The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 2, April 2025
https://doi.org/10.33022/ijcs.v14i2.4538

Detecting Distributed Denial of Service Attacks in Mobile Edge Computing using modified extreme machine learning

Sekgoari Semaka Mapunya[1], Mthulisi Velempini [2]
sekgoari.mapunya@ul.ac.za[1], mthulisi.velempini@ul.ac.za [2]
[1,2] Department of Computer Science, University of Limpopo, Sovenga, South Africa

Article Information

Received : 10 Dec 2024
Revised : 21 Jan 2025
Accepted : 26 Mar 2025

Keywords

Mobile Edge Computing, Distributed Denial of Service (DDoS), Firefly Algorithm, Extreme Learning Machine, Neighbourhood-Based Differential Evolution.

Abstract

Mobile Edge Computing (MEC) is a promising technology which enables 5G and reduces latency. By bringing cloud computing capabilities closer to end users, MEC enables latency-sensitive applications to perform more efficiently. However, security attacks pose significant challenges to the objectives of 5G with Distributed Denial of Service (DDoS) attacks being a major threat. These attacks can overwhelm target systems with excessive data preventing access to and disrupting network services. Effective mitigation strategies are required to protect MEC technology. Given the high data volume generated by such attacks, this paper utilizes a modified Firefly Algorithm to select relevant features. These selected features are then used to train a proposed variant of Extreme Learning Machine (ELM), where weights are initialized using Neighbourhood-Based Differential Evolution. MATLAB simulations demonstrate that the proposed modified ELM outperforms traditional approaches, providing an effective solution to DDoS attacks in MEC.

A. Introduction

Communication networks are increasingly becoming vulnerable to security threats prompting a need for robust security mechanisms [1], [2]. Mobile Edge Computing (MEC), an emerging communication technology is susceptible to security attacks [3]. Unlike traditional networks, MEC brings cloud computing resources closer to end-users to support latency-sensitive applications. However, this proximity introduces unique security challenges which cannot be mitigated by current security schemes.

Among the primary threats to MEC is the Distributed Denial of Service (DDoS) attack. DDoS attacks can lead to significant latency and system unavailability by overwhelming the network with excessive data requests which prevent legitimate users from accessing networks resources. Several approaches have been proposed to counter DDoS attacks in MEC. The authors in [4] proposed signature based scheme while in [5], a statistical based approached is proposed. In [6] and [7], schemes integrating signature and statistical approaches are discussed. As MEC continues to evolve, developing new specialized security measures is critical to safeguarding its performance and availability.

In this work, we proposed a statistical approach named Neighbourhood based deferential evolution extreme machine learning (NDE-EML) to address the DDoS attacks in MEC. The performance of model is compared to the traditional EML. Unfortunately, when addressing DDoS attacks, lot of data is generated. Performing statistics analysis on the entire data set leads to delays in detecting attacks [8]. To address this challenge, we first propose a feature selection technique known as Quasi opposite Firefly Algorithm (QOFA) to reduce the number of features resulting in early detection of attacks and accurate predictions. NDE-EML is a modified extreme machine learning model which makes use of Neighbourhood based differential evolution to assign input-hidden link weights and hidden biases. On the QOFA, we modified the firefly algorithm by initialising the population by using quisi-opposite based learning differential evolution.

The paper is organized as follows. Section 2 examines the various developments in DDoS attacks mitigation. In Section 3, we present the dataset and feature selection process for the model. Section 4 presents the development of the proposed model including training, testing, and performance evaluation. Section 4 presents the experiment results and comparative evaluation results. Section 7 concludes the study.

B. Related work

DDOS attacks were outlined and discussed as potential attacks on edge computing in [9]. The prevention of these attacks take place in the form of Statistical or packet monitoring. Our proposed work adopt a statistical approach where machine learning techniques are utilized.

The work in [10] proposed a MECshield where smart filters were deployed at the destination network. The filters cooperate using policies generated by the central controller and the policies are triggered by a given type of attack. Self-organizing map (SOM) in [11] is part of the component of the filters and they are trained simultaneously using local traffic supervised by the generated policies. The

trained SOM detects malicious IoT traffic by matching the traffic features to the SOM map to detect whether it represents a DDoS attack. The detection rate and accuracy in mitigating the attack were improved. Numerical results show that the proposed scheme outperformed distributed-SOM and centralized-SOM schemes proposed in [12]. This approach leads to a delay in the detection of the attack if large data sets are used.

The novel autonomic security system is proposed in [13] which protects 5G networks from DDoS attacks. The system is self-managing and implements all detection protocols in the detection and mitigation of the effects of the attack. Autonomous decisions are made and enforced automatically by the proposed scheme. The paper focused on combating User datagram flooding attacks. We focus on MEC, the technology that was proposed to enable the objectives of 5G. The proposed scheme is vulnerable to DDoS attacks.

The work in [14], proposed a detection framework, a composite multiplayer perceptron designed to detect any type of DDoS attack. The effectiveness of this framework was determined through simulation, and it achieved an accuracy level of 99.66%. The is a high level of accuracy which was achieved at the cost of high computational power. There is therefore a need for a lightweight, simple, and efficient scheme which requires less computational power. Our work uses feature selection in pre-processing phase which reduces the effort needed to classify the attack. This is desirable since the MEC has lover computational power compared to traditional cloud computing architectures.

The Cooperative defence framework for MEC known as CODE4MEC is proposed in [15] to reduce the costs associated with the implementation of security defence mechanisms to each MEC server. We modify traffic flow by automatically coordinating the network defence mechanisms amongst cooperative edge servers. Four control plane mechanisms that enable the functionality of this framework were proposed. These are CODE triggering, scheduling, coordination and releasing mechanisms. Hence, APP-DDOS attack mitigation scheme was applied on top of this framework and its effectiveness was validated and was evaluated using a testbed and simulation. This work is adopted in our study. We designed a mitigation scheme which enhances the functionality of this framework.

In [16], the authors proposed a traffic scheduling strategy to counter DDoS attacks in edge computing-enabled Time and Wavelength Division Multiplexed Passive Optical Networks (TWDM-PON). Application-level DDoS attacks were the focal of their study. The strategy schedules time sensitive services where Edge Computing Optical Network Units (EC-ONUs) are attacked with the central premise of reducing the impact of the attack on the services hosted at EC nodes.

Two strategies are proposed in this paper where the second one addresses the shortcomings observed in the first strategy. In the first strategy, EC-ONUs when attacked by DDoS, can still allow delay sensitive applications to be executed in the event were the node is still able to meet the requirements of the network in the presence of an attack. Hence, if a request from delay sensitive application comes with a high demand for the services and the node is already overwhelmed, the request can be transferred to other nodes. The sharing of EC nodes is allowed by the scenario where the is a cooperative communication among the nodes. Unfortunately, delay

sensitive application traffic at the attacked nodes can be subjected to delay in the event that they are enqueued.

In addressing the chalMlenge above the second strategy was proposed. The service request by legitimate nodes can be allowed to utilise the edge computing nodes dedicated to service delay tolerant applications. The proposed strategies do not provide solutions to a scenario where the is a coordinated attack targeting all EC nodes. Also, the scenario where other layer-level attacks are launched they may not be addressed. It assumes that there is always one EC node that is attacked.

In work [17], authors proposed the combination of source based and reactive prevention mechanisms to address DDoS attacks on Multy Access Edge Computing (MAEC). The mechanisms are designed to combat the attack at both the source and destination.

In paper [18], authors proposed the DDoS attack mitigation Scheme that utilised the visual environment and management entities of MEC. The attack is mitigated through the use of hybrid approaches which integrates anomaly and deep parked inspection techniques. Using artificial intelligence, the network is monitored continuously to detect any anomalous behaviour. The suspected network traffic is routed to a visual machine where deep inspection of the data packets is done. This reduces the load of the targeted node so that it continues to serve other requests. The management entities of MEC, reconfigure the nodes after data is routed to visual machines. The paper is an extension of the work in[19], and the authors did not provide any analytical or simulation results, hence the is no conclusion about the effectiveness of the proposed scheme.

In [20], an anomaly-based DDoS attack detection mechanism in Fog computing is proposed. The mechanism utilises the Naive Bayesian Classifier combined with the Markov Model and Virtual Honey Pot Devices. The model reduces false positive rates. Using the source and destination data, the probability of attack and normal is calculated using the Naive Bayesian Classifier. Packets are moved to the Markov Model for further inspection in the event when the probability of attack is greater than that of normal traffic. The Threshold Value (TV) is used to compare the probability calculated using the Markov model. Therefore, if the probability is greater than TV then the packet is transferred to Visual Honey Pot Device (VHD).

The log file generated by the VHD records the device information if it is malicious or not. Hence, the non-malicious data is transferred back to the fog network. The proposed scheme does not address the concept of queuing of the parkers which can lead to Quality-of-Service degradation and increased latency. The DDoS attack impact the availability of MEC and its resources. Delay-sensitive applications can be starved of resources, resulting in catastrophic events like car accidents in vehicular networks. The availability assures that the resources are available to users when robust detection and prevention schemes are implemented to detect and address the effects of malicious users.

In [21], DDoS attacks are outlined and discussed as potential attacks on edge computing. The prevention of these attacks is presented in the form of Statistical or packet monitoring techniques.

There are still some open research issues and gaps in the DDoS attack mitigation in MEC. To the best of our knowledge, much has not been done to address the multy-layered DDoS attack in MEC. Multi - layer DDoS attacks may be mitigated by

employing a hybrid attack detection strategies that combine signature-based and anomaly detection.

C. Proposed Scheme, Synthetic Data, and Simulation Results

i. Traditional extreme machine learning

Artificial neural networks (ANN) are "an interconnected of nodes inspired by biological neurons" [22]. Extreme machine learning (EML) is one of the ANN models proposed to train single-hidden layer feedforward neural networks (SLFN) [23]. Modified versions of EML are proposed in the literature [24], [25], [26], [27]. To prevent the machine learning algorithm from being subjected to a lengthy training period, we can reduce the high dimensionality of the dataset with the aid of the feature selection strategy that was proposed in the previous section.

In the feedforward networks, they are three basic layers of neurons: input, hidden and output layers. The neurons from the previous layer are connected to those of the next layer through weighted links. In the training procedure, input-hidden weights and hidden biases are randomly assigned a value in the range [0,1], while hidden-output weights are determined by Moore–Penrose inverse. This contributes to faster model training when compared to back-propagation which uses iteration after randomly assigning weights in the range [0,1]. Figure 1, presents the layout of extreme machine learning.

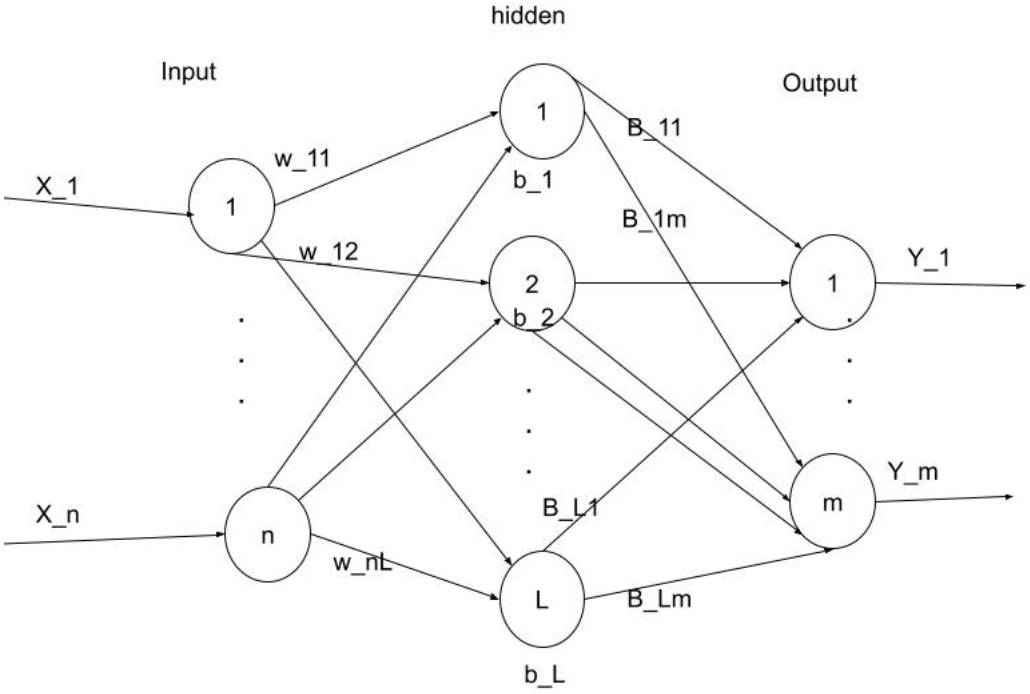

Figure 1. Extreme machine learning

Fig 1 presents EML with n input, L hidden and m output neurons. The input-hidden link weight ω_{ij} connects i^{th} input to j^{th} hidden neurons were i = 1 to n, and j = 1 to L. Link with weight β_{ij}, on the other hand, connects the i^{th} hidden neuron to the j^{th} output neuron where i = 1 to L and j = 1 to m, hidden layer biases are expressed as b_i. Matrices ζ and φ represent the set of all input-hidden link weights

and hidden-output link weights, respectively. $x_i = [x_{i2}, x_{i2}, x_{i3}, \dots, x_{in}]^{\mathrm{T}} \in R^n$ where n is the number of features, represents the i^{th} sample. $y_i = [y_{i1}, y_{i2}, y_{i3}, \dots, y_{im}]^{\mathrm{T}} \in R^m$ where m denotes the number of classes that the sample could belong to are the corresponding output of the model. The model is first trained before it can be used to classify data.

Suppose there are a training samples of the form $x_i^t = [\, x_{i1}^t, x_{i2}^t, x_{i3}^t, \dots, x_{in}^t]^{\mathrm{T}} \in R^n$, with their corresponding target vector of the form $\alpha_i = [\alpha_{i1}, \alpha_{i2}, \alpha_{i3}, \dots, \alpha_{im}]^{\mathrm{T}} \in R^m$. The output of this network can then be modeled as follows:

$$y_i = \sum_{j=1}^{L} f(w_j\,, b_j, x_i^t)\beta_j\,, \qquad for\ \ i = 1,2,3, \dots, a \tag{1}$$

Where, $w_j = [w_{j1}, w_{j2}, w_{j3}, \dots, w_{jn}]^{\mathrm{T}}$ and $\beta_j = [\beta_{j1}, \beta_{j2}, \beta_{j3}, \dots, \beta_{jm}]^{\mathrm{T}}$ are the weight vectors that connect the j^{th} hidden neuron to the n input neuron and the m output neuron, respectively. b_j represents the j^{th} hidden bias. For the entire training dataset of a samples, we may express the following equation:

$$y = \phi\varphi \tag{2}$$

Where,

$$\phi = \begin{pmatrix} f(w_1, b_1, x_1^t) & \cdots & f(w_L, b_L, x_1^t) \\ \vdots & \vdots & \vdots \\ f(w_1, b_1, x_a^t) & \cdots & f(w_L, b_L, x_a^t) \end{pmatrix}, \varphi = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{pmatrix}, y = \begin{pmatrix} y_1^T \\ \vdots \\ y_a^T \end{pmatrix}$$

in reducing error $|y - \alpha|$, where $\alpha = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_a]$ representing target vectors of all the training samples. We can randomly generate values for the parameters w_j and b_j independent of the input values. The hidden-output link weights can now be calculated by finding the least square solution to the equation below.

$$\varphi = \phi^{\urcorner}\alpha \tag{3}$$

where $\urcorner$ represents the generalized inverse matrix of Moore–Penrose. Now that the value of the hidden-output link weights matrix (φ) is known, the model can be used for classification.

ii. Neighbourhood Based Differential Evolution - Extreme Learning (NDE-EML).

The fact that it does not require iterative training, EML can be trained faster. However, because the input-hidden link weights and hidden biases are randomly initialized, it may not always produce an optimal training accuracy resulting in low testing accuracy. Furthermore, the number of neurons in the hidden layer must be greater than that of other SLFN resulting in a longer testing time for unknown samples.

In addressing the above-mentioned challenge this work proposes the optimization of EML. Neighbourhood-based differential evolution (NBD) [28] is utilised to optimize EML. The input-hidden layer weights and hidden biases are found by utilizing NBD Whereas Moore–Penrose is utilised to calculate the hidden-output weight. Neighbourhood Based Differential Evolution is a population-based algorithm whereby each unit of the population represents a solution. Until the optimal solution is found those units evolve in each successive generation. In the proposed scheme, the input-hidden link weights and hidden biases are encoded as a vector.

In achieving optimal training accuracy of NDE-EML the following steps are followed from generation to generation until the optimal training accuracy is reached: Input-hidden layer link initialisation, hidden-output layer link weights calculation, mutation crossover and selection.

Input-hidden layer link Initialization

An encoded solution vector is initialised during every generation composed of a population of size ⁔.The target vector refers to how each individual in the population is represented.

$\Gamma_{k,s}$=[

$w_{11}, w_{12}, \dots, w_{1L}, w_{21}, w_{22}, \dots, w_{2L}, w_{31}, w_{32}, \dots, w_{3L}, \dots, w_{n1}, w_{n2}, \dots, w_{nL}, b_1, b_2, \dots, b_L$]

Where k and s represent the target vector and generation number respectively. Target vectors $\Gamma_{k,1}$ are randomly initialized with values that fall within [0,1]. In other generations (s), the population is initialised by a selected solution vector from the previous generation (s-1).

hidden-output layer link weights calculation

s^{th} population target vector is calculated as:

$$\varphi_{k,s} = \phi_{k,s}{}^{\intercal}\alpha \tag{4}$$

Where $\varphi_{k,s}$ and α represent hidden layer output Matrix and target matrices respectively. The said matrices are defined as:

$$\phi_{k,s} = \begin{pmatrix} f(w_{1,s}, b_{1,s}, x_{1,s}) & \cdots & f(w_{k,s}, b_{k,s}, x_{1,s}) \\ \vdots & \vdots & \vdots \\ f(w_{1,s}, b_{1,s}, x_{k,s}) & \cdots & f(w_{k,s}, b_{k,s}, x_{k,s}) \end{pmatrix}, \ \alpha = \begin{pmatrix} \alpha_1^T \\ \vdots \\ \alpha_a^T \end{pmatrix}$$

Where, $\alpha_s = [\alpha_{s,1}, \alpha_{s,2}, \dots, \alpha_{s,a}]$ is the target vector of s[th] training sample and a is the total number of training sample. Mean squared error (MSE) is defined as

$$\text{MSE}_{k,s} = \frac{\sum_{a=1}^{n}\sum_{u=1}^{L}(\beta_{u,s} f(w_{u,s}, b_{u,s}, x_{a,s}) - \alpha_{a,s})}{n}$$

is utilised as an objective function. The vector with minimum RMSE in generation S is represented as $\Gamma_{best,s}$.

mutation crossover

The process of changing one or more components of the solution vector to produce a better solution is known as mutation. In this stage, a mutant vector $\kappa_{k,s}$ is produced for each target vector $\Gamma_{k,s}$. This study makes use of Differential evolution global and local neighbourhood (DEGL) mutation strategy proposed in [28]. To maintain the uniqueness of every neighbourhood, the vector indices are arranged randomly (as determined during initialization). Now, we establish a neighbourhood of radius z for each vector $w_{n,L}$, where z is a nonzero integer from 0 to (n-1)/2 since the neighbourhood size must be smaller than the population size, i.e. 2z + 1 ≤ n, made up of vectors $w_{n-z,L}$,..., $w_{n,L}$,..., $w_{n+z,L}$. In working with this method, we assume that the vectors are organised in a ring topology such that the two direct neighbours of $w_{1,L}$ are $w_{n,L}$ and $w_{2,L}$.

The best (fittest) vector from the member's neighbourhood and any two more vectors are used to build a local donor vector for each individual in the population. The model might be written as

$$U_{n,L} = w_{n,L} + \alpha \,.\, (w_{n\_bestn,L} - w_{n,L}) + \beta \,.\, (w_{p,L} - w_{q,L}) \quad (5)$$

Where the best vector in the neighbourhood of $w_{n,L}$ is indicated by subscript n_bestn and $p, q \in [i - k, i + k]$ with $p \neq q \neq n$. likewise, the global donor vector is created as

$$g_{n,L} = w_{n,L} + \alpha \,.\, (w_{g\_bestn,L} - w_{n,L}) + \beta \,.\, (w_{r1,L} - w_{r2,L}) \quad (6)$$

The best vector in the entire population at generation L is denoted by g_bestn and $r1, r2 \in [1, n]$ with $r1 \neq r2 \neq i$. α and β are the scaling factors.

The first perturbation term of equation (5) and (6) (the one multiplied by α) represents an arithmetical recombination operation, whereas the second term (the one multiplied by β) represents a differential mutation. As a result, we produce altered recombinants rather than pure mutants in both the global and local mutation models.

Now, to create the actual donor vector for the DEGL, both local and global donor vectors are merged using a scalar weight ¢ ∈ (0, 1).

$$V_{n,L} = ¢ \cdot g_{n,L} + (1\text{-}¢)\, U_{n,L} \quad (7)$$

If ¢ =1 and α = β = F, then the equation (7) becomes a donor vector generated by DE/target-to-best/1 technique. After the mutation, once the donor vector has been produced, the crossover phase is conducted. The objective and donor vectors are combined by this operator. The Binomial and Exponential crossovers are the two basic types of crossover operators. Binomial crossover is employed in DEGL. Based on the crossover probability, Binomial Crossover chooses the value for each gene in the final vector from one or the other ancestor.

Selection

The population vector of the next generation is selected at this stage. MSE is used for selecting the target vectors for the next generation. The calculated values of MSE for each vector of interest at the current generation are compared and the one with the minimum value of MSE is selected and included in the next generation.

iii. Simulation environment

In this section, we discuss the simulation environment set up for the implementation of the NDE-EML and NDE subjected to QOFA. We utilised synthetic data generated in MATLAB. The most relevant features were selected using QOFA. Then the selected data was used to train and test NDE-EML and NDE. Different performance evaluation metrics were evaluated.

Firstly, we generated synthetic datasets to simulate both normal traffic and DDoS attack patterns. DDoS attacks are dynamic, hence there is a need to develop up-to-date mitigation schemes. The available datasets used in the literature are outdated and have limitations [29]. if used there is a possibility that mutated attack strategies may be missed [30]. Hence, the synthetic data was generated for this reason. We began by defining feature distributions for both types of traffic. For normal traffic, the mean values were specified in the normal_dist array ranging from 10 to 300 in increments of 10. For DDoS attack traffic, the mean values were specified in the ddos_dist array ranging from 50 to 340 in increments of 10. A standard deviation of 10 was used for both distributions to introduce variability.

The total number of data points was 100,000, with a proportion of 0.2, 0.4, 0.6, 0.8 allocated to DDoS attack traffic. Consequently, 80000, 60000, 40000 and 20000 data points represented normal traffic respectively. For each of the 30 features, normal traffic data was generated using a normal distribution with the specified means and standard deviation, resulting in a dataset with dimensions normal data x 30 respectively. Similarly, DDoS attack data was generated using the specified means and standard deviation, producing a dataset with dimensions DDoS attack data x 30.

These two datasets were then combined into a single matrix, data containing both normal and DDoS traffic. Corresponding labels were created to form the `labels` vector, where `0` denotes normal traffic and `1` denotes DDoS attacks. This synthetic dataset provided a controlled environment for evaluating the performance of the traditional Firefly Algorithm and Quasi-Oppositional Firefly Algorithm in selecting

relevant features for network traffic classification. The generated data ensured a balanced representation of normal and attack traffic, which is crucial for reliable performance evaluation of feature selection and classification algorithms.
Secondly, the core of the environment consists of machine learning techniques. We employed QOFA to optimize the selection of features that are most relevant for distinguishing between normal and attack traffic. The Traditional Firefly Algorithm evaluates feature attractiveness between fireflies, while the variant with Quasi-Oppositional Learning enhances diversity by incorporating quasi-opposite solutions.

Thirdly, the evaluation metrics used in this environment encompass a wide range of performance indicators crucial for assessing the effectiveness of feature selection and classification models. These metrics include accuracy, precision, recall, F1 score and false positive matrices. Each metric provides insights into different aspects of model performance, such as its ability to correctly classify instances of normal traffic and DDoS attacks.

Fourthly, we partitioned the data into training and testing to ensure robust performance evaluation. The 70% of the data is used for training the models, while the remaining 30% is used for testing. This partitioning strategy helps in assessing how well the models generalize unseen data and avoid overfitting. Figure 2 is the flow diagram of the proposed and traditional EML.
Synthetic dataset is generated comprising of normal and DDoS attack data.
Using QOFA, feature selection is done on the dataset to minimize its dimensionality to the most pertinent features.
The reduced dataset is then partitioned into training and testing.
EML and NDE-EML is trained and tested
Finaly, the results of the models are extracted.

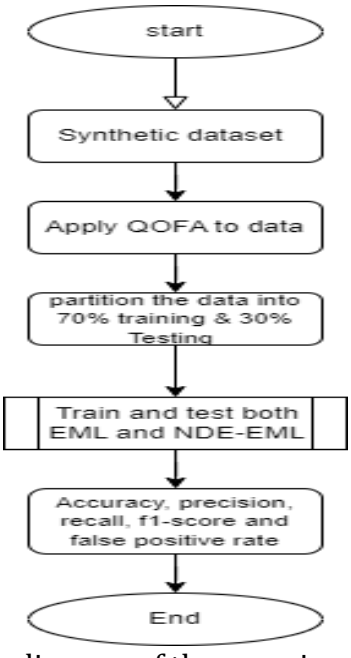

Figure 2. Flow diagram of the experiment carried out.

The simulations were conducted using MATLAB Online due to constraints in computational power on local systems. MATLAB Online provides the necessary computational resources and a cloud-based environment, ensuring that the simulations run efficiently without the limitations posed by local hardware. This platform allows for seamless collaboration, easy access to the latest MATLAB features, and the ability to run intensive computations without overloading the local machines.

iv. Simulation Results

In this section, we present and discuss the performance results of EML compared to NDE-EML subjected to QOFA obtained through MATLAB experiments. Figures 3-6 present the simulation results of EML compared to NDE-EML under different network scenarios. Comparison results of the investigated algorithm in a network scenario where 20% of the data is generated by DDoS attackers is presented in Figure 3.

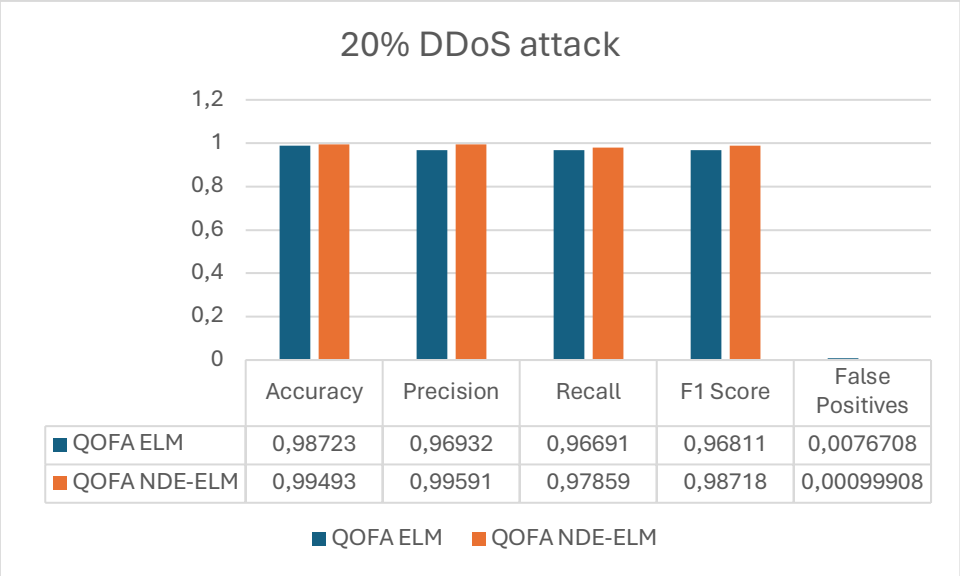

Figure 3. Comparing the Performance of NDE-EML and Conventional EML on Data with 20% DDoS Attack Traffic

The performance of the two models, EML and NDE-EML in a network containing 20% of data generated by attackers is compared and the results are presented in Figure 3. The QOFA algorithm was applied to both models to reduce their features. In every metric, NDE-EML performs better. It obtained a higher accuracy of 99.493% as opposed to 98.723% which suggests a lower total error rate. Additionally, it has a superior precision (99.591% vs. 96.932%), which means that there are fewer false positives than real positives. Furthermore, NDE-EML's recall is marginally higher (97.859% as opposed to 96.691%) which shows that it accurately identifies a greater percentage of true positives. For NDE-EML, the F1 score is higher (98.718% vs. 96.811%). Interestingly, NDE-EML's false positive rate of 0.099908% is far lower than EML's of 0.76708%. NDE-EML is the best model in this scenario since it performs better than EML in all the scenarios. Figure 4 compares NDE-EML

and EML in a network, the experiment was conducted using the dataset where 40% of was generated by DDoS attackers.

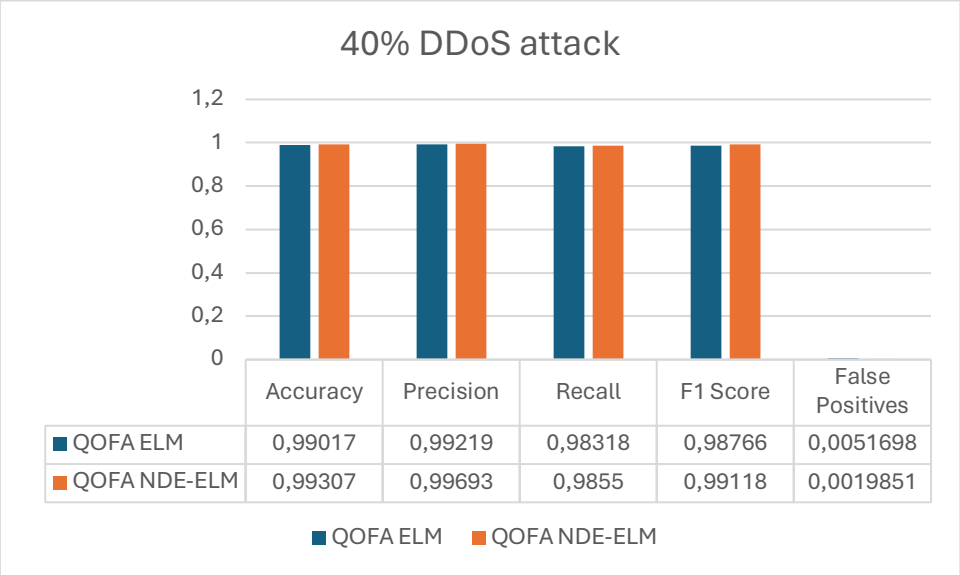

Figure 4. Comparing the Performance of NDE-EML and Conventional EML on Data with 40% DDoS Attack Traffic

Figure 4 shows the performance of EML and NDE-EML. They both used the QOFA for feature selection in a network where DDoS attackers generate 40% of the data. In every metric, the NDE-EML model outperformed EML. With an accuracy of 99.307% as opposed to 99.017% for EML, NDE-EML performed better and had fewer classification errors. With a precision of 99.693%, NDE-EML outperforms EML's 99.219%, meaning that NDE-EML has a lower ratio of false positives to true positives. Despite the close similarity of the recall results for both models, NDE-EML is marginally better than EML. It has a recall of 98.550% compared to 98.318%. The NDE-EML's F1 score is higher at 99.118% compared to EML's 98.766%. Moreover, compared to EML's 0.51698% false positive rate, NDE-EML its rate is 0.19851%. Even though the results are marginally different, Table 2 shows that NDE-EML is superior. Figure 5 depicts the comparison of NDE-EML and EML in a network where 60% of data was generated by DDoS attacks.

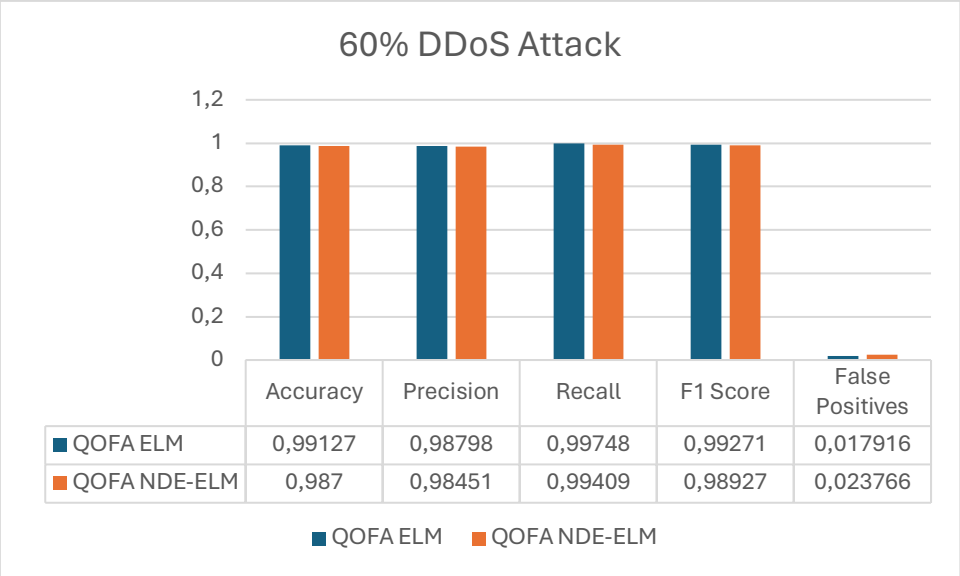

Figure 5. Comparing the Performance of NDE-EML and Conventional EML on Data with 60% DDoS Attack Traffic

In Figure 5, the performance of EML is compared to NDE-EML. The two models were both subjected to QOFA feature selection. The EML model achieved an accuracy of 99.13%. It also had a precision of 98.80%, indicating a low number of false positives, and a recall of 99.75%, identifying almost all true positives. Its F1 score of 99.27% reflects a balanced measure of precision and recall. However, EML misclassified 1.79% of instances as false positives. On the other hand, the NDE-EML model achieved a slightly lower accuracy of 98.70%, with a precision of 98.45%, a recall of 99.41%, and an F1 score of 98.93%. Notably, NDE-EML had a higher false positive rate of 2.38%. Thus, while EML demonstrated superior overall performance and a lower rate of false positives compared to NDE-EML, both models achieved good classification results with NDE-EML being slightly less effective in minimizing false positives. Figure 6 compares NDE-EML and EML results in a network where DDoS attacks generate 80% of data.

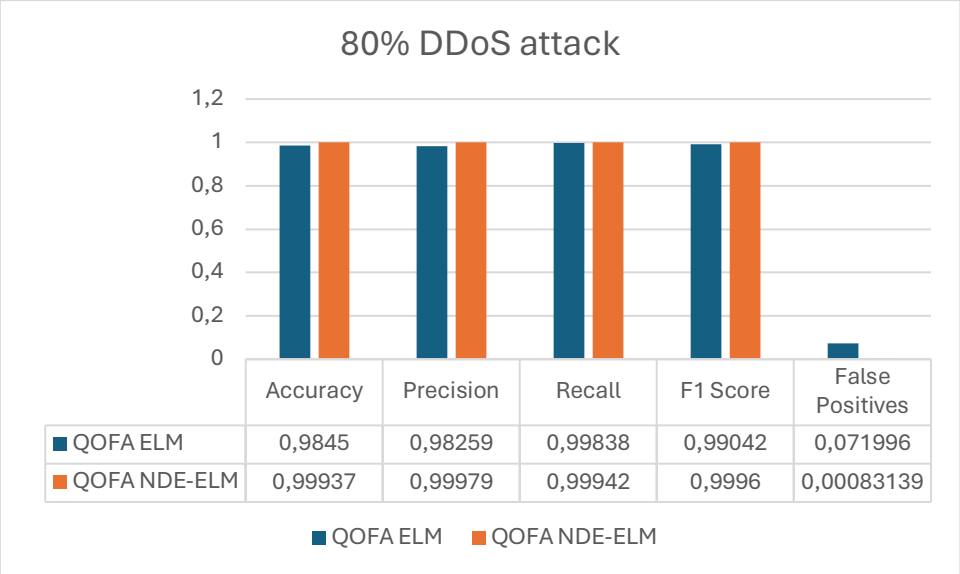

Figure 6. Comparing the Performance of NDE-EML vs traditional EML on Data with 80% DDoS Attack Traffic

Figure 6 presents the performance results of EML and NDE-EML in a network scenario where 80% of data is generated by a DDoS attack. The EML model achieved an accuracy of 98.45%, with a precision of 98.259%, indicating a reasonably low number of false positives. Its recall was very high at 99.838%, meaning it effectively identified nearly all true positives, resulting in an F1 score of 99.042%. However, EML had a higher false positive rate of 7.1996%. In contrast, the NDE-EML model significantly outperformed EML, with an accuracy of 99.937%, and precision of 99.979%, suggesting an almost negligible number of false positives.
It also maintained a high recall of 99.942%, and its F1 score of 99.96% demonstrates a good balance between precision and recall. Moreover, NDE-EML had a low false positive rate of 0.083139%. Thus, NDE-EML's performance results indicate it is superior to EML, particularly in minimizing false positives and achieving good accuracy, precision, recall, and F1 score. The comparative results of NDE-EML and EML in various network scenarios are shown in Figures 7–11. Figure 7 presents an investigation of the accuracy of the models in various network configurations.

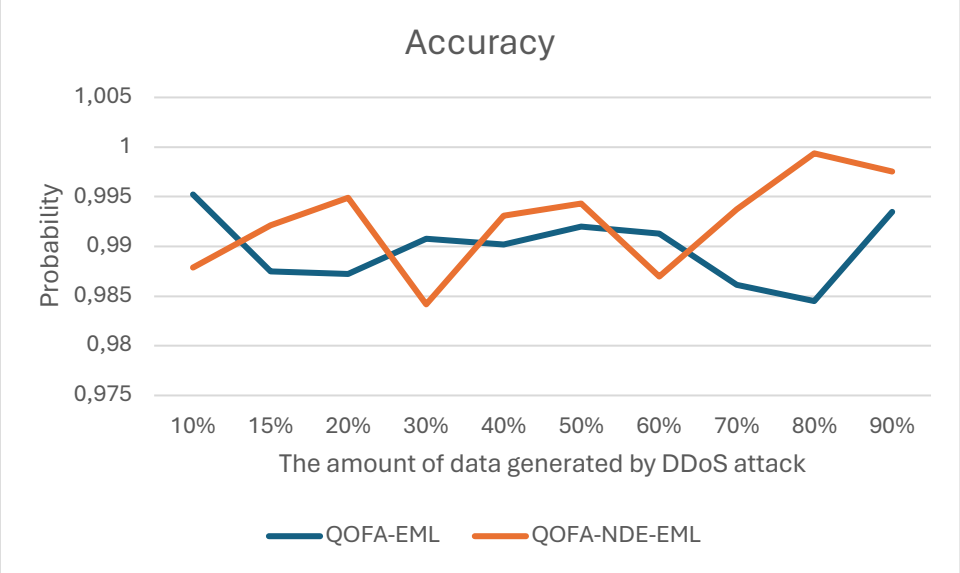

Figure 7. Accuracy of the algorithm in different network scenarios.

Figure 7 presents the accuracy results of two models, QOFA-EML and QOFA-NDE-EML in various network scenarios where the percentage levels of DDoS attack-generated data vary from 10% to 90%. Accuracy measures the proportion of correct predictions out of the total predictions made. QOFA-EML's accuracy ranges from 98.45% (at 80%) to 99.52% (at 10%), indicating high accuracy in all the scenarios. However, its accuracy decreases slightly as the percentage level increases, with the lowest accuracy at 80%. QOFA-NDE-EML's accuracy also ranges from 98.7% (at 60%) to 99.93% (at 80%), showing high accuracy across all levels. Notably, its accuracy increases as the percentage increases, with the highest accuracy at 80% (99.93%) and 90% (99.75%). Comparing the two models, QOFA-NDE-EML

generally outperforms QOFA-EML, particularly at higher percentage levels (80% and 90%). This suggests that QOFA-NDE-EML is more accurate in predicting outcomes when the DDoS traffic is high. Overall, both models demonstrate high accuracy, but QOFA-NDE-EML appears to have a slight edge, especially at higher percentage levels. Precision results of QOFA-NBEML and QOFA-EML are presented in Figure 8.

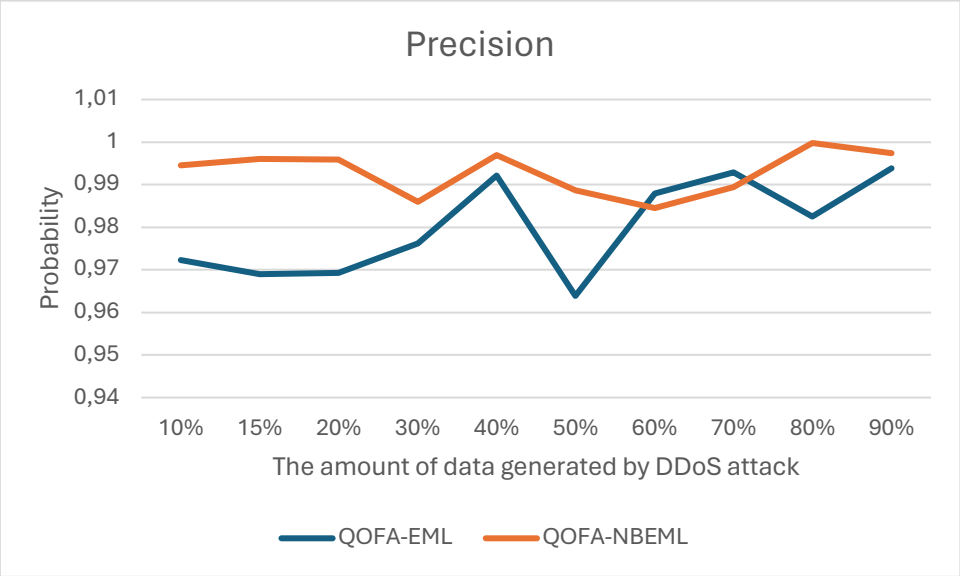

Figure 8: Precision of the algorithm in different network scenarios.

In Figure 8, QOFA-NDE-EML outperformed QOFA-EML in terms of precision, with a noticeable difference in all network scenarios. QOFA-NDE-EML's precision remains stable, ranging from 0.98451 at 60% to 0.99979 at 80%, demonstrating its reliability and consistency. In contrast, QOFA-EML's precision varies. It ranges from 0.96388 at 50% to 0.99385 at 90%, indicating some fluctuations in its performance. Notably, both models achieved higher precision at higher percentage levels, with QOFA-NDE-EML reaching 0.99979 at 80%, indicating its good accuracy at this level. Overall, QOFA-NDE-EML's consistently high precision makes it a more reliable choice, while QOFA-EML's varying precision indicates its unreliability. Figure 9 presents an analysis of the recall metric for both QOFA-NBEML and QOFA-EML, providing a comparative view of their outcomes.

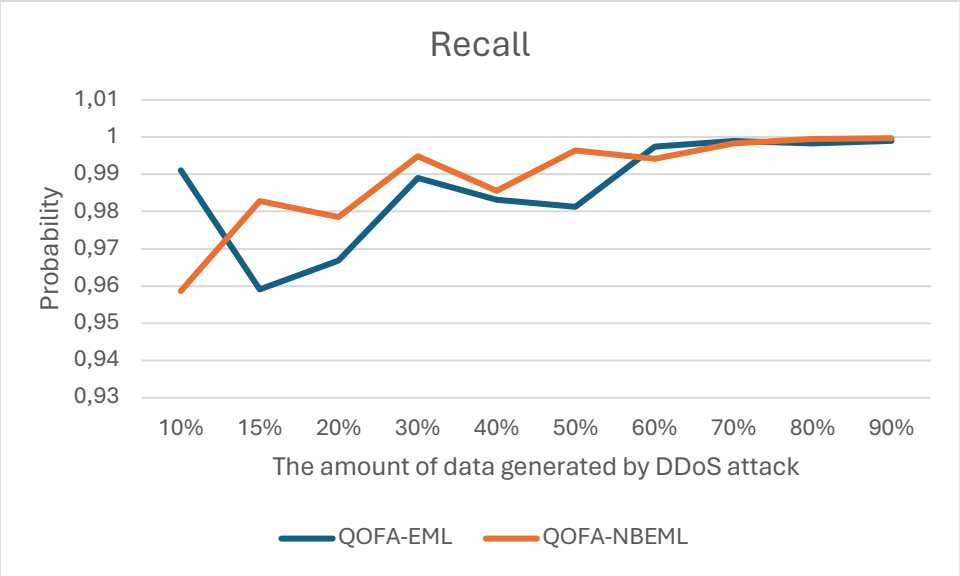

Figure 9. Recall of the algorithm in different network scenarios.

It is evidenced in Figure 9 that QOFA-EML's recall performance is good, but it fluctuates slightly across different network scenarios. It starts with a recall of 0.99103 at 10%, drops to 0.95911 at 15%, and then gradually increases to 0.99895 at 70%. However, it slightly decreases to 0.99838 at 80% before increasing to 0.99893 at 90%. On the other hand, QOFA-NDE-EML's recall performance exhibits a consistent and steady improvement as the percentage levels increase. It starts with a recall of 0.95866 at 10%, increases to 0.99483 at 30%, and continues to increase to 0.9997 at 90%. QOFA-NDE-EML's recall surpasses QOFA-EML's at most levels with a significant margin at higher percentage levels (80% and 90%). The consistent improvement in QOFA-NDE-EML's recall suggests that it is better at detecting true positives as the percentage levels increase.

This is particularly important in our study since false negatives have significant consequences, which can make the network unavailable. In contrast, QOFA-EML's fluctuating recall performance indicates that it is more sensitive to changes in the data or model parameters. While it still achieves high recall rates, its performance is less reliable than QOFA-NDE-EML's performance especially at higher percentage levels. In Overall, both models demonstrate high recall rates, but QOFA-NDE-EML's consistent improvement and superior performance at higher levels make it a more reliable in detecting true positives. Figure 10 presents an analysis of the QOFA-NBEML and QOFA-EML evaluation results which compares the F1-score measures of the models.

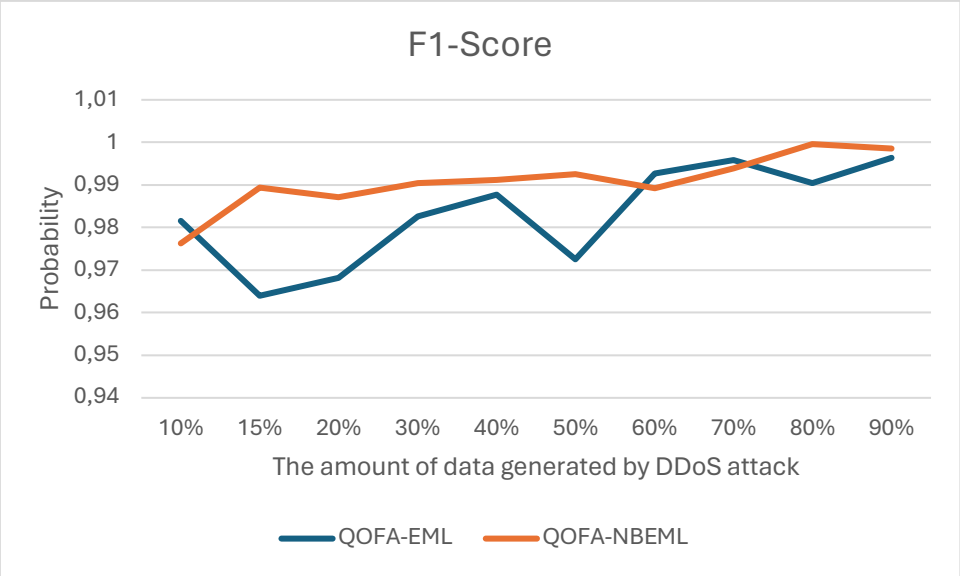

Figure 10. F1-Score of the algorithm in different network scenarios.

Figure 9 shows that the QOFA-EML's F1-score performance is good at lower percentage levels (0.98158 at 10% and 0.96398 at 15%) which indicates good detection rates in network scenarios with few DDoS attackers. However, its performance fluctuates as the attack intensity increases with a notable drop to 0.97247 at 50% and then increase to 0.99638 at 90%. This variability suggests that QOFA-EML is not efficient in detecting DDoS attacks of varying severity. In contrast, QOFA-NDE-EML's F1-score performance is more consistent and robust across all network scenarios. It starts strong at 0.97628 at 10% and maintains a high level of performance, with a slight drop to 0.98718 at 20% and then a steady increase to 0.99852 at 90%. This consistent performance indicates that QOFA-NDE-EML is effective in detecting DDoS attacks in wide range of network scenarios with increasing intensity of attacks. The superior performance of QOFA-NDE-EML is particularly notable at higher percentage levels (80% and 90%), where the accuracy of DDoS attack detection is critical. QOFA-NDE-EML's F1-score of 0.9996 at 80% and 0.99852 at 90% indicates its effectiveness in detecting severe DDoS attacks. Overall, the comparative F1-scores suggests that QOFA-NDE-EML is a more reliable and effective model for detecting DDoS attacks particularly in high DDoS attack intensity scenarios where detection efficiency is crucial. Figure 10 provides a comprehensive comparison of false positive rates for both QOFA-NBEML and QOFA-EML models.

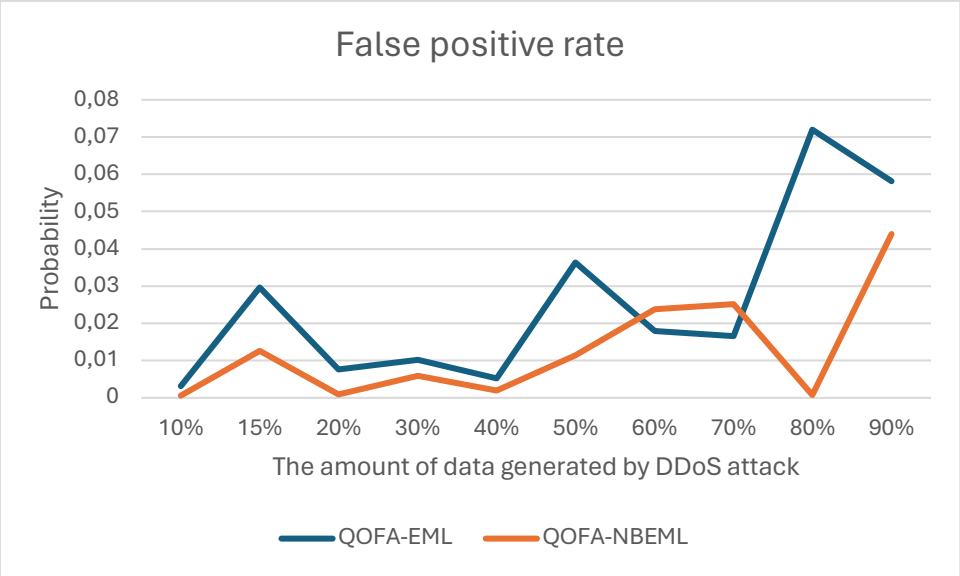

Figure 11. False positive rate of the algorithm in different network scenarios.

In Figure 11, QOFA-EML's False positive rate (FPR) is high, indicating a significant likelihood of misclassifying legitimate traffic as DDoS attacks. This could lead to unnecessary resource waste, legitimate traffic blocking, and potential security breaches. The FPR fluctuates across different attack intensities, suggesting that QOFA-EML's performance is inconsistent and may be affected by the severity of the DDoS attacks. In contrast, QOFA-NDE-EML's FPR is low indicating high accuracy in detecting legitimate traffic. This suggests that QOFA-NDE-EML is effective in reducing false positives, which is critical in high-security networks where non malicious traffic must be prioritized.

The FPR remains relatively stable across different attack intensities, indicating that QOFA-NDE-EML's performance is robust and consistent, even in the face of increasing attack severity. The significant difference in FPR between QOFA-EML and QOFA-NDE-EML is particularly notable at higher attack intensities (50%-90%). QOFA-EML's FPR increases indicating a higher likelihood of misclassifying legitimate traffic while QOFA-NDE-EML's FPR remains relatively stable, depicting a continued high accuracy in detecting legitimate traffic. Overall, figure 10 shows that QOFA-NDE-EML is a more reliable and accurate model for detecting DDoS attacks, with a lower likelihood of misclassifying legitimate traffic. QOFA-EML's higher FPR and fluctuating performance make it less suitable for high-security networks where accurate detection and minimal false positives are crucial.

v. Analytical Results

t-statistics

This section presents the results of a statistical analysis where the performance of QOFA-EML and QOFA-NDE-EML in detecting DDoS attacks were compared. The analysis includes t-test results for accuracy, precision, recall, F1-score, and false positive rate presented in Tables 1 to 5. The results validate the simulation results and show that QOFA-NDE-EML outperforms QOFA-EML in all scenarios. The difference is also significant in all scenarios. The study highlights that

QOFA-NDE-EML outperforms other detection methods in terms of detection and reliability. The significant differences in performance between the two models confirm that QOFA-NDE-EML is a more accurate and reliable model for DDoS detection. Table 1 compares QOFA-EML and QOFA-NDE-EML accuracy using a t-Test Two-tale and Assuming Equal Variances which shows a significant difference between the two methods.

Table 1: t-Test Two-tale Assuming Equal Variances for Accuracy

	QOFA-EML	**QOFA-NDE-EML**
Mean	0,989827	0,992414
Stand deviation	0,0034116	0,004770875
Variance	1,164E-05	2,27612E-05
Observations	10	10
t Stat	1,394812304	
t-Critical (two tail)	0,180048044	

Table 1 presents the t-test results where the accuracy means of QOFA-EML and QOFA-NDE-EML are compared. The mean of QOFA-EML is 0.989827, while the mean of QOFA-NDE-EML is 0.992414. The standard deviations are 0.0034116 and 0.004770875 respectively. With 10 observations in each sample, the t-statistic is 1.394812304 highlighting a statistically significant difference between the means. The critical t-value for a two-tailed test is 0.180048044, which is lower than the observed t-statistic which suggests that the difference between QOFA-EML and QOFA-NDE-EML is significant. These analytical results are a confirmation of the findings in Figure 6. Table 2 presents the comparison of the precision results of QOFA-EML and QOFA-NDE-EML using a t-Test Two-tale.

Table 2. t-Test Two-Sample Assuming Equal Variances for Precision

	QOFA-EML	**QOFA-NDE-EML**
Mean	0,980017	0,992911
stand deviation	0,0112882	0,005317258
Variance	0,0001274	2,82732E-05
Observations	10	10
t Stat	3,267749116	
t-Critical (two tail)	0,004274879	

The precision analytical results in Table 2 illustrate that QOFA-NDE-EML is a more accurate model in detecting DDoS attacks with a higher mean precision of 0.992911 compared to QOFA-EML's 0.980017. This means that QOFA-NDE-EML can detect a higher percentage of DDoS attacks which reduces the number of false negatives and improves overall detection accuracy. Furthermore, QOFA-NDE-EML's lower standard deviation (0.005317258) and variance (2.82732E-05) indicate that its precision is more consistent and reliable in all network scenarios and testing conditions. In contrast, QOFA-EML's higher standard deviation (0.0112882) and

variance (0.0001274) suggest that its precision may be more variable and less reliable.

The significant difference in precision between the two models as indicated by the t-statistic (3.267749116) and t-critical value (0.004274879) suggests that QOFA-NDE-EML's improved accuracy is superior in DDoS detection. Overall, the results suggest that QOFA-NDE-EML is a more accurate and reliable model for detecting DDoS attacks making it a better choice for network security applications where accurate detection is critical. Table 3 presents a comparison of the recall results of QOFA-EML and QOFA-NDE-EML using a t-Test Two-tale.

Table 3. t-Test Two-Sample Assuming Equal Variances for Recall

	QOFA-EML	**QOFA-NDE-EML**
Mean	0,986427	0,988836
stand deviation	0,0140292	0,012932537
Variance	0,0001968	0,000167251
Observations	10	10
t Stat	0,399250129	
t-Critical (two tail)	0,694406134	

The recall statistics results in Table 3 show that QOFA-NDE-EML is slightly superior at detecting DDoS attacks with a mean recall of 0.988836 compared to QOFA-EML's 0.986427. This suggests that QOFA-NDE-EML is more effective in detecting DDoS attacks which reduces the number of false negatives and improves overall detection accuracy. Moreover, QOFA-NDE-EML's lower standard deviation (0.012932537) and variance (0.000167251) indicate that its recall is more consistent and reliable across different scenarios and testing conditions.

This means that QOFA-NDE-EML's performance is more stable less prone to fluctuations and more effective in detecting DDoS attacks. Although the difference in recall between the two models is not statistically significant, QOFA-NDE-EML's consistent performance and slightly higher recall suggest that it may be more reliable and effective in detecting DDoS attacks. The results suggest that both models are effective in detecting DDoS attacks however, QOFA-NDE-EML's slightly higher recall and lower variability make it more ideal for network security applications where accurate and reliable detection is crucial. F1-Score comparison results using the t test are presented in Table 4.

Table 4. t-Test Two-Sample Assuming Equal Variances for F1-Score

	QOFA-EML	**QOFA-NDE-EML**
Mean	0,983186	0,990817
stand deviation	0,0116089	0,006479271
Variance	0,0001348	4,1981E-05
Observations	10	10
t Stat	1,815115994	
t-Critical (two tail)	0,086205482	

The F1-score, a harmonic mean of precision and recall provides an evaluation of a model's performance. In the DDoS dataset, QOFA-NDE-EML's higher mean F1-score (0.990817) shows that it achieves a better balance between precision (0.992911) and recall (0.988836) compared to QOFA-EML (mean F1-score: 0.983186, precision: 0.980017, recall: 0.986427). This is presented in Table 4. QOFA-NDE-EML's lower variability in F1-score (standard deviation: 0.006479271, variance: 4.1981E-05) shows that its performance is more consistent in different scenarios and testing conditions, making it more reliable for DDoS detection.

The significant difference in F1-score between the two models (t-statistic: 1.815115994, t-critical: 0.086205482) depicts that QOFA-NDE-EML's improvement in the balance between precision and recall is superior. QOFA-NDE-EML's higher F1-score and lower variability depict that it is a more effective and reliable model for DDoS detection. It achieves a better balance between detecting attacks and reducing false positives. This makes QOFA-NDE-EML a more suitable choice for network security applications where accurate and reliable detection is crucial. Table 5 presents the t-test results of the false positive rate of QOFA-NDE-EML and QOFA-EML.

Table 5. t-Test Two-Sample Assuming Equal Variances for False positive rate

	QOFA-EML	**QOFA-NDE-EML**
Mean	0,025687	0,012735062
stand deviation	0,0234893	0,014313386
Variance	0,0005517	0,000204873
Observations	10	10
t Stat	1,489003744	
t-Critical (two tail)	0,153798621	

Table 5 presents the results of the false positive rate statistical analysis which shows that QOFA-NDE-EML performs significantly better than QOFA-EML. Specifically, QOFA-NDE-EML has a lower mean false positive rate (0.012735062 vs 0.025687), with less false alarms. Additionally, QOFA-NDE-EML performs more consistently and reliably as evidenced by its lower variance (0.000204873 vs 0.0005517) and standard deviation (0.014313386 vs 0.0234893). The significant difference in the false positive rate (t-statistic: 1.489003744, t-critical: 0.153798621) confirms the improvement in QOFA-NDE-EML. The lower false positive rate depicts better detection accuracy and fewer false alarms making QOFA-NDE-EML a more effective and reliable model for DDoS detection which improves overall network security.

Effect Size: A Measure of the Magnitude of Difference

Effect size (ES) falls under the umbrella of statistical theories which can be used to analyse and interpret the performance of machine learning algorithms. This statistical theory provides a more meaningful in the interpretation of results beyond simply determining if the performance difference of the algorithm is statistically significance by quantifying the magnitude of the different groups. Above we performed a t-test to check if there is a statistical difference in the performance of

the machine learning algorithm. Effective size (ES) assesses and checks if the difference is large enough to be practical and important. It also helps by combining results from multiple scenarios in the meta-analysis as it allows standardized comparisons across different studies.

Multiple ES measures can be used such as Cohen's d, Pearson's r, and Eta-squared (η^2). Cohen's d is used for the comparison of the means of two groups of which the three different interpretations from the findings if we find *d* being 0.2 it means that there is a small effect, 0.5 medium effect, and a larger effect 0.8. *d is calculated as follows.*

$$d = m_1 - m_2 \,/\, s_{pooled} \tag{8}$$

Where:

m_1 and m_2 are the means of the two groups and s_{pooled} is the pooled standard deviation which is the weighted average of the standard deviations of the two groups. s_{pooled} is calculated as

$$s_{pooled} = \sqrt{((n_1 - 1) * s_1^2 + (n_2 - 1) * s_2^2)/(n_1 + n_2 - 2)} \tag{9}$$

Where: n_1 and n_2 , s_1 and s_2 are the sample size and standard deviation of the two groups. Figure 12 presents Cohen's value for metrics used to compare the effectiveness of QOFA-EML and QOFA-NDE-EML in several network scenarios. t-test has demonstrated that there is a significant difference in the precision of QOFA-NDE-EML compared to QOFA-EML. Figure 12 shows the highest recorded ES of 1,461382. This proves that QOFA-NDE-EML is good at identifying positive instances accurately. ES for accuracy and F1-Score is moderate indicating that QOFA-NDE-EML performs well but lower than precision. Recall and False Positive Rate have smaller Cohen's d values indicating a less significant effect or difference compared to the baseline.

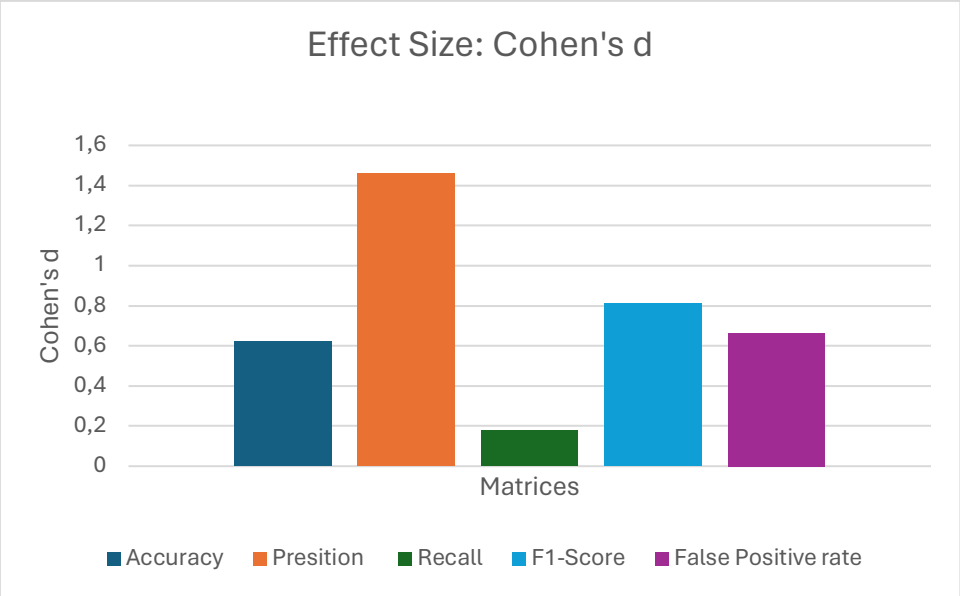

Figure 12. effect size of different matrices

D. Conclusions

In this work, we have demonstrated that the improvement of the EML algorithm leads to improved performance in addressing DDoS attacks. Neighbourhood differential evolution was utilised to initialise the input-hidden layer weights and hidden biases. The NDE-EML was designed and implemented in MATLAB to detect and classify the traffic as normal or DDoS attacks. The model was trained and tested using a synthetic dataset.

The results demonstrate that QOFA-NDE-EML outperforms QOFA-EML particularly at higher percentage levels (80% and 90%). This shows that QOFA-NDE-EML is more accurate in predicting outcomes at these levels. This is a good in the reduction latency in MEC caused by DDoS attacks.

Analytical results of both t-test and Effect Size also confirm the simulation results in showing that there is a significant difference in the performance of the proposed model. The proposed model may also be evaluated using other datasets.

E. References

[1] W. S. Admass, Y. Y. Munaye, and A. A. Diro, "Cyber security: State of the art, challenges and future directions," Jan. 01, 2024, *KeAi Communications Co.* doi: 10.1016/j.csa.2023.100031.

[2] B. Aryal, R. Abbas, and I. B. Collings, "SDN Enabled DDoS Attack Detection and Mitigation for 5G Networks," *Journal of Communications*, pp. 267–275, 2021, doi: 10.12720/jcm.16.7.267-275.

[3] E. Ahmed and M. H. Rehmani, "Mobile Edge Computing: Opportunities, solutions, and challenges," May 01, 2017, *Elsevier B.V.* doi: 10.1016/j.future.2016.09.015.

[4] H. Huang, Y. Wu, F. Xiao, and R. Malekian, "An Efficient Signature Scheme Based on Mobile Edge Computing in the NDN-IoT Environment," *IEEE Trans Comput Soc Syst*, vol. 8, no. 5, pp. 1108–1120, Oct. 2021, doi: 10.1109/TCSS.2021.3076209.

[5] A. Kumar and D. Singh, "Detection and prevention of DDoS attacks on edge computing of IoT devices through reinforcement learning," *International Journal of Information Technology (Singapore)*, vol. 16, no. 3, pp. 1365–1376, Mar. 2024, doi: 10.1007/s41870-023-01508-z.

[6] K. Alsubhi, "A Secured Intrusion Detection System for Mobile Edge Computing," *Applied Sciences (Switzerland)*, vol. 14, no. 4, Feb. 2024, doi: 10.3390/app14041432.

[7] G. S. Kushwah and V. Ranga, "Detecting DDoS Attacks in Cloud Computing Using Extreme Learning Machine and Adaptive Differential Evolution," *Wirel Pers Commun*, vol. 124, no. 3, pp. 2613–2636, Jun. 2022, doi: 10.1007/s11277-022-09481-9.

[8] Shikai Wang, Haotian Zheng, Xin Wen, and Fu Shang, "DISTRIBUTED HIGH-PERFORMANCE COMPUTING METHODS FOR ACCELERATING DEEP LEARNING TRAINING," *Journal of Knowledge Learning and Science Technology*, vol. 3, no. 3, pp. 108–127, Sep. 2024, doi: https://doi.org/10.60087/jklst.vol3.n3.p108-126.

[9] M. S. Ansari, S. H. Alsamhi, Y. Qiao, Y. Ye, and B. Lee, "Security of Distributed Intelligence in Edge Computing: Threats and Countermeasures," in *Palgrave Studies in Digital Business and Enabling Technologies*, Palgrave Macmillan, 2020, pp. 95–122. doi: 10.1007/978-3-030-41110-7_6.

[10] N.-N. Dao, T. V. Phan, U. S. ad, J. Kim, T. Bauschert, and S. Cho, "Securing Heterogeneous IoT with Intelligent DDoS Attack Behavior Learning," *IEEE Syst J*, vol. 16, no. 2, Jun. 2022, doi: 10.1109/JSYST.2021.3084199.

[11] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52–65, Jan. 2013, doi: 10.1016/j.neunet.2012.09.018.

[12] T. V. Phan, N. K. Bao, and M. Park, "Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks," *Journal of Network and Computer Applications*, vol. 91, pp. 14–25, Aug. 2017, doi: 10.1016/j.jnca.2017.04.016.

[13] A. Serrano Mamolar, P. Salvá-García, E. Chirivella-Perez, Z. Pervez, J. M. Alcaraz Calero, and Q. Wang, "Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks," *Journal of Network and Computer Applications*, vol. 145, p. 102416, Nov. 2019, doi: 10.1016/j.jnca.2019.102416.

[14] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks," *Computer Networks*, vol. 188, p. 107871, Apr. 2021, doi: 10.1016/j.comnet.2021.107871.

[15] H. Li *et al.*, "A Cooperative Defense Framework Against Application-Level DDoS Attacks on Mobile Edge Computing Services," *IEEE Trans Mob Comput*, vol. 22, no. 1, pp. 1–18, Jan. 2023, doi: 10.1109/TMC.2021.3086219.

[16] Y. Zhao *et al.*, "Traffic Scheduling Strategy for Mitigating DDoS Attack in Edge Computing-enabled TWDM-PON," in *25th Opto-Electronics and Communications Conference, OECC 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/OECC48412.2020.9273645.

[17] Nhu-Ngoc Dao, Duc-Nghia Vu, Yunseong Lee, Minho Park, and Sungrae Cho, "MAEC-X: DDoS prevention leveraging multi-access edge computing," in

International Conference on Information Networking (ICOIN), Chiang Mai, Thailand: IEEE, Apr. 2018.

[18] M. Gusatu and R. F. Olimid, "Improved security solutions for DDoS mitigation in 5G Multi-access Edge Computing," in *Innovative Security Solutions for Information Technology and Communications*, Peter Y.A. Ryan and Cristian Toma, Eds., Springer, Cham, Oct. 2022. doi: https://doi.org/10.1007/978-3-031-17510-7_19.

[19] L. Fernández Maimó, A. Huertas Celdrán, M. Gil Pérez, F. J. García Clemente, and G. Martínez Pérez, "Dynamic management of a deep learning-based anomaly detection system for 5G networks," *J Ambient Intell Humaniz Comput*, vol. 10, no. 8, pp. 3083–3097, Aug. 2019, doi: 10.1007/s12652-018-0813-4.

[20] Shivangi Singh, Khushboo Kumari, Shashank Gupta, Amit Dua, and Neeraj Kumar, "Detecting Different Attack Instances of DDoS Vulnerabilities on Edge Network of Fog Computing using Gaussian Naive Bayesian Classifier," in *IEEE International Conference on Communications Workshops (ICC Workshops)*, Dublin, Ireland: IEEE, Jul. 2020.

[21] N.-N. Dao *et al.*, "Securing Heterogeneous IoT With Intelligent DDoS Attack Behavior Learning," *IEEE Syst J*, vol. 16, no. 2, pp. 1974–1983, Jun. 2022, doi: 10.1109/JSYST.2021.3084199.

[22] E. Guresen and G. Kayakutlu, "Definition of Artificial Neural Networks with comparison to other networks," in *Procedia Computer Science*, 2011, pp. 426–433. doi: 10.1016/j.procs.2010.12.071.

[23] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, IEEE, pp. 985–990. doi: 10.1109/IJCNN.2004.1380068.

[24] G. Bin Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," in *Neurocomputing*, Oct. 2008, pp. 3460–3468. doi: 10.1016/j.neucom.2007.10.008.

[25] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[26] G. Bin Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, Oct. 2007, doi: 10.1016/j.neucom.2007.02.009.

[27] Gao Huang, Shiji Song, J. N. D. Gupta, and Cheng Wu, "Semi-Supervised and Unsupervised Extreme Learning Machines," *IEEE Trans Cybern*, vol. 44, no. 12, pp. 2405–2417, Dec. 2014, doi: 10.1109/TCYB.2014.2307349.

[28] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, Jun. 2009, doi: 10.1109/TEVC.2008.2009457.

[29] M. M. Shafi, A. H. Lashkari, V. Rodriguez, and R. Nevo, "Toward Generating a New Cloud-Based Distributed Denial of Service (DDoS) Dataset and Cloud Intrusion Traffic Characterization," *Information (Switzerland)*, vol. 15, no. 4, Apr. 2024, doi: 10.3390/info15040195.

[30] S. Bhatia, D. Schmidt, G. Mohay, and A. Tickle, “A framework for generating realistic traffic for Distributed Denial-of-Service attacks and Flash Events,” *Comput Secur*, vol. 40, pp. 95–107, Feb. 2014, doi: 10.1016/j.cose.2013.11.005.