# Denial of Service (DoS) Attack Detection Using Feed Forward Neural Network in Cloud Environment

1st Ponugoti Kalpana
*Department of Computer Science and Engineering*
*AVN Institute of Engineering and Technology*
Hyderabad, India
drkalpanacse@gmail.com

2nd P. Srilatha
*Department of AI & DS*
*Chaitanya Bharathi Institute of Technology*
Hyderabad, India
Pulipatisrilatha_aids@cbit.ac.in

3rd Gudepu Sai Krishna
*Department of Information Technology*
*Anurag University*
Hyderabad, India
saikrishnait@anurag.edu.in

4th Ahmad Alkhayyat
*Department of Computers Techniques Engineering*
*College of Technical Engineering*
*The Islamic University*
Najaf, Iraq
ahmedalkhayyat85@gmail.com

5th Debarshi Mazumder
*Department of AI & DS*
*Nitte Meenakshi Institute of Technology*
Bengaluru, India
debarshi.mazumder@nmit.ac.in

2024 International Conference on Data Science and Network Security (ICDSNS) | 979-8-3503-7311-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDSNS62112.2024.10691181

***Abstract*—In the recent years, the Denial of Service (DoS) developed as a big security threats for networks. Preventing as well as identifying DoS attacks is a big task within the industries. There are lot of existing methods are there to identify DoS attacks and minimize the damage. However, many of these methods do not efficiently distinguish noise from a signal. This research focuses on addressing the DoS problem and introduces a method called Feed forward Neural Network (FNN). First, the KDD cup 99 data set goes through a preprocessing step where data normalization is processed using Min-Max normalization which standardizes the data points values. This normalized data is then passed to the whale optimization method which selects the relevant features making the classification system easier. The selected features are then used by an FNN classifier to differentiate between normal and attacked data. The proposed method is implemented and simulated on MATLAB and tested experimentally resulting to a detection accuracy of 0.99. The proposed FNN classifier is compared with the existing classifiers such as Auto-encoder and LSTM where it outperformed in detecting DoS attacks efficiently.**

***Keywords*—*auto-encoder, denial of service, feed forward neural network, min-max normalization, whale optimization.***

## I. Introduction

Technology has transformed a lot in communicate with one another and managing with large industries. It has also changed how people interact online. Its service is also extended and includes applications in the areas of medicine, banking, education, government services, research, entertainment, and defense [1]. Simultaneously, the improvement of networks leads chances for hackers and intruders to discover illicit ways to breach an arrangement. Securing the cloud system is a critical step that involves in solving DoS attacks in the cloud computing architecture because they are destructive in nature [2]. Also, since cloud has a more extensive capability of hosting deep learning models, the large number of networks entails a large amount of data that require handling without any decrease in capability [3]. In cloud environments, the training of deep learning model-based detection methods can easily adapt into the existing environment [4].

These methods could be real-time based, which offer the ability to handle DoS and Distributed Denial of Service (DDoS) incidents automatically [5]. Deep learning also plays a part in the DoS and DDoS attack detection where customers can effectively protect cloud services through a preventive approach. Among various system attacks, DoS attack is an important attack which makes a host server or network resource inaccessible [6]. These models improve the capability to prevent attacks by learning from previous threats detected in a network traffic, and by constantly analyzing the traffic to identify any possible attacks before they occur [7]. Therefore, the deep learning methodologies provide the availability and reliability of the services in the cloud architectures, and defense with emerging threat of DoS and DDoS attacks to the significant infrastructures [8]. The main contributions of the research are:

- The KDD cup 99 dataset is preprocessed using min–max normalization which maps the attribute value into a specified range.
- The relevant features are selected using whale optimization technique which retrieves the optimal set of features.
- The proposed feedforward neural network is used for classifying the normal and attacked packet of information.

The remaining paper is arranged with different sections. Section 2 discusses the literature review, section 3 defines the proposed methodology, results and discussions are discussed in section 4 and section 5 gives the conclusion.

## II. Literature Review

Aktar, S. et al. [9] developed deep learning based model using a contractive autoencoder to detect DoS attacks. Three intrusion detection system datasets such as CIC-IDS2017, NSL-KDD, and CIC-DDoS2019 were used in evaluating the developed method. The attained results showed that the introduced deep learning model outperformed by achieving an accuracy ranging from 93.41% to 97.58%. However, by applying the developed method with other benchmark datasets will result in improving the performance.

979-8-3503-7311-0/24/$31.00 ©2024 IEEE

GSR, E.S. et al. [10] introduced a Fractional Anti Corona Virus Optimization-based Deep Neuro-Fuzzy Network (FACVO-based DNFN) for effective detection of DDoS attacks in the cloud. The combination of the ACVO method and Fractional Calculus (FC) in the FACVO algorithm improved the training process of the DNFN. The FACVO-based DNFN method achieved high testing accuracy and precision in the detection process. However, the introduced method does not support in real-time applications.

Kumar, D. et al. [11] implemented a Long Short-Term Memory (LSTM) based model for detecting DoS attacks in network traffic packets. The developed LSTM method was used for both feature selection and feature extraction which worked efficiently with smaller datasets also. The developed method was evaluated on NSL-KDD dataset and achieved a highest accuracy of 98%. However, by considering some new emerging attacks the performance of the developed method can be improved.

Kshirsagar, D. et al. [12] developed a DDoS detection technique for detecting reflection and exploitation-based DDoS attacks. The developed method combined two feature selection methods such as Information Gain (IG) and Correlation (CR). The developed method was tested on CICDDoS2019 and KDD Cup 1999 datasets and achieved improved detection accuracy. By using this method, the reduction rate was ranging between 56% to 82.92% by maintaining the detection efficiency. However, by using combination of various feature selection methods will improve the performance in detecting various attacks.

Najar, A.A. et al. [13] introduced various machine learning techniques to identify DoS attack packets and their types. Among these, RF established higher performance with an accuracy of 99.13% on both training and validation data and 97% on the full test data and MLP showed an accuracy of 97.96% on training data, 98.53% on validation data, but only 74% on the full test dataset. However, there were problems with overfitting and generalization due to the low performance of MLP method.

## III. Proposed Methodology

The block diagram and working flow of the proposed model is shown in figure 1. The dataset is pre-processed using the Min-Max normalization method which normalizes the data within a specified range. The Whale optimization method is used to select relevant and appropriate features in the next step. The selected features are exposed to the FNN classification model which is used to identify attacked and non-attached information.

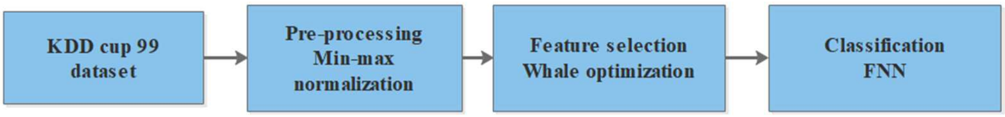
KDD cup 99 dataset
Pre-processing Min-max normalization
Feature selection Whale optimization
Classification FNN

Fig. 1. Block diagram of proposed method

### A. KDD Cup 99 Dataset

The KDDCUP'99 dataset was created by Defense Advanced Research Project Agency (DARPA) [14] in the year 1999 which is a benchmark dataset in the stream of cybersecurity. The dataset contains 4,898,430 samples in which each and every sample contains 41 features which belongs to one of 3 types such as Basic, Content, or Traffic features and divided further into Time or Host-based traffic features. A data point in this dataset is either facing attack or normal where every attack is labeled as DoS U2R, R2L or a Probing Attack. This dataset also has some errors which results in higher training time, less performance and evaluating intrusion detection methods in a poor manner.

### B. Pre-processing

The input dataset is first processed in order to convert the most dominating attributes having higher numerical values into finite series and fed into the normalization technique. The process used here is min–max normalization which maps the attribute value ranges from $[min(q), max(q)]$ to $[new_{min(q)}, new_{max(q)}]$ using the principles of linear transformation given in equation (1).

$$\hat{y}_j = \frac{[y_j - min(q)] \times [new_{max(q)} - new_{min(q)}]}{[max(q) - min(q)]} + new_{min(q)} \quad (1)$$

Here $y_j$ denotes the dataset which is given as input, $min(q), max(q)$ represents the dataset minimum and maximum value. The definite series of minimum and maximum value are given as $new_{min(q)}, new_{max(q)}$ respectively. The normalized output of pre-processing step is denoted as $\hat{y}_j$ used to remove scale difference effect. The specified range is replaced in the tuples which have mismatched values in the dataset. A new dataset is retrieved after pre-processing step from the input dataset for each tuple.

### C. Feature selection using whale optimization

After pre-processing, whale optimization method is used for selecting and retrieving the optimum set of features. The classification method's performance is improved by using optimization method for feature selection which optimally choses the best feature sets. The whale optimization imitates the basic idea behind whale hunting behavior. According to this algorithm, whales begin their hunt by using their three primary functions: searching for prey, encircling it, and creating a bubble net to perform the hunting process. First, a random initialization is applied to the whale positions according to the set of feature vectors derived from the pre-processed output. On the search space, the population's starting number of whales is randomly determined. After obtaining the highest accuracy function, the fitness function is measured to verify that the population contains the ideal set of attributes. Next, the best features that were acquired are updated in terms of the positions of whales. Finally, the best whale is chosen as the feature that provides the most information. Lastly, the optimization method ends when entire features have been searched. The initial process is given in equation (2)

$$I_j = (\hat{y}_1, \hat{y}_2, \hat{y}_3 \cdots \hat{y}_j) \quad (2)$$

Here the set of 'j' number of feature set is indicated by $\hat{y}_1, \hat{y}_2, \hat{y}_3 \cdots \hat{y}_j$ . The original set of features which are normalized and obtained after pre-processing are denoted as $I_j$. Those normalized features are fed into the fitness function for choosing the optimal set of feature vectors. The best characteristics are picked using an optimization technique in order to identify DoS attacks in an effective manner.

### D. *Classification using FNN*

A feedforward neural network (FNN) [15] is a type of multi-layer neural network among the prevalent ones. It consists of three types of layers: Between these, the input layer is the first layer that receives raw inputs from the users while the output layer produces the final result of the analyses made on the inputs by the hidden layer. Input layer provides the network with the input variables while the output layer is the terminal or last layer that displays the network results. The intermediate layers of the network are denoted as the hidden layers because they are not directly involved in the input or output process. The hidden layer consists of several neurons which help in achieving the goal. The flow of information occurs from the input, through the hidden, to the output layer. Since the linear function of a neural network cannot model any function, an activation function is incorporated into the network. One of the most commonly used activation functions is the Rectified Linear Unit (ReLU) and defined as equation (3).

$$ReLU = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{3}$$

Where x represents the input variable. In FNN training process, the difference is frequently occurred between estimated and true output of the network. To measure these errors, a loss function is used in order to weigh these errors, a loss function is used. For multi-class classification problems, the cross-entropy loss function is commonly used and since the total probability of any outcome is equal to 1, the likelihood function is maximized using equation (4).

$$l(\Theta) = -\frac{1}{n}\sum_{i=1}^{n} \sum_{j=1}^{q} y_j^{(i)} \ln \hat{y}_j^{(i)} \tag{4}$$

Where $y_j^{(i)}$ denotes the predicted label and $\hat{y}_j^{(i)}$ denotes the actual label. FNN in turn are effective mathematical models, used for modeling the dependence of the dependent variable on independent variables. This is established once the training is carried out by the use of training data that is well labeled. In the course of this training process, the weights of the network are required to be changed so that the network gives more accurate output than the actual output. Therefore, the previous relationships connecting the input and the output variables in its model are ingrained in the network. These weights are then used to fine-tune the local network on the training samples so that the network is capable of predicting outputs for new and unseen training data within a given class.

## IV. Results And Discussions

The implementation is done on the MATLAB working platform, which is equipped with a 3.10 GHz Intel (R) Core (TM) i5-3570S CPU processor. Experiments are performed on a benchmark dataset naming KDD CUP 99 database. The main purpose of this section is (I) Using Accuracy, Precision, Recall and F-measure metrics to analyze the performance analysis of proposed method, (II) Comparing other existing techniques with the proposed method based. The proposed method is evaluated by using the below metrics which are given in equation (5) to (8).

$$Accuracy = \frac{(TP+TN)}{(TP+FP+FN+TN)} \tag{5}$$

$$P = \frac{TP}{TP+FP} \tag{6}$$

$$R = \frac{TP}{TP+FN} \tag{7}$$

$$F = \frac{2PR}{P+R} \tag{8}$$

where, TP means True Positive, FP signifies False Positive, TN represents True Negative, and FN means False Negative.

### A. *Performance analysis*

The proposed FNN method is compared with other classification methods such as K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) to prove the efficiency. The proposed system achieves the extreme Accuracy, Precision, Recall and F-measure when compared with other approaches as shown in Table 1.

TABLE I. Performance Analysis of Other Classifiers With Proposed Method

| Method | Accuracy | Precision | Recall | F-measure |
|---|---|---|---|---|
| KNN | 0.9283 | 0.9126 | 0.9014 | 0.9438 |
| SVM | 0.9527 | 0.9802 | 0.9650 | 0.9710 |
| FNN | 0.9962 | 0.9928 | 0.9847 | 0.9853 |

### B. *Comparative analysis*

The proposed method is compared with existing method namely Contractive Auto-encoder [9] and LSTM [11]. The performance of the proposed method is evaluated in terms of accuracy, precision, recall and f-measure as given in table 2.

TABLE II. Comparative Analysis of proposed FNN method with existing methods

| Classifier | Accuracy | Precision | Recall | F-measure |
|---|---|---|---|---|
| Auto-encoder [9] | 0.9608 | 0.9610 | 0.9608 | 0.9608 |
| LSTM [11] | 0.9820 | 0.9841 | 0.9715 | 0.9702 |
| FNN (proposed) | 0.9962 | 0.9928 | 0.9847 | 0.9853 |

The method Auto-encoder [9] by using with other benchmark datasets will result in improving the performance and LSTM [11] by considering several new developing attacks the performance can be improved. To overcome these limitations the proposed FNN method is used and achieved 0.99 of accuracy. Unlike the existing methods, the FNN is used to effectively identify the DoS attacks. The proposed method has achieved 0.99 of accuracy in identifying DoS attacks in cloud computing environment.

### C. *Discussion*

The FNN algorithm is proposed in this research which is used to identify all the DoS attacks which occur in a cloud environment. The proposed method is compared with the existing methods such as Contractive Auto-encoder [9] and LSTM [11] to know the performance. The method Auto-encoder [9] by using with other benchmark datasets will result in improving the performance and LSTM [11] by considering several new developing attacks the performance can be improved. The FNN algorithm is proposed to overcome all the limitations of the existing methods. The proposed method achieved 0.9962 of accuracy in identifying the DoS attacks in cloud environment.

## V. Conclusion

The proposed methodology aims in identifying DoS attacks in a cloud system by using pre-processing of data, feature extraction, and classification stages. The first step of the research is pre-processing in which all input data values are scaled to the interval of specified range using the min-max normalization technique. After normalization, the retrieved data is applied to whale optimization algorithm for the selection of features from the best subset. After the features are chosen, the proposed FNN classifier that differentiate between the attacked and non-attacked data. Also, the proposed detection model is intended to be used to mitigate the occurrence of DDoS attacks in bigger industries. The proposed model learns to detect DoS attacks with an overall accuracy of 0.99 and is more efficient compared to other related approaches. The further research focuses on enhanced optimization method with deep learning for identifying the attack in several other applications.

## References

[1] R. Priyadarshini, and R. K. Barik, “A deep learning based intelligent framework to mitigate DDoS attack in fog environment,” J. King Saud Univ. Comput. Inf. Sci, vol. 34, no. 3, pp. 825–831, 2022.

[2] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, “Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques,” Computers & Electrical Engineering, vol. 98, p. 107716, 2022.

[3] J. F. C. Garcia, and G. E. T. Blandon, “A deep learning-based intrusion detection and preventation system for detecting and preventing denial-of-service attacks,” IEEE Access, vol. 10, pp. 83043–83060, 2022.

[4] Berbineau, M., Sabra, A., Deniau, V., Gransart, C., Torrego, R., Arriola, A., Val, I., Soler, J., Yan, Y., Vizzarri, A. and Mazzenga, F., 2021, April. Zero on site testing of railway wireless systems: the Emulradio4Rail platforms. In 2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring) (pp. 1-5). IEEE.

[5] H. Aydın, Z. Orman, and M. A. Aydın, “A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment,” Comput. Secur, vol. 118, p. 102725, 2022.

[6] D. M. B. Lent, M. P. Novaes, L. F. Carvalho, J. Lloret, J. J. Rodrigues, and M. L. Proença, “A gated recurrent unit deep learning model to detect and mitigate distributed denial of service and portscan attacks,” IEEE Access, vol. 10, pp. 73229–73242, 2022.

[7] A. Ahmim, F. Maazouzi, M. Ahmim, S. Namane, and I. B. Dhaou, “Distributed denial of service attack detection for the Internet of Things using hybrid deep learning model,” IEEE Access, vol. 11, pp. 119862–119875, 2023.

[8] Kumar, K.N., Mohan, C.K. and Cenkeramaddi, L.R., 2023. The Impact of Adversarial Attacks on Federated Learning: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[9] S. Aktar, and A. Y. Nur, “Towards DDoS attack detection using deep learning approach,” Comput. Secur, vol. 129, p. 103251, 2023.

[10] E. S. GSR, R. Ganeshan, I. D. J. Jingle, and J. P. Ananth, “FACVO-DNFN: Deep learning-based feature fusion and Distributed Denial of Service attack detection in cloud computing,” Knowledge-Based Syst, vol. 261, p. 110132, 2023.

[11] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, “DDoS detection using deep learning,” Procedia Comput. Sci, vol. 218, pp. 2420–2429, 2023.

[12] D. Kshirsagar, and S. Kumar, “A feature reduction based reflected and exploited DDoS attacks detection system,” J. Ambient Intell. Hum. Comput, vol. 13, no. 1, pp. 393–405, 2022.

[13] A. A. Najar, and S. M. Naik, “DDoS attack detection using MLP and Random Forest Algorithms,” Int. J. Inf. Technol, vol. 14, no. 5, pp. 2317–2327, 2022.

[14] Parasuraman, S., Yogeeswaran, S. and Ramesh, G.P., 2020, September. Design of Microstrip Patch Antenna with improved characteristics and its performance at 5.1 GHz for Wireless Applications. In IOP Conference Series: Materials Science and Engineering (Vol. 925, No. 1, p. 012005). IOP Publishing.

[15] S. Avinash, H. N. Kumar, M. S. G. Prasad, R. M. Naik, and G. Parveen, “Early detection of malignant tumor in lungs using feed-forward neural network and k-nearest neighbor classifier,” SN Comput. Sci, vol. 4, no. 2, p. 195, 2023.