Science, Engineering and Technology Vol. 6, No. 1, pp. 39-51
www.setjournal.com https://doi.org/10.54327/set2026/v6.i1.293

Optimised Feature Selection for Mobile Edge DDoS Detection Using Quasi-Opposite Firefly Algorithm

Sekgoari Semaka Mapunya, Mthulisi Velempini

University of Limpopo, Private Bag X1106, Sovenga, 0727, Polokwane, 0700, South Africa.

Abstract

Mobile edge computing (MEC) reduces latency for delay-sensitive applications by bringing computations closer to end users. However, this technology is vulnerable to security threats, notably Distributed Denial of Service (DDoS) attacks. DDoS attacks are characterised by distributed malicious nodes that flood the target with data packets, causing system unavailability or performance degradation. This contradicts the objective of the MEC, which is to reduce delay and latency. To address this, we propose a feature selection technique to improve the detection of DDoS attacks in MEC using machine learning techniques. The proposed approach employs quasi-opposite-based learning (QOBL), a concept often utilised in differential evolution algorithms, to modify the Firefly Algorithm (FA) to form a Quasi-Opposite Firefly Algorithm (QOFA) to optimise feature selection. FA excels at navigating complex feature spaces for global optimisation but suffers from premature convergence to local optima. QOBL mitigates this by guiding FA toward local solutions, improving efficiency and detection accuracy. By selecting only the most relevant features, the QOFA reduces computational complexity while maintaining robust performance. Simulations in MATLAB demonstrated that QOFA outperformed traditional FA, achieving a higher detection accuracy (up to 95%). This approach enhances the efficiency of machine learning models for DDoS detection in MEC, ensuring a reliable and low-latency network performance that is critical for real-time applications.

Keywords*:* Mobile Edge Computing (MEC), Distributed Denial of Service attack (DDoS), Feature selection, Quasi-Opposite Based Learning (QOBL), Firefly algorithm.

1. Introduction

MEC technology brings cloud computing capabilities closer to end users, reducing latency in the execution of tasks by the user [1]. However, this proximity also exposes MEC to severe security threats, notably DDoS attacks. In a DDoS attack, compromised nodes flood the target with malicious packets, overwhelming resources and negating MEC's core benefit, which is low latency. This not only disrupts service availability but also delays detection and response, especially when high-dimensional network data overwhelms traditional intrusion detection systems.

Current machine learning-based DDoS detection methods suffer from high computational complexity, slow training, and reduced accuracy due to irrelevant, redundant, and noisy features [2]. While feature selection is critical for improving efficiency, conventional approaches, such as Sequential Feature Selection (SFS) and standard swarm-based methods, often converge prematurely or fail to explore optimal feature subsets effectively [3].

Hence, there is a need to design schemes that can detect attacks by analysing the least possible amount of data for faster detection. This study addresses two key research questions: (1) How can premature convergence

Corresponding author: Sekgoari Semaka Mapunya (sekgoari.mapunya@ul.ac.za)
Received: 2 June 2025; Revised: 10 December 2025; Accepted: 17 December 2025; Published: 21 Februrary 2026

in swarm-based feature selection be overcome to improve DDoS detection in MEC? (2) Can a quasi-oppositional enhancement to the Firefly Algorithm yield faster and more accurate detection under varying attack intensities? To answer these, we propose QOFA, a novel improved method that uniquely integrates QOBL into the Firefly Algorithm to accelerate global exploration and escape local optima, enabling robust selection of discriminative features in high-dimensional MEC traffic. The specific objectives were as follows:

- To propose QOFA as a superior alternative to traditional FA for MEC DDoS detection.
- Reduce dimensionality and noise to enable low-latency, real-time classification.
- Evaluate QOFA vs. TFA using Extreme Learning Machine (ELM) and Support Vector Machine (SVM) across synthetic datasets (10%–90% DDoS traffic).
- Validate statistically significant gains in accuracy, precision, recall, F1-score, and false positives via t-tests.
- Demonstrate QOFA-ELM's efficiency for resource-constrained MEC environments.

2. Related work

Feature selection has been widely explored to enhance DDoS detection efficiency, particularly in resource-constrained environments such as MEC. This section critically reviews recent approaches and contrasts them with the proposed QOFA.

We employed wrapper methods that are capable of detecting feature connections. Current methods, such as Sequential Feature Selection (SFS), generate good results; however, their performance is not efficient [4], which can be addressed by swarm algorithms.

In [5]the authors analyse the increasing complexity and quantity of network-related attacks. They also highlighted how traditional intrusion detection systems fail to withstand these advanced cyberthreats. To improve the detection efficiency, the authors proposed a novel feature selection technique and a hybrid attack detection model. To enhance the detection capability of machine learning-based intrusion detection systems, researchers have focused on feature selection as a crucial preprocessing step. The research presents novel techniques for feature selection and classification, making a substantial contribution to the IDS literature. The work achieved high accuracy rates for detecting intrusion attacks using the proposed methods, which outperforms most advanced techniques [5]. A technique that includes a training stage with datasets such as KDD'99 and UNSW-NB15, and a testing stage with matching test datasets is presented. The advantages of the technique include faster detection times and higher accuracy rates.

The FSAP technique reduced the number of features from 41 to 10, simplifying the model without sacrificing its predictive capabilities. However, the study only uses the KDD '99 dataset, which is known for having flaws such as built-in biases and outdated attack characteristics that do not accurately reflect the state of current network threats, despite its widespread use and popularity. For a more comprehensive analysis, the UNSW-NB15 dataset was employed; however, validating the models on even more modern and diverse datasets should be considered.

There is a need to address the challenges of this scheme in practical settings. Although the simulation findings are promising, there is a need to implement them in a real-world setting. These improvements may provide valuable perspectives on the efficacy of the FSAP and SABADT techniques in dynamic network scenarios featuring dynamic attack vectors. An analysis of how these techniques may be integrated with current IDSs and related to a larger cybersecurity infrastructure would also be beneficial. The model's performance should be considered in the context of an integrated security system.

The authors in [6] provided a unique technique that makes use of the feature and model selection (FAMS) Framework to detect DDoS Attacks. Motivated by the biological principle of "survival of the fittest", this framework combines several datasets, feature selection algorithms, and an optimised machine learning model to achieve superior performance over discrete feature selection and machine learning techniques. This study achieved good experimental results that demonstrate the efficacy of the FAMS framework. However, there remains room for improvement.

The scalability and resilience of the method must be considered in several network scenarios and against unknown attack vectors. Furthermore, this study does not examine the operational expenses and resource requirements related to implementing the solution in real-

world scenarios. It focuses on the efficacy of feature selection and ensemble learning techniques. Lastly, the generalisation of the performance of the scheme in large-scale datasets due to the lack of more varied datasets, especially those with real-world network traffic, may be explored further.

Future research projects should focus on large-scale deployment and testing in various network environments to address these challenges and provide a more comprehensive understanding of the resilience of the FAMS framework against various DDoS attacks. Furthermore, a computational and resource utilisation analysis would provide information about how well the proposed approach can be adapted to real-world scenarios. Moreover, incorporating a more diverse range of datasets, particularly those sourced from actual attack situations, would make the model more relevant and improve its efficacy in real-world deployment settings.

Active research is being conducted in this field owing to the constant requirement for network technology security enhancements. We examine the most recent research on the advancement of feature-selection methods. Numerous studies have investigated feature selection strategies for DDoS attack detection.

In this work [7], the authors proposed an anomaly-based intrusion detection system to address the effects of DDoS attacks in the Ryu controller-based software-defined network. The approach involves feature extraction and classification. Training and testing of the models were done using 7 features extracted from a dataset created from live traffic. Support vector machine was found to be the best-performing technique. A framework to extract features from the southbound traffic of SDN is presented. The extracted features resulted in accurate training and testing of the models. We propose a feature extraction method for the MEC.

3. Method

This section describes the feature selection technique that was used. Given the higher dimensionality of the dataset that we utilised, the feature selection approach was used to reduce the high dimensionality of the data to allow for fast training and testing of the machine learning algorithm. To design the best solution, the FA was integrated with quasi-opposite-based learning.

High dimensionality, noisy, irrelevant, and redundant data all contribute to slower and more complex model learning. Consequently, the most important representation technique is feature selection. In machine learning and statistics, it is also known as variable, attribute, or variable subset selection. This process removes insignificant and duplicate data. This technique improves the model performance, enhances the predictive accuracy, and makes it more comprehensible.

Feature selection methods have been used in a wide range of applications, including statistical pattern recognition, machine learning, and data mining for data reduction [8][9], [10]. Existing feature selection methods are broadly classified into filter, wrapper, and embedded approaches. Filter methods categorise attributes based on the inherent information of the data [11]. Wrapper feature selection methods generate numerous models with various subsets of input features and then choose the features that lead to improved performance of the model based on a given performance metric. During training, embedded approaches perform automatic feature selection.

3.1. Firefly Algorithm

Swarm intelligence (SI) is an artificial intelligence (AI) technique based on collective habits in distributed, self-organised systems, and it is typically composed of agents that interact with one another in an environment. It is based on the observation of the natural behavior of insects and birds. These exhibit a level of collective intelligence that is greater than the intelligence of each insect or bird – for example, when looking for food, fleeing from predators, or mating. Although SI cannot provide a good solution to a problem, it can provide an optimal solution faster.

The FA is a swarm intelligence algorithm proposed in [12]. It is inspired by the behavior of fireflies, particularly their attraction to one another owing to their brightness. The brighter the firefly, the more visible it is. The brightness of a firefly reflects the value of the objective function $f(y)$. The brightness of a firefly can be minimised at a specific location, which is denoted by y, which can be calculated using the following method:

$$G(y) = \begin{cases} \frac{1}{f(y)}, & if\ f(y) > 0 \\ 1 + |f(y)|, & otherwise \end{cases} \quad (1)$$

The fitness function value is represented by Equation (1). $f(y)$ indicates the value of the objective function at point *y*, while *G(y)* denotes the fitness of the firefly. Equation (2) shows the decrease in brightness or reduction in attraction as the distance from the light source increases.

$$G(r) = \left(\frac{G_0}{1 + \gamma^{r^2}}\right) \quad (2)$$

where *r* is the distance between any two fireflies, G(r) is the light intensity, and G_0 is the intensity of the light at the source's position. γ is the light absorption coefficient. It may be estimated by considering the combined light absorption and the inverse square law for distance, which is as follows:

$$G(r) = G_0 e^{-\gamma r^2} \quad (3)$$

Similarly, a firefly's attractiveness can be characterised as follows:

$$H(r) = H_0 e^{-\gamma r^2} \quad (4)$$

H_0 is the attractiveness at *r*=0. For practical purposes, this equation is substituted with:

$$H(r) = \frac{H_0}{1 + \gamma r^2} \quad (5)$$

Moving a firefly located at *y'* to a brighter one located at *y*, the updated position of the firefly at *y'* is:

$$y' = y' + H_0 e^{-\gamma r^2}(y - y') + \alpha\beta \quad (6)$$

Where α is the randomisation parameter with $0 \leq \alpha \leq 1$, and β is a random vector.

Algorithm 1 is the standard Firefly Algorithm used as the baseline in this study. It begins by generating a population of random solutions (fireflies), evaluates their brightness (fitness), and iteratively moves each firefly toward brighter (better) ones using distance-dependent attractiveness combined with a randomisation component; if no brighter firefly exists, the firefly moves randomly. The process repeats until a termination criterion (e.g., maximum iterations or convergence) is met, making it suitable for global optimisation but prone to premature convergence in complex feature spaces.

Algorithm 1: Firefly Algorithm

1: Create a set of random solutions, $\{y_1, y_2, \ldots, y_k\}$.
2: Calculate the brightness for each solution member, $\{H_1, H_2, \ldots, H_k\}$.
3: **for** each firefly i **do**
4: Move firefly i towards other brighter fireflies.
5: If there are no brighter ones, move it randomly.
6: **end for**
7: Update the solution set.
8: **if** termination requirement is met **then**
9: Terminate.
10: **else**
11: Return to step 2.
12: **end if**

3.2. Quasi-opposite-based learning

FA is also susceptible to shortfalls that are common to other swarm intelligence algorithms. In [13], the authors found through simulation that in the early stages of iteration, the original FA may be stuck in the sub-optimal domain. This is caused by inefficient exploration, which can cause poor-quality results.

The exploration procedure is not incorporated in the original FA, and the diversification is determined by α. The convergence of the algorithm to the optimal solution is determined by the value of α [14]. Due to the application of a better exploration mechanism, even if the algorithm does not converge to the expected solution, a satisfactory solution can still be found by FA. The lack of thorough evaluation of the original FA can be addressed by the use of the best initialisation mechanisms. In this method, the probability of the initial population being closer to the optimal solution is high. Hence, we propose the use of the QOBL initialisation strategy. The QOBL was proposed in [15] as an enhancement of opposite-based differential evolution (ODE). Opposite numbers to the initial population were used to accelerate the differential evolution. Hence, in QOBL, the study utilised quasi-opposite points to accelerate differential evolution. In [16], QOBL was applied to solve the reactive dispatch

problem of the power system, and the experimental results demonstrated that QOBL performs better compared to other evolutionary methods. The quasi-opposite points $Y_{k,J}^{qol}$ of $Y_{k,j}^{l}$ Can be defined as:

$$Y_{k,J}^{qol} = rnd\left(\frac{u_j + l_j}{2}, Y_{k,J}^{ol}\right) \quad (7)$$

$$Y_{k,J}^{ol} = u_j + l_j - Y_{k,j}^{l} \quad (8)$$

Where j^{th} lower and upper values are represented by u_j and l_j respectively. k= {1,2, ⋯, Z} for a population of size Z, j= {1,2, ⋯, D} for the dimensional problem D and l represent the number of maximum iterations.

For each solution $Y_i$found in the initial population, opposite and quasi-opposite points are formed by using Equations (8) and (7), respectively. After calculating the quasi-opposite population (QOP_0), a new population is created from the initial population P_0 union QOP_0. Select from P_0 U $QOP_0$the fittest values as the initial population. The fittest values were considered the initial population of FA.

3.3. Proposed Feature Selection

The proposed feature selection algorithm is a modification of FA by using QOBL, named Quasi-Opposite Firefly algorithm (QOFA). The proposed algorithm is presented in Algorithm 2.

Algorithm 2 is the novel contribution of the paper, enhancing the traditional Firefly Algorithm with Quasi-Opposite-Based Learning (QOBL). After generating the initial random population, QOFA computes the quasi-opposite point for each solution, merges both sets, and selects the fittest individuals as the starting population to accelerate convergence and improve diversity. The remaining steps follow the conventional firefly movement and updating rules, resulting in superior exploration, reduced risk of local optima trapping, and significantly better feature selection performance for DDoS detection in mobile edge computing environments.

Algorithm 2: QOFA

1: Generate the initial population of fireflies Y = {y_i}, (i = 1, 2, 3, . . . , Z) where each firefly represents a potential feature subset.

2: Each firefly is encoded as a binary vector, indicating the selected features (1: selected, 0: not selected).

3: The intensity of light Gi at position y_i is defined by G(r).

4: Define the coefficient of light absorption γ.

5: Define the maximum number of iterations max.

6: For each firefly y_i, generate a quasi-opposite solution $Y_{k,J}^{qol}$ and create a quasi-opposite population $(QOP)_0$.

7: Generate union $P_0 \cup (QOP)_0$, sort all individuals according to their fitness, and select Z solutions.

8: **while** a < max **do**

9: **for** i = 1 to K **do**

10: **for** j = 1 to x **do**

11: **if** $G_j < G_i$ **then**

12: Move jth in the direction of an ith firefly in D dimension.

13: Change in the distance r as [−γr].

14: Replace the worst with the best solution after evaluating them.

15: **end if**

16: **end for**

17: **end for**

18: All fireflies are ranked, and the best current solution is identified.

19: end while

4. Results

4.1. Simulation results

Feature selection is essential for enhancing the model's performance and preventing overfitting. This study evaluated the overall performance of the model using both conventional and proposed feature selection techniques. The metrics considered for comparison included false positives, recall, accuracy, precision, and F1 score. We used synthetic data generated in MATLAB. The most relevant features were selected using the two algorithms. The selected data were then used to train and test the machine learning models.

First, we generated synthetic datasets that simulated both normal traffic and DDoS attack patterns. DDoS attacks are dynamic in nature. Hence, there is a need to develop up-to-date mitigation schemes. The available datasets used in the literature are outdated and have some

limitations [17]. Our synthetic generator introduces time-varying mean drifts (linear ramp ±15 % over 5 thousand samples) and intermittent spike bursts (σ = 30 for 1% of attack packets) to emulate mutated attack signatures that real botnets adopt to evade signature-based filters. These dynamics are absent in KDD'99 and UNSW-NB15, ensuring that the selected features remain discriminative against evolving non-stationary DDoS patterns. Hence, if we continue to use them for the study, there is a great probability of missing the mutated attack strategies [18]. To generate synthetic data, we first defined 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 the 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.

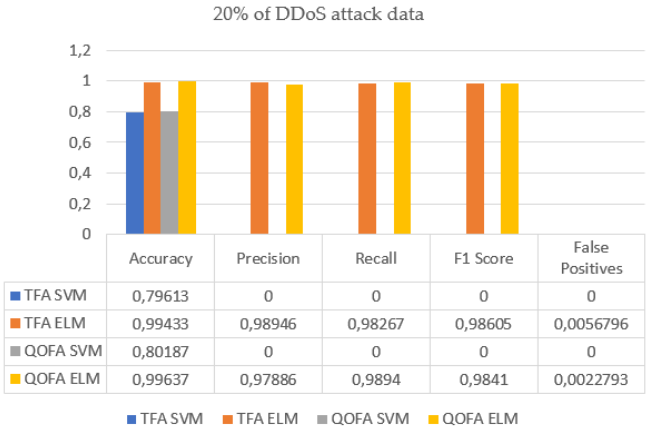
The total number of data points was 100,000, with a proportion of 0.2, 0.4, 0.6, and 0.8 allocated to DDoS attack traffic. Consequently, 80000, 60000, 40000, and 20000 data points represented normal traffic. For each of the 30 features, normal traffic data were generated using a normal distribution with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

These two datasets were then combined into a single matrix, 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 traditional FA and Quasi-Opposite FA in selecting relevant features for network traffic classification. The generated data ensured a balanced representation of normal and attack traffic, which is crucial for the reliable performance evaluation of feature selection and classification algorithms.

This study focused on feature selection techniques. A set of graphs generated from MATLAB simulations illustrating the algorithm's performance is presented in Figures 1 to 8.

The results of the simulation of the two models, ELM and support vector machine (SVM), are presented in this section. Based on the study conducted in [19] SVM was selected because SVM outperforms Logistic Regression, Random Forest, Random Tree, and Decision Tree. To evaluate the effectiveness of the proposed feature selection method, the SVM and ELM were evaluated in the traditional and proposed FA. Figure 1 presents the DDoS attack results.

Figure 1 shows that, as compared to TFA, QOFA improves the efficiency of ML algorithms. The accuracy of ELM and SVM is improved under QOFA. ELM under QOFA achieved higher accuracy (0.99637) and recall of 0.9841 compared to TFA, which had an accuracy of 0.99433 and recall of 0.98265 while maintaining comparable precision (0.97886) and F1 score (0.9841). QOFA enhances the likelihood of converging to a near-global optimum by injecting quasi-oppositional jumps that expand the early search space and reduce premature stagnation in local optima. We can conclude that in a dataset consisting of 20% of data generated by a DDoS attacker, QOFA applied to ELM is effective in detecting the true positives and in reducing false positives, leading to improved performance. Interestingly, SVM models did not perform well in detecting true positives, resulting in undefined accuracy, recall, and F1 scores. Still, this highlights how feature selection techniques in cybersecurity could benefit from the use of QOFA. Figure 2 demonstrates the performance of the algorithm in a network with 40% of data generated by a DDoS attack.



	Accuracy	Precision	Recall	F1 Score	False Positives
TFA SVM	0,79613	0	0	0	0
TFA ELM	0,99433	0,98946	0,98267	0,98605	0,0056796
QOFA SVM	0,80187	0	0	0	0
QOFA ELM	0,99637	0,97886	0,9894	0,9841	0,0022793

Figure 1. Performance of feature selection algorithms on 20% DDoS attack traffic.

Figure 2 shows the performance of two machine learning models, ELM and SVM, optimized using two variants of the FA, QOFA and TFA. For the QOFA-ELM model, the accuracy is high at 0.98393, which shows its strong performance in prediction. The F1 score is 0.9695, reflecting a better balance between precision and recall. With false positives at 0.0126283 and a precision of 0.95651, this model demonstrates robust prediction capability with minimal errors. In contrast, the QOFA-SVM model has an accuracy of 0.6047. However, the F1 score and precision are not applicable (NaN), and there are no false positives, which suggests potential model performance issues.

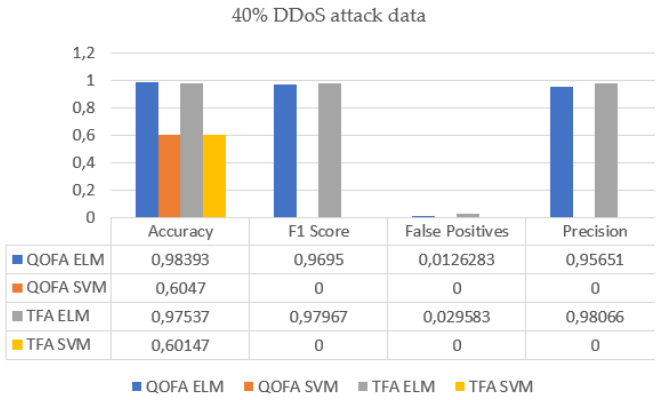


Figure 2. Performance of feature selection algorithms on 40% DDoS attack traffic.

The TFA-ELM model achieved a slightly lower accuracy of 0.97537 compared to QOFA-ELM. It has an F1 score of 0.97967 and a precision of 0.98066, indicating strong predictive capabilities, albeit with a higher rate of false positives at 0.029583. Lastly, the TFA-SVM model has an accuracy of 0.60147, like QOFA-SVM. The F1 score and precision are again not applicable (NaN), and there are no false positives, which may indicate similar performance issues as the QOFA-SVM model. The undefined metrics for SVM models indicate that these models, when using the features selected by either FA variant, consistently failed to classify any instances as DDoS attacks (i.e., zero true positives). This suggests a fundamental inability of SVM to detect the positive class under these experimental conditions, possibly due to the characteristics of the synthetic data or the selected feature space being unsuitable for SVM's decision boundaries.

In summary, the QOFA-ELM model exhibits the best overall performance with the highest accuracy, high precision, and F1 score, whereas the SVM model underperforms with potential performance issues due to model limitations. Since the QOFA approach provided pertinent dataset extracts, ELM performed best. Additionally, ELM is usually faster to train and test and is suitable for larger datasets.

Figure 3 presents the performance of the proposed algorithm in a network where 60% of the data is generated by malicious data. Figure shows that in a network 60% of the data generated is DDoS attacks. This shows a significant difference in the performance of QOFA and TFA feature selection. When integrated with the ELM model, QOFA obtains high accuracy (99.36%) and precision (99.07%); however, when integrated with SVM, it degrades and achieves lower accuracy (60.00%) and precision (60.00%). Comparatively, TFA does well with ELM, obtaining accuracy and precision of 98.29% and 99.11% respectively, while the results of SVM are poor with accuracy and precision of 60.37% and 60.37% respectively. These findings imply that ELM performs better and is an efficient model for the TFA and QOFA algorithms. In every case that was taken into consideration, QOFA performs better than TFA because it chose the best characteristics, which results in strong performance. Figure 4 compares the performance of the proposed scheme and the traditional model.

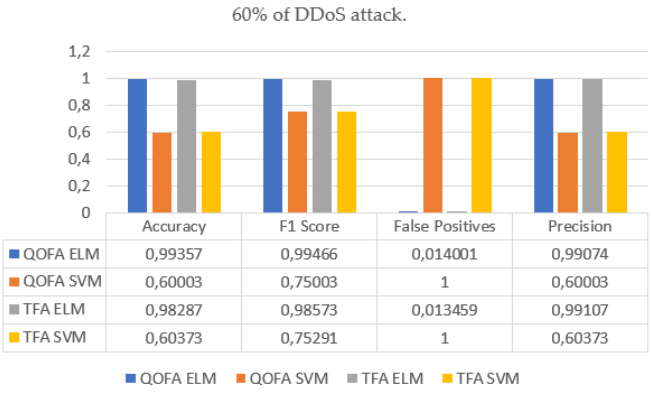


Figure 3. Performance of feature selection algorithms on 60% DDoS attack traffic.

Figure 4 displays the performance of the two algorithms, TFA and QOFA, when integrated with two models, SVM and ELM, in a scenario where 80% of the network data was generated by DDoS attacks. The ELM model outperformed SVM in both algorithms, achieving higher accuracy, precision, recall, and F1 scores. Specifically, TFA-ELM reaches an accuracy of 98.52%, precision of 98.44%, recall of 99.72%, and F1 score of 99.08%, while TFA-SVM achieves 79.90% accuracy, precision, and recall, with an F1 score of 88.83%.

Similarly, QOFA-ELM achieves 99.39% accuracy, 99.31% precision, 99.94% recall, and 99.62% F1 score compared to QOFA-SVM's 80.09% accuracy, precision, and recall, with an F1 score of 88.94%. The ELM model also reduces false positives, with 6.25% for TFA and 2.80% for QOFA, compared to SVM's 100% false positives for both algorithms.

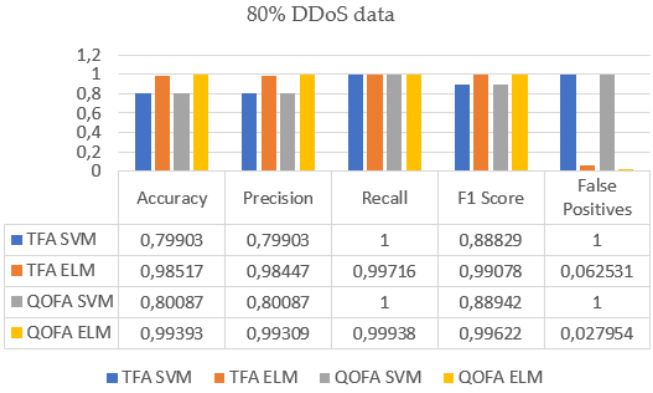


Figure 4. Performance of feature selection algorithms on 80% DDoS attack traffic.

SVM models consistently yield undefined or near-chance metrics (accuracy ≈ 0.60- 0.80, NaN precision/recall) across all DDoS proportions. This stems from two interacting factors:

- Feature-scale sensitivity – QOFA/TFA select a compact subset (≈ 8–12 features) that is highly correlated and non-linearly separable in the original input space. SVM with a linear or RBF kernel requires careful hyper-parameter tuning (C, γ) and explicit scaling, which was fixed at default values (C = 1, γ = auto) for fair comparison with ELM.
- Class-imbalance amplification – At low DDoS ratios (≤ 20%), the minority attack class is under-represented after feature reduction, pushing SVM decision boundaries toward the majority class. ELM's random hidden-layer mapping implicitly regularizes this effect. Future work will include kernel-specific re-optimization and class-weighted SVM to isolate whether the failure is algorithmic or merely configurational

In Figures 5 – 9, we present each metric in a separate graph. Figure 5 shows the performance of TFA and QOFA in various network scenarios. Figure 5 presents the accuracy of the models under different network scenarios. QOFA-ELM outperformed TFA-ELM in all scenarios except at 30% and 90%. Model accuracy increased because of QOAF's enhanced feature space exploration, which extracted pertinent, nonredundant features and reduced noise. At 30 % and 90 % DDoS traffic, QOFA selects highly stable feature subsets, resulting in marginal accuracy gains over TFA, which exhibits higher subset fluctuation. Hence, under these observations, QOFA can be selected and used in machine learning than TFA. Figure 6 shows how the feature selection methods impact the machine learning model's accuracy.

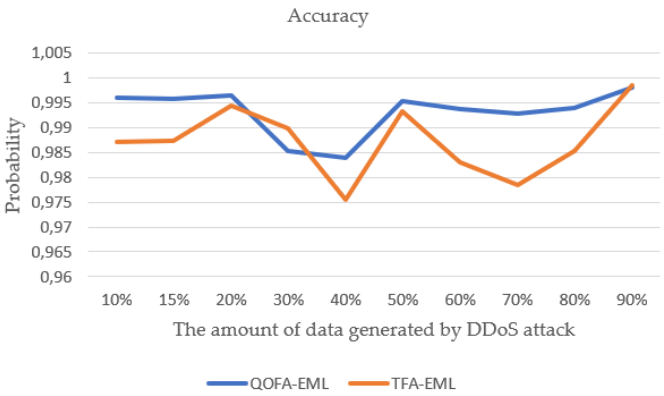


Figure 5. Accuracy of the algorithm in different network scenarios.

Figure 6 presents the comparative results of the precision of TFA-ELM and QOFA-ELM in different network scenarios ranging from 10% to 90%. The precision values for both methods vary, but certain trends can be observed. For instance, at the 10% threshold, QOFA-ELM has higher precision (0.98323) compared to TFA-ELM (0.94324). As the threshold of DDoS attack data increases to 15%, the precision of QOFA-ELM improves to (0.99459), which is higher than TFA-ELM (0.93349). At the 20% threshold, TFA-ELM has a sharp increase in precision (0.98946), which is more than QOFA-ELM (0.97886).

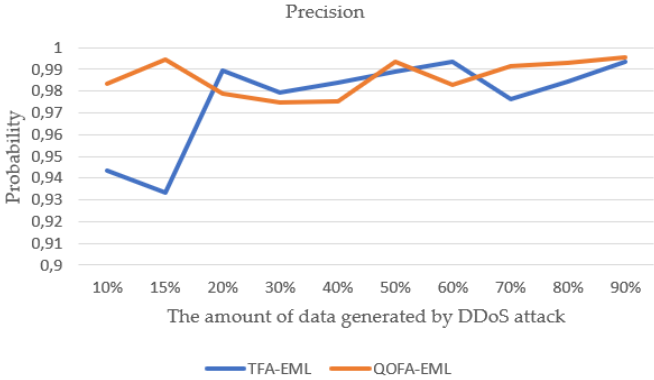


Figure 6. Precision of the algorithm in different network scenarios.

This trend continues with fluctuating results in higher thresholds, where both methods exhibit comparable performance at certain points. Notably, QOFA-ELM maintains relatively consistent high precision at 70% (0.99147) and 90% (0.9953) thresholds, whereas TFA-ELM also achieves high precision but with more variability. Overall, both methods demonstrate high precision, but QOFA-ELM exhibits more consistent superiority across the majority of tested scenarios compared to TFA-ELM. TFA-ELM slightly outperforms QOFA-ELM in precision at the 20% threshold (0.989 vs. 0.979) under moderate attacks, but QOFA dominates at higher loads due to quasi-oppositional diversification. The effect of feature selection techniques under investigation on the recall performance of the machine learning models is depicted in Figure 7.

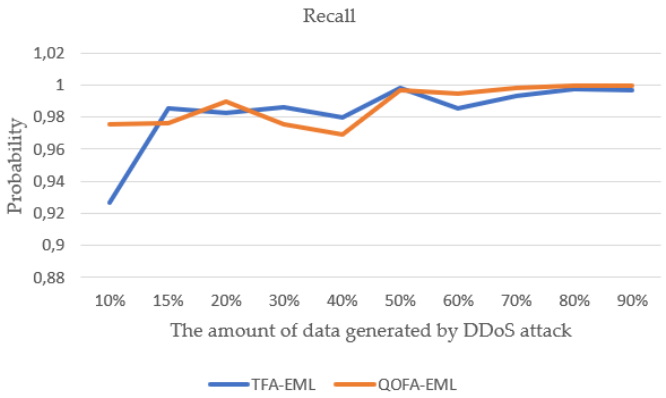


Figure 7. Recall results of the algorithms in different network scenarios.

Figure 7 displays the performance of TFA-ELM and QOFA-ELM in terms of recall on various network scenarios, with QOFA-ELM slightly outperforming TFA-ELM in the network with higher levels of data generated by DDoS attacks. Both sets of values increase as the amount of DDoS attack data rises, indicating positive skewness. The data points for both TFA-ELM and QOFA-ELM converge towards 1 at higher percentage levels (50% and above) have high consistency and reliability. While TFA-ELM and QOFA-ELM exhibit similar patterns, QOFA-ELM achieves slightly higher values, particularly at the 80% and 90% levels, which is a marginal higher and significant. Overall, the figure highlights the comparable performance of TFA-ELM and QOFA-ELM, with QOFA-ELM achieving a marginally higher performance at higher percentage levels. The effect of feature selection techniques under investigation on the F1-score of the machine learning models is depicted in Figure 8.

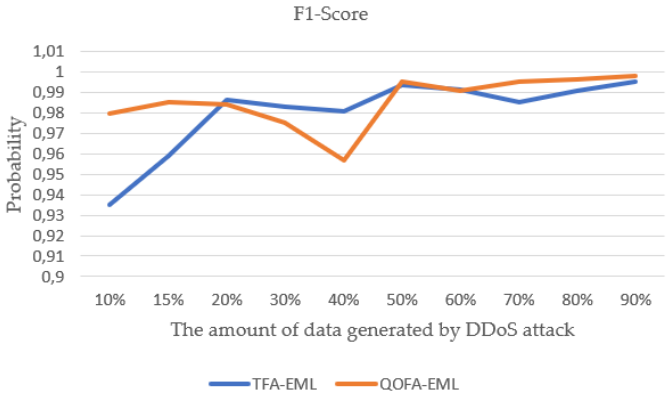


Figure 8. F1-Score results of the algorithms in different network scenarios.

The F1-Score results for both TFA-ELM and QOFA-ELM simulated under different network scenarios are presented in Figure 8. Both models obtained high F1-Scores, indicating high precision and recall. QOFA-ELM outperforms TFA-ELM, with a higher margin at higher percentage levels of data generated by DDoS attackers (80% and 90%). The F1-Scores for QOFA-ELM range from 0.9752 to 0.99761, which is a strong performance in all levels. TFA-ELM's F1-Scores range from 0.93487 to 0.99505, showing a slight dip in performance at lower percentage levels of data generated by DDoS attacks (10% and 15%). The results suggest that QOFA-ELM is a better-performing model, with a more significant advantage at higher percentage levels, while TFA-ELM still exhibits strong performance. The effect of the feature selection techniques under investigation on the false positive rate of the machine learning models is depicted in Figure 9.

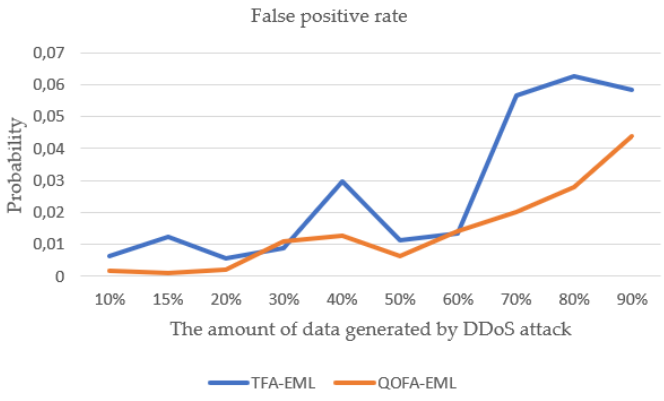


Figure 9. False positive results of the algorithms in different network scenarios.

The analysis of false positive rates for TFA-ELM and QOFA-ELM in various network scenarios is presented in Figure 9. The figure shows a consistent trend where QOFA-ELM outperformed TFA-ELM. Specifically, QOFA-ELM maintains a lower false positive rate in lower network scenarios (10%-20%), with 0.0018522 at 10% compared to TFA-ELM's 0.0062252. As the network scenarios where the percentage of DDoS attack data increases, both algorithms show a rise in false positive rates. However, TFA-ELM's increase is steeper. For example, 0.062531 at 80% and 0.05823 at 90%, whereas QOFA-ELM's rates are lower at 0.027954 and 0.043971, respectively. This pattern indicates that QOFA-ELM is more reliable and accurate, and it also minimises false alarms in different network scenarios. This depicts its superior performance in detecting true positives from false positives compared to TFA-ELM.

4.2. Why SVM Failed Across All Experiments

One surprising result was how poorly SVM performed across almost all experiments; it often detected zero attacks, giving undefined or near-random scores. The reason became clear: both TFA and QOFA pick very small, highly correlated feature sets (8–12 features. An out-of-the-box SVM (default C = 1, γ = auto, no scaling) simply can't cope with that kind of reduced, non-linear space, especially when attack traffic is low. The already small attack class almost disappears after feature selection, so SVM just predicts "everything is normal". ELM, on the other hand, is a random hidden layer, and a direct solution makes it naturally robust to these issues. So, the same excellent feature subsets that push ELM to near-perfect performance completely break an untuned SVM.

4.3. Analytical Modelling

The simulations can be validated using a variety of techniques, such as analytical modelling. Mathematical, statistical and computational methods are applied in analytical modelling to analyse and comprehend system behaviours [20]. Several analytical modelling tools, including probability theory, differential equations, linear algebra, and optimisation techniques, are employed in the literature [21]. We examined the behaviour of the two FA variants, TFA and QOFA, using probability theory. From a statistical perspective, it is not convincing to evaluate and draw conclusions about the efficiency of the two algorithms based solely on the metrics used. It is critical to test if there is any significance in their performance. Figures 1 to 9 were generated to present the simulation results. Subsequently, in this section, we utilised a t-test to model the link between the performance of the FA variants. The following assumptions were made:

1. There is homogeneity of standard deviation
2. The model's matrices are normally distributed.

Hypothesis:

H_0: QOFA-ELM and TFA-ELM perform the same

H_1: QOFA-ELM and TFA-ELM perform differently.

Rejection Region:

If t Stat > t Critical two-tailed value, we reject H_0 using a significance level of 0.05; if not, we accept H_0. This means that there is a significant difference between the performance of the two algorithms. In this study, QOFA performed better than TFA when integrated with ELM in the detection of DDoS attacks. We analyse the performance of the algorithms on varying metrics. On each metric, the instances are derived from network scenarios with varying amounts of data generated by DDoS attackers. This analysis is crucial to investigate the performance of the two algorithms.

The analytical results presented in this section are based on Figures 5 - 9, where data were collected at 10%,15%, 20%, 30%, 40%, 50%, 60%, 70%, 80% and 90% of the data was generated by DDoS attackers. The data set was utilised to conduct t-test statistics, and the results are presented in Tables 1 to 5. The primary goal of this investigation is to determine how the algorithms impact the probability of false alarms, F1 score, recall, accuracy, and precision in ELM. Table 1 presents the t-test results of the accuracy of ELM trained and tested using the dataset generated by two different feature selection techniques.

In Table 1, we observe that the t-stat is greater than the t-critical (two-tailed) value in terms of the accuracy. This provides sufficient evidence that QOFA outperforms TFA. Generally, QOFA performed much better than ELM, which recorded a 0.993047 average in a network with different scenarios. QOFA lowers the risks of overfitting and computational complexity by reducing

redundant or unnecessary features. This enables ELM to achieve faster training and improve generalisation on unknown data, particularly in dynamic situations or large datasets.

Table 1. Accuracy.

t-Test Two-Sample Assuming Equal Variances		
	QOFA-ELM	TFA-ELM
Mean	0,993047	0,987203
stand deviation	0,004731246	0,0071545
Variance	2,23847E-05	5,1187E-05
Observations	10	10
t Stat	2,154542673	
t-Critical (two-tailed)	0,044991052	

In Table 2, the t-test results for the precision of the ELM under two different feature selection techniques are presented.

Table 2. Precision.

t-Test Two-Sample Assuming Equal Variances		
	QOFA-ELM	TFA-ELM
Mean	0,976623	0,986302
stand deviation	0,021051	0,008228
Variance	0,000443	6,77E-05
Observations	10	10
t Stat	1,354203684	
t-Critical (two-tailed)	0,192429371	

Table 2 depicts that the T-value is more than the p-value for precision results, which shows that there is a significant difference between the precision obtained when QOFA and TFA were integrated with ELM. The features selected by QOFA are of better quality, which improved the training and testing of the model. This proves that QOFA performs better than TFA. This result confirms Figure 6 results, in which it was superior in seven out of ten network scenarios. The t-test results of ELM's recall performance are shown in Table 3.

Table 3. Recall.

t-Test Two-Sample Assuming Equal Variances		
	TFA-ELM	QOFA-ELM
Mean	0,983168	0,98756
stand deviation	0,020892	0,011939559
Variance	0,000436	0,000142553
Observations	10	10
t Stat	0,577174752	
t-Critical (two-tailed)	0,570970578	

The statistics in Table 3 show a comparison of the Recall means of QOFA and TFA. The mean of TFA-ELM (0.983168) is slightly lower than the mean of QOFA-ELM (0.98756). The standard deviation of TFA-ELM (0.020892316) is higher than the standard deviation of QOFA-ELM (0.011939559), which shows variability in the TFA-ELM. The variance of TFA-ELM (0.000436489) is also higher than the variance of QOFA-ELM (0.000142553). The t-statistic (0.577174752) is slightly greater than the critical t-value (0.570970578), indicating that the difference between the means is statistically significant. Therefore, we can conclude that there is a significant difference between the means of QOFA and TFA. Table 4 presents F1-Score t-test results.

Table 4. F1-Score.

t-Test Two-Sample Assuming Equal Variances		
	TFA-ELM	QOFA-ELM
Mean	0,979828	0,98552
stand deviation	0,01884	0,012720517
Variance	0,000355	0,000161812
Observations	10	10
t Stat	0,791822068	
t-Critical (two-tailed)	0,438775405	

The t-test results in Table 4 show the F1-Score Statistical analysis between QOFA-ELM and TFA-ELM assuming equal variances. The mean of QOFA-ELM (0.979828) is slightly lower than the mean of TFA-ELM (0.98552). The standard deviation of QOFA-ELM (0.01884) is higher than that of TFA-ELM (0.012721), which shows more variability in QOFA-ELM. The variance of QOFA-ELM (0.000355) is also higher than that of TFA-ELM (0.000162). The calculated t-statistic (0.791822068) is greater than the critical t-value (0.438775405), which shows that the difference between the means is marginal. Therefore, we can conclude that there is a slight but not statistically significant difference between the means of QOFA-ELM and TFA-ELM. Table 5 presents the t-test results of the false positive rate.

Table 5. False positive rate.

t-Test Two-Sample Assuming Equal Variances		
	TFA-ELM	QOFA-ELM
Mean	0,026509	0,014115279
stand deviation	0,023512	0,013568115
Variance	0,000553	0,000184094
Observations	10	10
t Stat	1,443759525	
t-Critical (two-tailed)	0,165986755	

The t-test results in Table 5 show a significant difference in false positive rates of TFA-ELM and QOFA-ELM. TFA-ELM exhibits a higher mean false positive rate (0.0265) compared to QOFA-ELM (0.0141), indicating a greater propensity for false positives. TFA-ELM's false positive rate shows more variability given its higher standard deviation (0.0235) and variance (0.0006) compared to QOFA-ELM's (0.0136 and 0.0002, respectively). The statistically significant difference (t-statistic = 1.4438, critical t-value = 0.1660) suggests that QOFA-ELM is more accurate and reliable in minimising false positives.

The improved optimisation of QOFA, which selects the most pertinent and non-redundant features by integrating the Firefly Algorithm with quasi-oppositional learning, enables QOFA-ELM to be superior. This increases computational efficiency, improves generalisation, and lowers noise and overfitting. By guaranteeing superior feature selection, QOFA makes it possible for ELM to perform well in a variety of settings, achieving superior results in metrics such as accuracy, precision, recall, and false positive rate.

5. Conclusions

The proposed QOFA enhances machine learning performance for detecting DDoS attacks in MEC by optimising feature selection. Integrating the Firefly Algorithm with Quasi-Opposite Differential Evolution, QOFA, overcomes traditional FA's premature convergence, reducing dimensionality and noise. Simulations show QOFA-ELM outperforms TFA-ELM in accuracy (e.g., 99.63% vs. 99.43% at 20% DDoS traffic), precision, recall, and false positive rates across 10%-90% DDoS scenarios. T-tests confirm significant improvements (e.g., accuracy t Stat = 2.1545 > t Critical = 0.045). QOFA-ELM is robust and efficient, with future work needed for real-world validation and scalability.

Although the proposed QOFA-ELM combination shows clear superiority in the experiments, the study relies on synthetic data and only two classifiers. Real-world network traffic is often noisier and more non-stationary than even our time-varying synthetic traces, so validation on recent datasets such as CIC-DDoS2019 or live MEC testbeds remains necessary. Moreover, while ELM proved remarkably robust to the very compact feature subsets produced by QOFA, default SVM consistently failed, reminding us that aggressive feature reduction must be paired with suitable classifiers. Future work will therefore focus on testing QOFA with a broader range of models, evaluating actual inference latency and power consumption on real edge devices, and comparing against newer meta-heuristics. These steps will help transform the promising simulation results into practical, deployable DDoS mitigation at the mobile edge.

Competing Interest Statement

The authors declare no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

Data Availability Statement

Supplementary materials and data used in this research are accessible upon request. For access, please contact the corresponding author via *sekgoari.mapunya@ul.ac.za*

Author Contributions

The study was completed by the joint efforts of the authors, whose contributions are outlined below:

Sekgoari Semaka Mapunya: Conceptualization, Methodology, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. Mthulisi Velempini: Supervision, Validation, Writing – review & editing.

Statement on the Ethical Use of AI Tools

AI tools were used for proofreading and improving linguistic quality. All corrections were manually reviewed and approved by the authors, who take full responsibility for the content.

References

[1] M. Mahbub and B. Barua, “Joint energy and latency-sensitive computation and communication resource allocation for multi-access edge computing in a two-tier 5G HetNet,” *International Journal of Information*

Technology (Singapore), vol. 15, no. 1, pp. 457–464, Jan. 2023, doi: 10.1007/s41870-022-01037-1.

[2] A. J. Ibrahim, S. R. Répás, and N. Bektaş, "Feature-Optimized Machine Learning Approaches for Enhanced DDoS Attack Detection and Mitigation," *Computers*, vol. 14, no. 11, pp. 1–33, Nov. 2025, doi: 10.3390/computers14110472.

[3] R. Liu *et al.*, "A Novel Adaptive Sand Cat Swarm Optimization Algorithm for Feature Selection and Global Optimization," *Biomimetics*, vol. 9, no. 11, Nov. 2024, doi: 10.3390/biomimetics9110701.

[4] D. W. Aha and R. L. Bankert, "A Comparative Evaluation of Sequential Feature Selection Algorithms," Doug Fisher and Hans-J. Lenz, Eds., New York: Springer, 1996, pp. 199–206. doi: https://doi.org/10.1007/978-1-4612-2404-4_19.

[5] M. Ozkan-Okay, R. Samet, Ö. Aslan, S. Kosunalp, T. Iliev, and I. Stoyanov, "A Novel Feature Selection Approach to Classify Intrusion Attacks in Network Communications," *Applied Sciences (Switzerland)*, vol. 13, no. 19, Oct. 2023, doi: 10.3390/app131911067.

[6] R. Ma, X. Chen, and R. Zhai, "A DDoS Attack Detection Method Based on Natural Selection of Features and Models," *Electronics (Switzerland)*, vol. 12, no. 4, Feb. 2023, doi: 10.3390/electronics12041059.

[7] R. K. Chouhan, M. Atulkar, and N. K. Nagwani, "A framework to detect DDoS attack in Ryu controller based software defined networks using feature extraction and classification," *Applied Intelligence*, vol. 53, no. 4, pp. 4268–4288, Feb. 2023, doi: 10.1007/s10489-022-03565-6.

[8] W. Xie, L. Wang, K. Yu, T. Shi, and W. Li, "Improved multi-layer binary firefly algorithm for optimizing feature selection and classification of microarray data," *Biomed Signal Process Control*, vol. 79, Jan. 2023, doi: 10.1016/j.bspc.2022.104080.

[9] L. N. Driff and H. Drias, "Fuzzy improved firefly-based MapReduce for association rule mining," *International Journal of Innovative Computing and Applications*, vol. 14, no. 1/2, p. 104, 2023, doi: 10.1504/IJICA.2023.129376.

[10] M. Karthikeyan, D. Manimegalai, and K. RajaGopal, "Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection," *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-50554-x.

[11] A. L. Bluma and P. Langley, "Artificial Intelligence Selection of relevant features and examples in machine," 1997.

[12] X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," 2009.

[13] J. Liu, Y. Mao, X. Liu, and Y. Li, "A dynamic adaptive firefly algorithm with globally orientation," *Math Comput Simul*, vol. 174, pp. 76–101, Aug. 2020, doi: 10.1016/j.matcom.2020.02.020.

[14] N. B. M. T. Ivana Strumberger, "Enhanced firefly algorithm for constrained numerical optimization," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, Donostia-San Sebastián, Spain.: IEE Xplore, Jun. 2017.

[15] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Quasi-oppositional differential evolution," in *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 2007, pp. 2229–2236. doi: 10.1109/CEC.2007.4424748.

[16] M. Basu, "Quasi-oppositional differential evolution for optimal reactive power dispatch," *International Journal of Electrical Power and Energy Systems*, vol. 78, pp. 29–40, Jun. 2016, doi: 10.1016/j.ijepes.2015.11.067.

[17] 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.

[18] 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.

[19] E. S. Chaki, S. S. Mapunya, and M. Velempini, "Evaluating the Effectiveness of Machine Learning Algorithms in Detecting Distributed Denial of Service Attacks in Mobile Edge Computing," *International Conference on Intelligent and Innovative Computing Applications*, vol. 2022, pp. 264–269, Dec. 2022, doi: 10.59200/ICONIC.2022.029.

[20] R. J. Murphy, O. J. Maclaren, and M. J. Simpson, "Implementing measurement error models with mechanistic mathematical models in a likelihood-based framework for estimation, identifiability analysis and prediction in the life sciences," Jan. 31, 2024, *Royal Society Publishing*. doi: 10.1098/rsif.2023.0402.

[21] R. Abdulkadirov, P. Lyakhov, and N. Nagornov, "Survey of Optimization Algorithms in Modern Neural Networks," Jun. 01, 2023, *MDPI*. doi: 10.3390/math11112466.