecting intrusions in IDS. In most circumstances, intrusion detection systems don't actively prevent invasions; rather, they inform system administrators of a possible security breach, making them a proactive tool. IDSs may be host-based, network-based, online, or offline. Third, a mistake or abnormality led to the allegation. A host-based IDS takes data from computer log files, whereas a network-based IDS analyzes data packets. An online IDS raises a flag if it detects an intrusion in process, whereas an offline IDS analyzes records after the fact and raises a flag if a security breach has happened after the previous check. Two forms of IDS detect harmful activity: anomaly-based and misuse-based [4].

When attack frequency reduces, ANN becomes susceptible. Low-frequency assaults have an insufficient learning sample size. Because ANN can't easily learn these assaults' properties, detection accuracy is low. Rare assaults aren't unimportant. If these assaults succeed, disaster will ensue [5]. If a User-to-Root (U2R) attack is successful, the attacker has total control over the compromised system or network appliance and may conduct any root-level operation. In IDS, uncommon assaults are the norm. ANN is intrinsically unstable since it converges to the local minimum [6]. Despite advancements in IDS accuracy and stability, low-frequency assaults remain a concern for most IDS systems.

Low-frequency assaults continue to be an issue and a difficulty for most IDS systems, despite the fact that some researchers have developed a novel strategy employing Artificial Neural Networks-based fuzzy clustering algorithm to improve the accuracy and stability for the intrusion detection system. As such, a modified version of the Fuzzy Clustering approach based on Artificial Neural Networks (FC-ANN-MD) has been provided for the same goal [7].

The next part provides an overview of the intrusion detection system, followed by a literature study and related studies. Afterwards, a part describing the materials and

Manuscript received December 24, 2022; revised January 31, 2023; accepted March 4, 2023; published May 24, 2023.

methods in addition to a descriptive analysis precedes the section on implementation, following which the findings are examined, and the research concludes with a section on the conclusion.

II. Related Work

Malware evolution puts IDS design to the test (IDS). Malicious assaults have become more sophisticated, with the most difficult difficulty being recognizing unknown and camouflaged software. To escape detection by intrusion detection systems, malware writers use a variety of information-concealing strategies. The number of zero-day assaults on internet users has also grown [8]. Computer security is becoming more important as we utilize more information technology.

Any activity conducted without authorization that causes damage to a computer system is considered an intrusion. This implies that any effort to undermine the security of the information, whether in terms of privacy, integrity, or accessibility, will be considered an incursion. An intrusion, for example, is any activity that prevents legitimate users from using computer services.

An Intrusion Detection System (IDS) is software or hardware that detects infiltration attempts on a network. An IDS is a form of firewall that detects threats that traditional firewalls cannot detect, this is a critical step in achieving a high degree of security against assaults that compromise computer system stability, authenticity, or secrecy [9].

As one of the most effective machine learning techniques, ANN has been successfully used to the detection of a broad variety of malware. While ANN-based IDS has greatly improved its ability to identify attacks, particularly less common ones, there is still potential for improvement. Since the training sample is smaller for less common attacks, it is more difficult for the ANN to appropriately understand their attributes. The likelihood of seeing assaults that occur less often will thus decrease, in reference to the integrity of digital data [10].

Fuzzy clustering is an unsupervised method used to partition the training data into smaller groupings. We utilize these subsets to teach ANNs. As the data size decreases, the amount of time needed to train each ANN also decreases. When these ANNs' outputs are aggregated by a final aggregating ANN, the detection rate rises because the final aggregating ANN fixes the misclassifications made by the various ANNs. With this in mind, we aim to increase the detection rate of attacks while decreasing the amount of time spent in training [11].

Appasha and Ghatule [12] describes a similar Fuzzy Clusters-Artificial Neural Network (ANN) structure, based on FC-ANN with system restore points. System Restore enables you to restore the cloud server to a prior state, including all registry keys, system files, saving and promoting, and the project database, in the event of a computer failure or malfunction or system infiltration. In cloud computing settings, the CloudIDS Generic Cloud Security Structure handles security duties. CloudIDS protects virtual servers and instances in many ways.

Zhong *et al.* [13] focuses on large data security. Massive data set security increases quicker than a node's CPU. Distributed computing improves accuracy and performance. Cloud-based intrusion detection includes a monitoring server, Hadoop master server, IDS server, node, and terminal administration. Hadoop-based intrusion detection performs better in experiments. This study optimizes neural network weights. Hadoop delivers genetic and neural network algorithms to the cloud. Improved algorithms identify intrusions better. Intrusion detection protects application systems against assaults, according to these studies.

Creating attack detection systems utilizing machine learning and data mining is the most popular way to halt network infiltration, according to Li and Qu *et al.* [14]. These technologies protect networks by identifying and blocking harmful traffic. This research uses the KDD 99 benchmark dataset to analyze fuzzy logic and neural network intrusion detection systems. Both the FC-ANN-based and hierarchical SOM-based approaches increased detection rates, as demonstrated in the accompanying table. TSK-based intrusion detection showed the greatest detection performance across normal, DoS, and probing classes.

Samrin and Vasumathi [15] presents an intrusion identification system that incorporates Weighted K-means Clustering (WKMC) and ANN. This study discusses clustering and intrusion detection. WKMC in the clustering module clusters the input dataset. Intrusion detection module stores the clustered data's ANN-trained structure. Selecting an ANN classifier based on distance or similarity metrics that best matches the test data cluster. Using a benchmark database, they discovered the suggested strategy was more accurate than existing methods.

Ashfaq *et al.* [16] divide unlabeled samples and their categorization results according on fuzziness. SSL was designed to boost classifier efficiency on ID datasets. NNR was chosen as the basis classifier because to its outstanding learning performance and minimal computing cost. Weights and biases for NNR's "hidden nodes" are randomly chosen. This research focuses on increasing classification accuracy by analyzing the link between fuzziness and misclassification. Experiments have shown that training the NNR to generate fuzzy vectors and classifying unlabeled data based on their fuzziness may increase classification accuracy. After adding unidentified samples and their projected labels (from low and high fuzziness groups) to the current training set, the classifier is retrained. This research demonstrates that IDSs misclassify more often with middle-fuzziness samples. In this study, we explored ordinary and uncommon examples.

In order to enhance detection accuracy and stability, decrease false positives, and increase the detection of rare attacks, Amini *et al.* [17] presents a unique ensemble classifier that integrates RBF neural networks with fuzzy clustering. Combining basic classifier predictions improves detection accuracy. Their suggested approach provides greater detection accuracy than the best available classification techniques, according to NSL-KDD test

results. It's more susceptible to infrequent assaults. The suggested approach outperforms conventional ensemble methods.

FC-ANN is an ANN and fuzzy clustering-based intrusion detection technique [18]. Fuzzy clustering divides the training set into comparable groupings. Reduced sub-training set complexity enhances detection performance. By doing so, the system becomes more efficient and stable despite overcoming decreased detecting accuracy and stability. System restoration points enable for dependable backups.

It's generally agreed that intrusion detection is a major challenge in the cloud. Although current methods can identify common threats, they struggle when it comes to less common ones. Optimal type-2 fuzzy neural networks (OT2FNNs) and Kernel Fuzzy C-means Clustering (KFCM) have been presented as a unique cloud-based Intrusion Detection System (IDS) to solve this problem. The researchers do this by making judicious use of the Lion Optimization Algorithm (LOA) for weight optimization while setting the parameters of T2FNN. The suggested IDS can detect an intrusion and ensure that only legitimate data is kept in the cloud. The proposed IDS system outperforms state-of-the-art IDS methods in terms of accuracy, recall, and F-measure [19].

A better training method for anomaly identification in unlabeled sequential data has been created [20], like time-series, to tackle one of the most difficult issues in the business. Under normal circumstances, the researchers claimed that the results obtained by a well-designed system are picked at random from a distribution whose parameters are not known. The probability criteria that a data-point is taken from has been proposed, and it is based on the classical central limit theorem. This allows for dynamic data labeling. A deep Long Short-Term Memory (LSTM) auto encoder is trained on normal data and can tell when the reconstruction error is too high and identify abnormalities. Two real-world industrial case studies with both slow-developing and sudden abnormalities were used.

Due of the high number of data, the network becomes extended with false alarm rate of intrusion and detection accuracy lowered. Unknown events cause this. The main goal in [21] was accuracy and false alarm reduction (FAR). Crow Search Optimization with Adaptive Neuro-Fuzzy Inference System (CSO-ANFIS) addresses these issues. A fuzzy interference system, and an artificial neural network are all optimized using the crow search optimization approach. The proposed model's intrusion detection performance was evaluated and compared to existing approaches using the NSL-KDD data set. The intrusion detection based on the NSL-KDD dataset has a 95.80% detection rate and 3.45% FAR, outperforming those models.

Tammi and Biswas *et al.* [22] uses k-means and neural networks for intrusion detection. In order to improve outcomes, the benchmark dataset was split into a training and testing set and grouped into five categories. After obtaining cluster data, several kinds of Artificial Neural Networks (Feed Forward, Elman, Generalized Regression, Probabilistic, and Radial Basis) were utilized to train the system (RBNN). They compared various functions and chose the most accurate. In this scenario, clustering may improve the probabilistic neural network and Fuzzy Neural Network. This result emphasizes the importance of selecting accurate feature sets.

Ambikavathi and Srivatsa [23] describes cloud computing as a "network of networks" spanning the internet, making it vulnerable to sophisticated assaults. Current technologies can't prevent security breaches. Intrusion detection is vital to network security. IDS may reduce the number of workers required for monitoring, enhance detection efficiency, provide data not otherwise available, educate the information security community on new dangers, and act as court evidence. FC-ANN is an ANN-and-fuzzy-clustering-based intrusion detection method. Fuzzy clustering clusters the training set. This minimizes sub-training set complexity to improve detection.

Khazaee and Rad [24] suggested technique enhances classifier performance using fewer features and a dynamic fuzzy C means algorithm. KDD Cup 99 is used for intrusion detection. Before using the suggested technique, they normalized KDD 99 training and test datasets. Fuzzy C enhances clustering performance. Performance is evaluated by comparing the suggested approach to others. Experiments show the suggested method's excellent accuracy and rapidity. WEKA's Java packages contain associations, classifiers, clusters, etc. Unimplemented FCM. WEKA's FCM and IDFCM algorithms were used. We modify FCM's first cluster center selection technique to intrusion data. Improved DFCM converges quicker with fewer mistakes than FCM. DFCM beats K-means and FCM.

The chaotic ant optimization (CAO) method is used to conduct optimum cluster formation in the suggested EIDR system [25]. The second innovation is a method for assigning confidence levels to individual sensor nodes using multi-objective differential evolution (MODE). The calculated trust value is utilized to create the Intrusion Reaction Action (IRA) system, which provides extra functions and varied response characteristics to reduce intrusion consequences. The suggested EIDR system enhances detection and false positive rates while keeping network performance stable, according to the simulation results.

Formal specification logic and a novel immune-inspired security architecture (I^2MANETs) [26] provide secure and reliable broadband services. A synchronizing agent lets federated domain immune components replicate, monitor, detect, classify, and block/isolate problematic packets and nodes. The framework has immunological traits including first and second response, adaptability, distributability, survival, and others. I2MANETs may propagate to all network nodes after installation on one node [27–30].

III. Proposed System

We first describe the full new procedure. The core three components are fuzzy clustering, artificial neural network, and fuzzy aggregation.

As with other machine learning frameworks, FC-ANN has a training and testing phase. Three separate epochs:

A. Stage One

Random data is turned into a training set (T_R). Fuzzy clustering provides various T_R training subsets (T_{R1}, T_{R2}, …, T_{RK}).

We cleansed the data for clustering and training. First, fuzzy clustering. After sorting the data, subgroups with different numbers and kinds of linkages emerge. The subset is separated into three groups with various training rates (tr_1, ts_1, vd_1, tr_2, ts_2, vd_2, ..., tr_K, ts_K, vd_K), with tr accounting for 70% of the training, ts for 15% of the test, and vd for 15% of the validation check.

The fuzzy clustering module is composed of the following steps represented by Eqs. (1)–(10):

Step 1: Initialize U^{TR}

$U^{TR} = [u_{ij}^{TR}]$ matrix: $U^{TR}(0)$ and q=1.

Step 2: At q-step: calculate the centers vectors

$C^{TR}(q) = [c_j^{TR}]$ with $U^{TR}(q)$

$$c_j^{TR} = \frac{\sum_{i=1}^{n} U_{ij}^{TRm} \cdot x_j^{TR}}{\sum_{i=1}^{n} U_{ij}^{TRm}} \quad (1)$$

Step 3: Update $U(q + 1)$

$$U_{ij}^{TR} = \frac{1}{\sum_{p=1}^{k} \left(\left\| \frac{x_i^{TR} - c_j^{TR}}{x_i^{TR} - c_p^{TR}} \right\| \right)^{\frac{2}{m-1}}} \quad (2)$$

Step 4: if $\|U^{TR}(q + 1) - U^{TR}(q)\| < \varepsilon$ then

Step 5; in any other case, go back to the second step.

Step 5: Based on argmax (U_{ij}^{TR}), every individual sample of TR can be allocated into subsets TR_K

Following these five steps, the training set TR can be split into k distinct subsets, designated TR_K.

B. The Second Stage

Each ANN model, ANN i, (i = 1, 2, ..., k), is trained using a unique learning algorithm on a unique training set, TR_i, i = 1, 2, ..., k).

Back-propagation-trained, feed-forward neural networks will anticipate intrusions. Here's the algorithm's breakdown.

- Create an ANN with a big enough hidden layer to fit all dataset features, an output layer with as many nodes as output classes, and input layers with as many nodes as input features. The number of hidden nodes was determined by empirical formula $\sqrt{i + O} + \alpha$ ($\alpha = i-10$). "i" is the total number of nodes receiving data, "O" is the total number of nodes sending data, and "α" is a random integer. We restricted our trial to 10 since intrusion detection is tough.
- Randomize weights. We use membership grades based on the number of data sets (n) and clusters (k).

Forward-propagate the input for each training sample through the network.

a. Each hidden node receives the weighted summation of the inputs and bias

$$hid(j) = b_j + \sum_{i=1}^{n} x_i \quad w_{ij} \quad (3)$$

where j is devoted to the hidden unit and i denoted to the data point.

b. This is then passed through a non-linear activation function. A unipolar sigmoid (logsig, output (0, +1)) activation function is used:

$$f(x) = \frac{1}{(1+exp\,(-x))} \quad (4)$$

c. Output of the ANN layers are moved through Tan-Sigmoid transformation function (tansig, output (+1, −1)), instead of Purelin function in FC-ANN algorithm (in MATLAB).

$$a = tansig(n) = \frac{2}{(1+ e^{-2n})} - 1 \quad (5)$$

d. After then, the output produced by the ANN is compared to the desired outcome, and the error is estimated using the error function, which is:

$$Em = \frac{1}{2n} \sum_k \sqrt{(T_k - Y_k)^2} \quad (6)$$

While n denotes the number of training patterns, Y_k and T_k denote the output and goal values, correspondingly.

e. This inaccuracy is then transmitted back through the ANN, and the weights are changed in accordance with the expression:

$$w(t + 1) = w(t) - \eta \partial E(t)/ \partial w(t) \quad (7)$$

t denotes the number of epochs and η refers to the learning rate.

f. The momentum parameter α (0< α <1) is used to accelerate the learning process.

$$w(t+1) = w(t) - \frac{\eta \partial E(t)}{\partial w(t)} + \alpha \Delta w(t) \quad (8)$$

If the error E_m < predetermined threshold then training will be terminated. Else return to Step 3.

Although it is known that the back-propagation technique may be used to successfully train feed-forward neural networks, the challenge then becomes how to merge the findings from several ANN_i base models.

C. Stage Three

Every ANN I has its error reduced by simulating it on the full TR of training data. Once the membership grades have been generated by the fuzzy clustering module, we use them to compile the data. We then utilize this merged dataset to educate a new ANN.

The ultimate result may be generated using the last remaining fuzzy aggregation module. This is how we teach a brand new ANN to spot and rectify inaccurate predictions:

Step 1: Considering that the whole training set T_R as data to be input for every trained ANN_i and get the outputs:

$$Y_j^{TR} = [\, Y_{j1}^{TR}, Y_{j2}^{TR}, ..., Y_{jk}^{TR}], j = 1,2, ..., n \quad (9)$$

n denotes the number of training dataset.: TR, y_{jk}^{TR} is the output of ANN_k.

Step 2: The Formula of the input for new ANN:

$$Y_{input} = [Y_1^{TR}.U_1^{TR}, Y_2^{TR}.U_2^{TR}, \dots, Y_n^{TR}.U_n^{TR}] \quad (10)$$

where U_n^{TR} TRn belongs to CTR at the membership grade

Step 3: train the subsequent ANNs. The new ANN may be trained using Y input as input and the whole TR's class label as output.

IV. Results and Discussion

The KDD CUP 1999 dataset has been used for testing. The KDD CUP 1999 dataset is developed and maintained by the MIT Lincoln Laboratory. It is based on the 1998 DARPA evaluation software for intrusion detection. Approximately 4,898,431 connections make up the KDD collection, this may be obtained from the webpage http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html; the KDDCUP dataset has five distinct kinds of links.

When evaluating an IDS's detection performance, true positives, true negatives, false positives, and false negatives should be analyzed, as advised by Kasongo and Sun [31]. In the context of intrusion detection, a "true positive" refers to an actual attack being detected. To detect an intended condition, an IDS must be "true negative." When an IDS detects a threat that did not really exist, this is known as a false positive. The inability to see a pattern, the limitations of a particular methodology, or external factors all contribute to the occurrence of false positives. It's a test of how well the detecting system works. If the system remains unsecure, administrators will continue to dismiss security alerts. Failure of an intrusion detection system to identify a true attack, this is known as a false negative. Perhaps the intrusion detection system lacks the specifics of the intrusion or the recognition data required to properly label the occurrence. The monitoring system is all-inclusive. These figures are inadequate as a benchmark since the training set contains so few U2R and R2L assaults. Results from performance tests might be skewed if these numbers are used.

Due to its incapacity to learn and its tendency to converge to the local minimum, ANN is inherently unstable. One of the most important aspects of IDS detection accuracy. The proportion of successful training is also measured as a metric since it is indicative of the consistency of detection, which is crucial for ANN-based IDS.

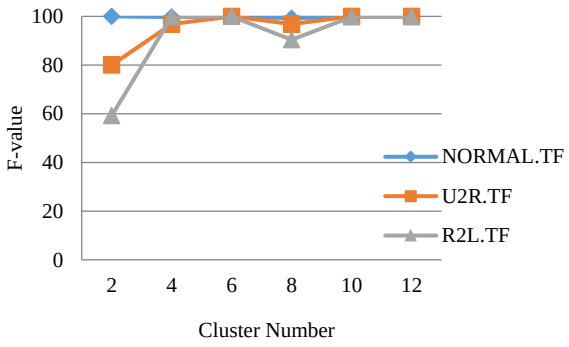

Figure 1. F-value (%) of different clustering numbers use TF.

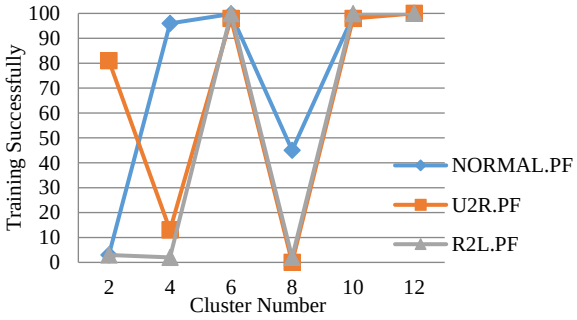

Figure 2. F-value (%) of different clustering numbers use PF.

The artificial neural network (ANN) module and the fuzzy aggregation part both makes use of a stand-alone three-layer network. The prior method of counting hidden nodes relied on an empirical method: (I + O) + (α = 1–10). Thus, the ANN stage neural network structure in the first experiment is notated as [41; 17; 3], whereas in the second experiment, it is notated as [5; 13; 5]. Tansig was utilized at the output node whereas Logsig was employed at the input and hidden nodes as the transformational function. MSE during training was 0.001. Our rate of improvement was 0.1%, and our momentum factor was 0.2. Ten individual tests are performed based on the sample procedures described earlier. We'll also compare these figures to the ones in the [11] publication to show how much more effective the new model is in finding intrusions.

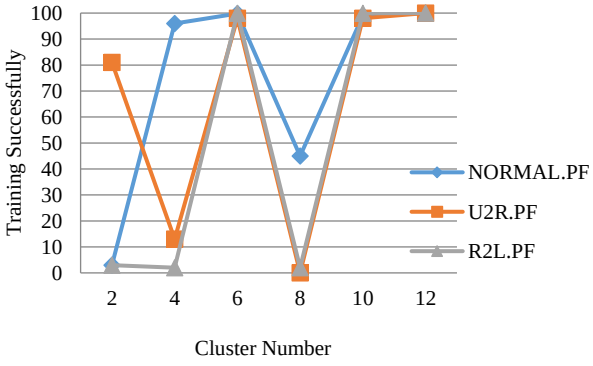

Figure 3. The stability (%) of different clustering numbers use PF.

F-value findings for the updated method (FC-ANN-MD) using Tan-sigmoid function (TF) as a transformation function on the output layer of the ANN structure are shown graphically in Fig. 1 Rates decrease into certain KDD dataset types over a range of cluster sizes. We conclude that the optimal number of clusters at which to calculate the F-value is 6.

The percentage of F-value achieved while using the Wang method (FC-ANN) [2] and the modified version (FC-ANN-MDare depicted in Fig. 1 and Fig. 2 correspondingly. We also get the F-value over a range of cluster sizes, from 2 clusters up to 12 clusters, and it is clear from the cited examples that the F-value rates improve when we switch from a linear to a Tan-sigmoid transformation function in the final layers. Stability findings for Wang's method (FC-ANN) using Purelin Function (PF) as a transformation function on the output layer of the ANN structure are shown graphically in Fig. 3. Rates decrease into certain KDD dataset types over a range of cluster sizes. Better stability results are achieved by dividing the number of clusters into 6 (k=6), as shown by our analysis, which lends credence to the conclusion drawn by Wang's method.

Stability findings for the updated method (FC-ANN-MD) using Tan-sigmoid function (TF) as a transformation function on the output layer of the ANN structure are shown in Fig. 4. Rates decrease into certain KDD dataset types over a range of cluster sizes. We conclude that six clusters provides the best stability outcomes.

The percentages of stability when we use the original Wang method (FC-ANN) and our modified version (FC-ANN-MD) are shown in Fig. 4. We also get the stability over a range of cluster sizes, from 2 clusters up to 12 clusters; the rates of stability improve with the use of the Tan-sigmoid transformation, as seen in the cited figures.

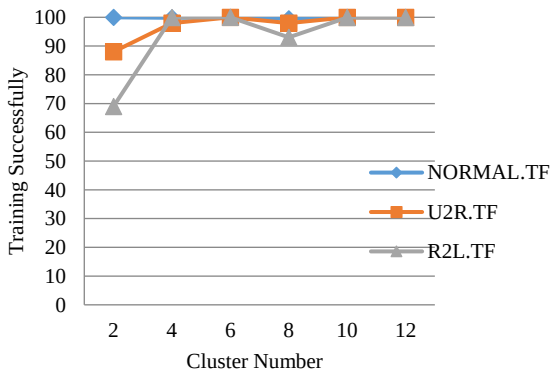

Figure 4. The stability (%) of different clustering numbers use TF.

In this experiment, the subjects are vulnerable to three distinct kinds of attacks: R2L, U2R, and Data norm (Normal class). Low-frequency assaults are represented by the R2l and U2R, whereas high-frequency attacks are represented by the Data norm. The detection accuracy for infrequent assaults is the mean of the accuracy rates for clusters of R2L and U2R attacks of varying sizes (from 2 to 12). Additionally, the detection accuracy for very frequent assaults is an average of the rates of precision in clusters of varying sizes (from 2 to 12) (Normal class).

With findings produced by averaging the assessment criterion rates over a spectrum of cluster sizes (2 to 12) regarding R2L attack, it was discovered that the Precision for Wang's approach (FC-ANN) was 51%, whereas the Precision for the improved FC-ANN-MD was 93.5%. Recall was 50.5% for FC-ANN and 89.9% for Wang's approach, whereas the F-value for the improved algorithm was 91.9%.

The R2L attack rates of Precision, Recall, and F-value vary between the Wang method (FC-ANN) and the modified one (FC-ANN-MD). Again, by averaging the rates of assessment criteria over a variety of cluster sizes (2 to 12), the updated method obtained 97.2% accuracy, 94.2% recall, and 95.5% F-value, compared to 64.9%, 61.2%, and 62.8% for Wang's approach.

Precision, Recall, and F-value rates for Data norm (Normal class) in the Wang technique (FC-ANN) were 73.5%, 77%, and 75%, respectively, compared to 99.8%, 99.9%, and 99.8% for the modified algorithm FC-ANN M.D.

For high-frequency attacks, the FC-ANN M.D. outperformed the Wang approach (FC-ANN) by a factor of 24.8% in accuracy, 26.1% in stability (during training), and 99.7 second less.

V. Conclusion and Future Work

Artificial Neural Network (ANN): ANN is a popular machine-learning technique that has been shown to be efficient in detecting various infections. However, detection precision, especially for less frequent assaults, and detection accuracy for ANN-based IDS still need to be improved. As a consequence, detection accuracy for fewer frequent assaults is reduced. We modified and benchmarked with Wang's ANN structure to improve accuracy and stability for low-frequency attacks, as Gang Wang is the most recent researcher to design an Artificial Neural Networks-based fuzzy clustering method (FC-ANN) aiming to improve the accuracy and stability of intrusion detection systems. First, we alter the output layer's transformation function to Tan-sigmoid from purelin. Second, we employ fuzzy clustering's U matrix as a weight matrix. Artificial Neural Networks using Fuzzy Clustering updated method (FC-ANN-MD). In the first experiment, we train some KDD1999 dataset classes using FC-ANN algorithm and compare the results with those from training the same number of classes using FC-ANN-MD algorithm. There great change in the accuracy and stability rates for low frequent attacks, increasing by 39.4% and 39.3% respectively, and reducing training time by 99.7%. The updated method increases accuracy from 56.7% to 93.5% and stability from 58% to 95.5%. The foregoing findings showed that FC-ANN-MD improves accuracy and stability for low-frequency assaults and provides the optimal training duration. Training the FC-ANN-MD algorithm increases accuracy and stability for high-frequency assaults by 24.8% and 26.1%. The whole KDD1999 dataset will be a participant in the second experiment, and the number of participant vectors for each

class will be limited to 48 vectors. For low-frequency attacks, accuracy and stability are increased by 5.7% and 4%, respectively. For high-frequency assaults, accuracy and stability are reduced by 0.2% for each, and training time is halved. Our technique has enhanced the effectiveness of IDS-based artificial neural networks against low-frequency assaults by improving accuracy, stability, and training time. Our findings indicate that we are making headway toward our primary and secondary research objectives.

It is advised to use this notion to create fair comparisons between the two algorithms (FC-ANN & FC-ANN-MD), since the authors of this study were unable to employ all 18.543 attack items used in the research by Gang Wang.

REFERENCES

[1] V. Aikaterini and V. Vlachos. "Emerging malware threats," *Cybersecurity Issues in Emerging Technologies*, pp. 153–170, 2021.

[2] A. Parihar and R. P. Singh. "Intrusion Detection using data mining," *Artificial Intelligence and Data Mining Approaches in Security Frameworks*, pp. 209–227, 2021.

[3] V. Hajisalem and S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," *Computer Networks*, vol. 136, pp. 37–50, 2018.

[4] M. Turčaník and J. Baráth, "Intrusion detection by artificial neural networks," in *Proc. 2022 New Trends in Signal Processing (NTSP)*, 2022, pp. 1–6.

[5] X. Li, *et al.*, "Detection of low-frequency and multi-stage attacks in industrial internet of things," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8820–8831, 2020.

[6] R. F. Al-Shammari, "New approach for classification R2L and U2R attacks in intrusion detection system," *International Journal of Biology, Pharmacy and Allied Sciences*, vol. 7, no. 4, pp. 1–12, 2018.

[7] S. M. Sangve, "Anomaly based improved network intrusion detection system using clustering techniques," *International Journal of Advanced Research in Computer Science*, pp. 808–815, 2017.

[8] A. Khraisat, *et al.*, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, 2019.

[9] S. Shamshirband, *et al.* "Computational intelligence intrusion detection techniques in mobile cloud computing environments: Review, taxonomy, and open research issues," *Journal of Information Security and Applications*, vol. 55, 102582, 2020.

[10] I. Al-Turaiki and A. Najwa, "A convolutional neural network for improved anomaly-based network intrusion detection," *Big Data*, vol. 9, no. 3, pp. 233–252, 2021.

[11] G. Wang, J. Hao, J. Ma, and L. Huang, "A new approach to intrusion detection using artificial neural networks and fuzzy clustering," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6225–6232, 2010.

[12] C. A. Appasha and A. P. Ghatule, "Cloud intrusion detection system using fuzzy clustering and artificial neural network," *Journal of Physics: Conference Series*, vol. 1478, 012030, 2020.

[13] W. Zhong, *et al.* "Applying big data based deep learning system to intrusion detection," *Big Data Mining and Analytics*, vol. 3, no. 3, pp. 181–195, 2020.

[14] J. Li, Y. Qu, F. Chao, H. P. H. Shum, E. S. L. Ho, and L. Yang, "Machine learning algorithms for network intrusion detection," *AI in Cybersecurity*, pp. 1–27, 2018.

[15] R. Samrin and D. Vasumathi, "Hybrid weighted k-means clustering and artificial neural network for an anomaly-based network intrusion detection system," *Journal of Intelligent Systems*, vol. 27, no. 2, pp. 135–147, 2018.

[16] R. A. Ashfaq, *et al.*, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017.

[17] M. Amini, *et al.*, "A neural network ensemble classifier for effective intrusion detection using fuzzy clustering and radial basis function networks," *International Journal on Artificial Intelligence Tools*, vol. 25, no. 2, 1550033, 2016.

[18] M.Kordos, *et al.*, "Fuzzy clustering decomposition of genetic algorithm-based instance selection for regression problems," *Information Sciences*, vol. 587, pp. 23–40, 2022.

[19] D. Srilatha and G. K. Shyam, "Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network," *Cluster Computing*, vol. 24, no. 3, pp. 2657–2672, 2021.

[20] S. Maleki, S. Maleki, and N. R. Jennings, "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering," *Applied Soft Computing*, vol. 108, 107443, 2021.

[21] S. Manimurugan, *et al.*, "Intrusion detection in networks using crow search optimization algorithm with adaptive neuro-fuzzy inference system," *Microprocessors and Microsystems*, vol. 79, 103261, 2020.

[22] W. M. Tammi, N. A. Biswas, Z. Nasim, K. Z. Shorna, and F. M. Shah, "Artificial neural network based system for intrusion detection using clustering on different feature selection," *International Journal of Computer Applications*, vol. 126, no. 12, pp. 21–28, 2015.

[23] C. Ambikavathi and S. K. Srivatsa, "Integrated intrusion detection approach for cloud computing," *Indian Journal of Science and Technology*, vol. 9, no. 22, 2016.

[24] S. Khazaee and M. S. Rad, "Using fuzzy c-means algorithm for improving intrusion detection performance," in *Proc. 2013 13th Iranian Conference on Fuzzy Systems (IFSC)*, 2013, pp. 1–4.

[25] A. Kathirvel, M. Subramaniam, S. Navaneethan, and C. Sabarinath, "Improved IDR response system for sensor network," *Journal of Web Engineering*, vol. 20, no. 1, pp. 53–88, 2021.

[26] M. Yasir and A. Abdullah, "I^{2}MANET security logical specification framework," *International Arab Journal of Information Technology*, vol. 9, no. 6, Nov. 2012.

[27] R. A. A. Farah and Y. A. Mohamed, "Adaptive immune-based system for network security," in *Proc. 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, 2018.

[28] Y. Abdelgadir and A. Abdullah, "Biologically inspired model for securing hybrid mobile ad hoc networks," in *Proc. 2008 International Symposium on Information Technology*, 2008.

[29] Y. A. Mohamed and A. B. Abdullah, "Securing mobile ad hoc domain by immune-inspired mechanism," in *Proc. 2009 IEEE Region 10 Conference (TENCON 2009)*, 2009, pp. 1–6.

[30] M. A. Ahmed and Y. A. Mohamed, "Enhancing intrusion detection using statistical functions," in *Proc. 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1–6, 2018.

[31] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 1–20, 2020.