Journal of Basics and Applied Sciences Research (JOBASR)
ISSN (print): 3026-9091, ISSN (online): 1597-9962
Volume 3(6) November 2025 Special Issue
DOI: https://dx.doi.org/10.4314/jobasr.v3i6.10s

# Enhanced FOA with Dimensional Search Control and Memory-Based Strategy for Intrusion Detection Feature Selection

**Abubakar Abdulkadir[1*], Umar Iliyasu[2] & Zaharraddeen Sani[3]**
**[1]Department of Computer Science Education, Isah Kaita College of Education Dutsin-ma, P.M.B 5007, Katsina State**
**[2&3]Department Of Computer Science and IT, Faculty of Computing and Artificial Intelligence, Federal University, Dutsin-ma Katsina State, Nigeria**
[*]Corresponding Author Email: aakado123@gmail.com

**ABSTRACT**
Feature selection plays a crucial role in improving the accuracy and efficiency of Network Intrusion Detection systems (NIDS) by reducing dataset dimensionality and eliminating redundant or irrelevant features that do not contribute meaningfully to classification outcomes. The Fruit Fly Optimization Algorithm (FOA) used for feature selection and its variants blindly search the solution space which leads to an imbalance between exploration and exploitation, reduce convergence speed and stuck at local optima. In this study, an enhanced feature selection algorithm based on fruit Fly Optimization Algorithm (FOA) is proposed to improve the balance between exploitation and exploration, faster convergence and avoid staganation at local optima. The enhanced version integrates two intelligent mechanisms Dimensional Search Control (DSC) and Memory-Based Strategy (MBS) which effectively guide and regulate the search process, enabling the algorithm to identify the most relevant features more efficiently.This study contribute by eliminating or reducing the high computational complexity faced by basic FOA and other FOA metaheuristic algorithm in feature selection and reduce the number of selected features as well as increase in the accuracy of the selected features. The proposed algorithm was implemented in Google Colab using Python programming language and evaluated using standard datasets NSL-KDD and CICID2017 against several well-known metaheuristic algorithms, including SCMWOA, BIFOA, ALO, and the basic FOA. The comparison was conducted using key performance metrics such as computational complexity (execution time and memory usage), number of selected features, classification accuracy, and fitness values .Experimental results demonstrate that the proposed enhanced FOA consistently outperformed the compared algorithms across all evaluation criteria as shown in section 4.1.5 fitness values of 99.9%,!00%,across the two datasets used, Section 4.1.4 accuracy values of 100% ,across the two datasets used, Section 4.1.3 28 and 18 number of selected features in NSL-KDD and CICID2017 datasets, Section 4.1.1 execution time of 854.45s and 4025.67s in NSL-KDD and CICID2017 respectively and Section 4.1.1 memory usage of 20.04 MB and 39.07 in NSL-KDD and CICID2017 respectively . Its superior efficiency, accuracy, and scalability make it highly suitable for deployment in modern Network Intrusion Detection System (NIDS) designs.

**Keywords:**
IDS NIDS IoT,
MBS, DSC, FOA

## INTRODUCTION

Feature selection through metaheuristic algorithms has become a widely adopted strategy in computational intelligence due to its ability to identify the most informative features from large and complex datasets. This process improves model accuracy, reduces computational overhead, and enhances interpretability Wang et al., (2022). Popular metaheuristic approaches such as Particle Swarm Optimization (PSO), Fruit Fly Optimization Algorithm (FOA), and Ant Lion



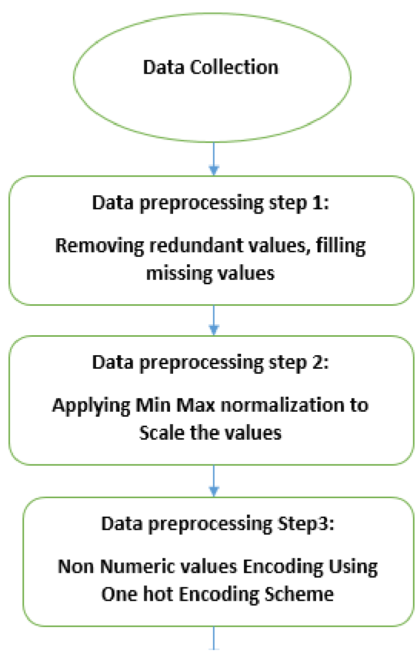
**How to cite this article:** Abubakar A., Umar I. & Zaharraddeen S. (2025). Enhanced FOA with Dimensional Search Control and Memory-Based Strategy for Intrusion Detection Feature Selection. *Journal of Basics and Applied Sciences Research*, *3*(6)s, 82-91. https://dx.doi.org/10.4314/jobasr.v3i6.10s

Optimization (ALO) have demonstrated considerable success in handling nonlinear, high-dimensional feature spaces (Louk and Tama 2022). Inspired by natural and biological processes—such as swarm behavior in PSO, sensory-driven foraging in FOA, and predatory dynamics in ALO—these algorithms employ stochastic search mechanisms to explore solution spaces efficiently Fu et al., (2022). Their increasing relevance is driven by their capability to balance exploration and exploitation, avoid local optima, and outperform traditional statistical or deterministic techniques across diverse domains including machine learning, pattern recognition, and bio informatics (Jain et al., 2022). Despite their strengths, recent studies show that many metaheuristic algorithms—including ALO, PSO, FOA, and their variants—exhibit limitations when applied to high-dimensional datasets in Network Intrusion Detection Systems (NIDS). These challenges stem from premature convergence, weak global exploration, and high computational demands, which impair their ability to consistently identify optimal feature subsets without degrading detection accuracy. Notably, FOA and its extended versions often show inconsistent performance in feature selection tasks, frequently struggling with an imbalanced exploration–exploitation process, slow convergence, and vulnerability to local optima Hou et al., (2019). These deficiencies reduce their suitability for complex IDS datasets that require robust search adaptability, scalability, and efficient dimensionality reduction. Although several FOA enhancements have been introduced—including chaotic FOA by Ye et al., (2017), Levy Flight–guided FOA by Huang et al., (2019), evolutionary population–based BFOA by Hou et al., (2019), and hybrid models like dimension-selection PSO (DSPSO) by Shami et al., (2024) , Sine Cosine–Whale Optimization Algorithm (SCMWOA) by Sayed et al., (2022), Optimized adaptive artificial neural network by Patil et al., (2022), ensemble approach proposed by Kiziloz (2020) and Feed Forword implemented by Sharma and Sing (2023) , current solutions still fall short in simultaneously achieving high classification accuracy, selecting minimal feature subsets, and maintaining low computational complexity in large-scale environments. Existing FOA variants also lack advanced mechanisms such as Dimensional Search Control (DSC) and memory-based exploration, which are essential for steering the algorithm toward promising feature dimensions and preventing repetitive exploration of unproductive regions. Furthermore, most current designs do not offer a unified framework that supports strong global search capability, stable convergence behavior, and adaptiveness to high-dimensional spaces.

To address these persistent limitations, this study proposes an enhanced FOA-based feature selection algorithm that integrates Dimensional Search Control and a memory-driven strategy. The DSC mechanism directs the search toward the most promising feature dimensions, while the memory component enables the algorithm to store, recall, and refine high-quality solutions throughout the optimization process. Together, these mechanisms significantly strengthen the exploration–exploitation balance, accelerate convergence, and minimize the likelihood of stagnation in local optima. The resulting approach is expected to deliver more compact, accurate, and computationally efficient feature subsets, making it highly suitable for modern machine learning tasks and next-generation NIDS environments.

## MATERIALS AND METHODS

The methodology adopted in this research paper comprises of a six steps workflow starting from removing all redundant and irrelevant attributes and filling in missing values and then applying the Min Max normalization techniques to normalize the data to a uniform scale for the final classification algorithm. After normalizing the datasets, the categorical values were encoded in to numeric values. Feature selection was conducted in the final step using the proposed algorithms. Before conducting the feature selection using the algorithm, new mechanisms were introduced to the basic Fruit Fly Optimization Algorithm (FOA): Dimensional Search Control (DSC) and Memory-Based Strategy (MBS) to help the algorithm improve its search performance. A comparison was conducted between the proposed algorithm and state-of-the-art optimization algorithms. The diagram below shows the workflow of the methodology framework describing each step



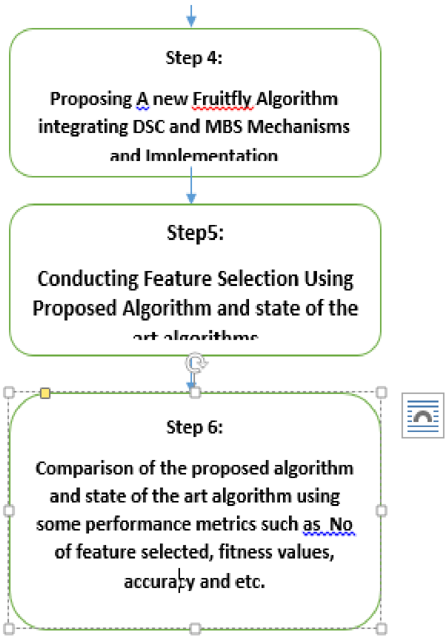



Figure.1 General Workflow or research frame work

The diagram presented above illustrates the complete research framework, which consists of six major stages ranging from dataset acquisition to the performance evaluation of the proposed algorithm against state-of-the-art techniques. In the first stage, the datasets were collected from Kaggle and stored in CSV format. These datasets then underwent a series of preprocessing procedures as outlined in the general framework. Initially, missing values and redundant attributes were identified and removed. Subsequently, Min–Max normalization was applied to scale and standardize feature values, thereby improving the effectiveness of the downstream classification models. The third stage involved transforming categorical or non-numeric attributes into numeric representations through one-hot encoding. This step ensured that all input features were compatible with the optimization and classification algorithms, which operate exclusively on numeric data. In the fourth stage, the proposed algorithm was developed by integrating the basic Fruit Fly Optimization Algorithm (FOA) with two enhancement mechanisms: Dimensional Search Control and a Memory-Based Strategy(MBS). These mechanisms were incorporated to guide the search process toward more promising solution dimensions and to retain previously visited high-quality solutions, thus preventing the algorithm from revisiting inferior search regions. The pseudocode and flowchart detailing the workflow of the enhanced algorithm are provided in the subsequent section. In the fifth stage, both the proposed algorithm and selected state-of-the-art optimization algorithms were implemented in Python within the Google Colab environment, where feature selection experiments were conducted.

Finally, the sixth stage involved a comparative analysis of all algorithms based on several performance metrics, including fitness values, number of selected features, classification accuracy achieved using the selected feature subsets, and overall computational complexity—measured in terms of execution time and memory consumption.

**Description of the Datasets**

The study utilizes two widely recognized benchmark datasets NSL-KDD and CIC-IDS2017 to ensure a comprehensive and reliable evaluation of the proposed Network Intrusion Detection System (NIDS). These datasets were selected for their ability to represent diverse network traffic patterns and attack behaviors, thereby enhancing the assessment of the model's performance and generalization capability.

The NSL-KDD dataset, introduced by Tavallaee et al. (2009) as an enhanced version of the original KDD'99 dataset, resolves key issues such as redundancy and class imbalance. It provides a more balanced and representative sample distribution, improving the credibility of model evaluation. Containing 125,973 training and 22,544 testing records, NSL-KDD supports both binary and multi-class classification (DoS, R2L, U2R, and Probe) and is lightweight (about 20 MB) for computational efficiency. It is publicly accessible via the University of New Brunswick's repository.

The CIC-IDS2017 dataset, developed by the Canadian Institute for Cybersecurity, mirrors contemporary real-world network conditions. Collected over seven days, it features normal and multiple attack types such as DDoS, Port Scanning, Brute Force, Botnet, and Web Attacks captured from realistic user interactions. Each record includes comprehensive flow-based and time-based features, making it suitable for both anomaly and signature-based detection approaches. To prepare the data, Min–Max normalization was applied to standardize feature ranges, and SMOTE (Synthetic Minority Oversampling Technique) was used to correct class imbalances between attack and normal traffic. Collectively, these preprocessing steps and dataset choices ensure that the proposed NIDS is evaluated under diverse and realistic conditions, enhancing its robustness and adaptability

**Dimensional Search Control**

The Dimensional Search Control (DSC) mechanism is employed to guide and prioritize the search process of the

Fruit Fly Optimization Algorithm (FOA) toward the most promising dimensions (variables) within the solution space. Rather than treating all variables equally during optimization, DSC enables the algorithm to concentrate more on influential dimensions while reducing the search effort in less significant ones. This selective focus enhances the algorithm's exploration–exploitation balance, promotes faster convergence, and mitigates the risk of prematurely converging to local optima. In this mechanism, each fruit fly updates its position across dimensions and explores the neighborhood of the current best solution according to the update expression provided below.

$X_{i+1}\ (d) = X_{best}\ (d) + \propto X_{rand}(d), (X_{i,1}(d) = X_{best}$ (1)

Where:

$X_o^{t+1}(d)$ *the new position of ith individual fly in dimension d at the next iteration* $(t+1)$

$X_i^t(d)$*is the current position of the i − th individual in dimension d at iteration t*

$X_{best}$ *is the position of the best solution (best fruit fly) found so far in dimension d*

$X_{rand}$ *a random number between* 0 *and* 1 *− it add randonmness or exploration ability*

$\propto$ *(Alpha)A control paramter (step size )that determines how far the algorithm explores around the best position d the current dimension (a feature or variable in the search space*. The explanation of each term in the above equation is given below:

1. $X_i^t - X_{best}(d)$ This term measures the different (or direction) between the current fly and the best fly in that dimension. It shows which way to move toward or away from the best.
2. *Multiply by* $X_{rand}$ This adds random variation, so not all flies move exactly the same helping the algorithm explore
3. *Multiply by* $\propto$ This controls the magnitude (step size) of movement- larger values explore farther, smaller values fine tune near the best solution.
4. *Add* $X_{best}(d)$ This ensures the movement is centered on the best known position, guiding the search toward better regions.

Memory Based Strategy

The memory-based strategy serves as a mechanism to enhance the overall performance and search efficiency of the Fruit Fly Optimization Algorithm (FOA). It achieves this by systematically storing, recalling, and exploiting previous search experiences, such as solution positions, fitness values, and search directions, to guide subsequent search processes more intelligently. Through this mechanism, the algorithm effectively avoids redundant exploration of inferior solutions, intensifies the search around previously identified promising regions, and maintains a balanced trade-off between exploration and exploitation. The memory update equation enables the algorithm to compare the fitness of the previously stored best solution, ( f(M_{best}) ), with that of the newly generated solution, ( f(X_i^{t+1}) ), subsequently retaining the one with the superior (i.e., smaller) fitness value as the updated ( M_{best} ). This ensures that the algorithm continuously preserves and builds upon the most optimal solution discovered throughout the optimization process.

$M_{best} = argmin(f(M_{best})f(X_{i+1})$ (2)

Where:$M_{\ best}$ *Stored best solution in the memory it remember the best position found so far*

$X_i^{t+1}$*The new candidate solution (fruifly)found at the current iteration*

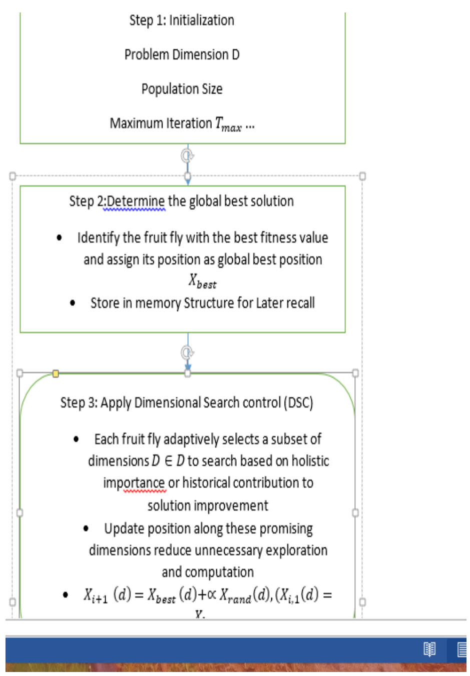
$f(.)$ *The fitness function, which measures how good each solution is*

*argmin A mathematical operator meaning " choose the arqurment (solution) that gives the smallest* fitness values.

**Enhance Fruit fly optimization algorithm with Dimensional Search control and memory based Strategy**.

The proposed Enhanced Fruit Fly Optimization Algorithm (EFOA) integrates the two improvement mechanisms Dimensional Search Control (DSC) and Memory-Based Strategy (MBS) to significantly enhance the efficiency and robustness of the traditional Fruit Fly Algorithm (FOA) and its variants. While the fundamental structure and operational flow of the proposed algorithm remain consistent with the original FOA, the incorporation of these two mechanisms introduces greater adaptability, intelligence, and optimization accuracy. In essence, the Dimensional Search Control mechanism enables the algorithm to dynamically guide its search toward the most promising dimensions or regions within the solution space. This targeted exploration helps maintain an effective balance between exploration (searching new areas) and exploitation (refining known good solutions), resulting in faster convergence and a reduced likelihood of getting trapped in local optima.

Simultaneously, the Memory-Based Strategy enhances the learning ability of the algorithm by allowing it to store, recall, and reuse information about previously discovered high-quality solutions. This memory retention prevents the algorithm from revisiting poor or unproductive regions of the search space, thereby improving search efficiency and promoting steady progress toward the global optimum. Together, these two mechanisms empower the proposed EFOA to perform a more intelligent and adaptive search process, overcoming the limitations of the basic FOA and its earlier variants. The flowchart and corresponding mathematical formulation of the proposed algorithm, presented below, illustrate the integration and operational workflow of these newly introduced mechanisms, providing a clear depiction of how the enhanced model achieves superior optimization performance..



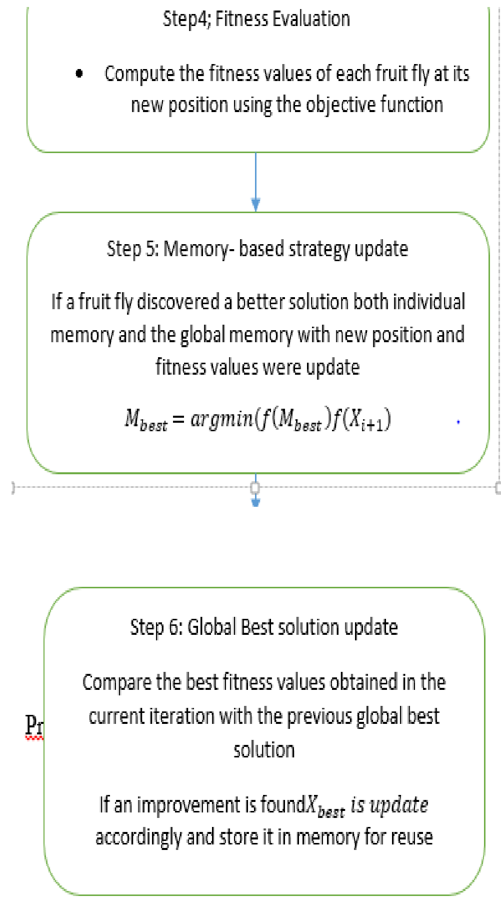




Figure 3.2 representing the algorithm flow chart diagram

## Implementation Requirements and Parameter Setting

We used KNN classifier based on Euclidean distance to measure the accuracy of the selected features by the algorithms. In this experiment different Fruit fly variants and non Fruit fly algorithm were adopted to find the one with the optimal reduction with minimal error. In each of 20 runs the algorithms iterates the datasets is randomly divided in two sets 80% of the instances are used for training, and the remaining are used for testing. This datasets partitioning was used in various previous works in the literatures (46)-(48).Note that we choose KNN because it is simple and cheap. Previous research (43) has shown that using cheap and simple classifier to assess the feature selected accuracy in a wrapper approach can select a good feature subsets for other complex learning/classification algorithm which are computationally expensive but able to achieve better classification accuracy.

However, the algorithms were implemented in a google Colab environment using python programming language with the hardware specification **of** system RAM of

12.7GB, GPU RAM of 15GB with Window 10 OS. The preprocessing of the data as well as selecting features will be done by using libraries such as Scikit-Learn and Pycaret.

## RESULTS AND DISCUSSION

This section presents and discusses the results obtained from comparing the proposed algorithm developed in this research with other existing metaheuristic algorithms for feature selection. The comparison was conducted using five key performance metrics: fitness value, classification accuracy, computational time, number of selected features, and memory utilization. These evaluation criteria collectively provide a comprehensive assessment of the efficiency, effectiveness, and scalability of the proposed approach compared to its counterparts.

### Result Based on Execution Time

This section provides a comparative analysis of the execution times of five algorithms applied to suboptimal feature selection on two benchmark intrusion detection datasets, namely NSL-KDD and CICIDS2017. The proposed Enhanced FOA, incorporating dimensional search control and a memory-based strategy, was evaluated against four established metaheuristic approaches: Basic FOA, BIFOA, and ALO. Experimental results reveal that the Enhanced FOA consistently achieved the shortest execution time across both datasets, thereby demonstrating superior computational efficiency. The reduced processing time indicates that the proposed approach is better suited for handling large-scale, high-dimensional datasets and is particularly advantageous in time-sensitive environments such as intrusion detection systems. These findings underscore the practical benefits of integrating dimensional search control and memory-based mechanisms into FOA, establishing its effectiveness over competing algorithms. The table given below presents the execution time taken by each algorithm in ten distinct runs.

Table 1: Presents Algorithm Execution Time

| Algorithm and Datasets | ALO | BIFOA | BASIC FOA | ENHANCED FOA | SCMWOA |
|---|---|---|---|---|---|
| Dataset one | 670.56s | 950.03s | 1002.00 | 854.45s | 909.62s |
| Dataset two | 4429.49s | 3898.27s | 4331.07s | 4025.67s | 2341.23s |

The corresponding figures presented below depict the distribution of execution times for each algorithm under consideration.

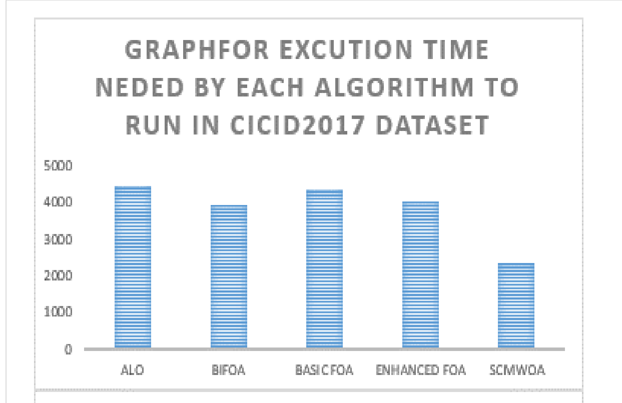



Figure 1: Execution time of four different metaheuristic algorithm for NSL-KDD dataset feature selection

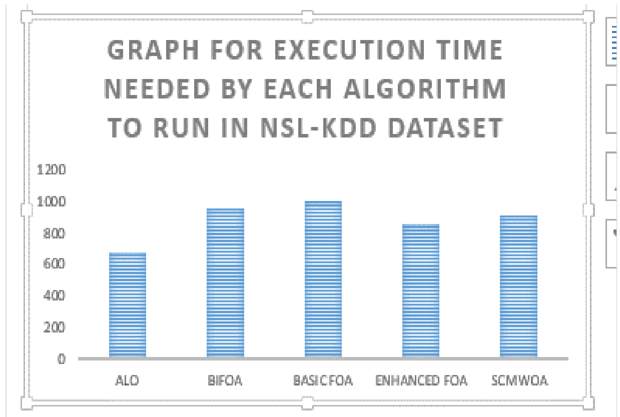



Figure 2 execution time of four different metaheuristic algorithm for CICID2017 dataset feature selection

### Result based on Memory space required

The analysis of memory consumption across the five metaheuristic algorithms for feature selection on the NSL-KDD and CICIDS2017 datasets reveals notable differences in efficiency. As shown in the graphs, both the Enhanced FOA and ALO consistently required the least memory, whereas Basic FOA, BIFOA, and other counterparts exhibited higher memory usage. This finding suggests that the Enhanced FOA, with its dimensional search control and memory-based strategy, not only improves search effectiveness but also minimizes computational overhead. The reduced memory footprint is particularly advantageous in real-world intrusion detection systems, where large-scale data must be processed under resource constraints. Consequently, the results emphasize the practicality and scalability of the Enhanced FOA compared to traditional metaheuristic approaches. The table below presents the memory usage of each algorithm in ten distinct runs.

Table 2: Presents Memory Usage by the Algorithms

| Algorithms and Datasets | ALO | BIFOA | BASIC FOA | ENHANCED FOA | SCMWOA |
|---|---|---|---|---|---|
| Dataset one | 84.83 mb | 33.61 MB | 46.23 MB | 20.04MB | 49.36 MB |
| Dataset two | 54.93 MB | 41.33 MB | 52.05 MB | 39.07MB | - 17.63 MB |

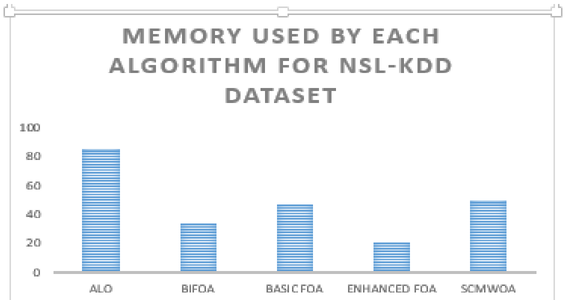



Figure 3 showing memory space required to perform suboptimal feature selection for NSL-KDD

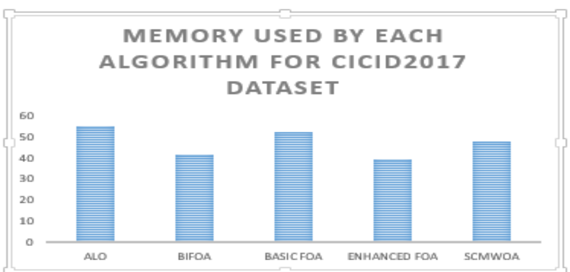
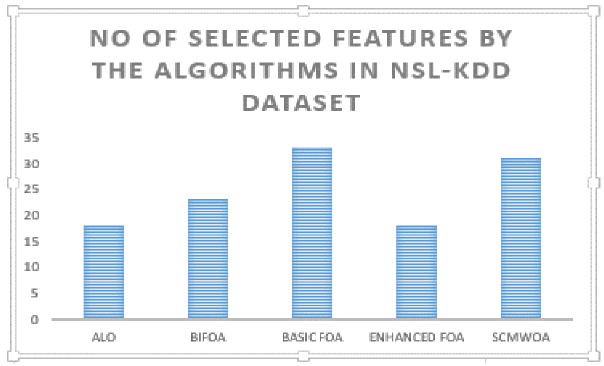



Figure 4 showing memory usage by four different algorithm in performing suboptimal feature selection for CICID2017

**Result Based on Number of Feature Selected**

This section present the result analysis of average number of features selected by each algorithm used for the two different intrusion detection datasets, as shown from the graph given below at figure 4 and 5 , it is evident that not all algorithms produced the same number of features at the end of the feature selection task. Each algorithm generated a different number of features for the two datasets (NSL-KDD and CICIDS2017). Notably, the BIFOA and Enhanced FOA produced the smallest number of features in both datasets, outperforming their counterparts by eliminating more redundant and irrelevant attributes. This indicates that these algorithms are capable of developing simpler models that allow for faster training and testing, as well as easier interpretation. When considered alongside the accuracy results for both datasets, where all algorithms achieved competitive accuracy, it can be concluded that the algorithms selecting fewer features are the better choice for feature selection. By focusing on smaller yet more informative feature subsets, these algorithms reduce the risk of overfitting—since irrelevant or noisy features often mislead classifiers—and promote better generalization. In conclusion, selecting fewer but highly discriminative features is critical for improving speed, interpretability, and cost efficiency in real-world applications such as IoT and biomedical data analysis.

Figure 5 showing different number of feature selected by five algorithm for NSL-KDD datasets




**Result Based on accuracy of the Selected Features**

This section presents the accuracy value of the five metaheuristic algorithms used for conducting feature selection task on two different intrusion detection task NSL-KDD and CICID2017 Datasets and The results demonstrate that all the algorithms achieved high accuracy values on both the NSL-KDD and CICIDS2017 datasets, confirming their ability to select generalizable feature subsets. However, as illustrated in the Figure 6 and 7, Enhanced FOA and BIFOA achieved superior performance on Dataset 2 (CICIDS2017), highlighting their effectiveness in feature selection when applied to modern intrusion detection challenges. In contrast, on Dataset 1 (NSL-KDD), the performance gap among the algorithms was less pronounced, as all methods were able to identify improved feature subsets that enhanced the final classification accuracy. These findings suggest that while all algorithms are effective in handling traditional intrusion detection datasets, Enhanced FOA and BIFOA demonstrate clear advantages in addressing the complexities of more recent and large-scale datasets. The table below presents the accuracy values obtained

Table 4: Presents the accuracy of the selected Features

| Algorithm and Datasets | ALO | BIFOA | BASIC FOA | ENHANCED FOA | SCMWOA |
|---|---|---|---|---|---|
| Dataset one | 1 | 1 | 1 | 1 | 1 |
| Dataset two | 1 | 0.9999 | 0.9999 | 0.9999 | 0.9999 |

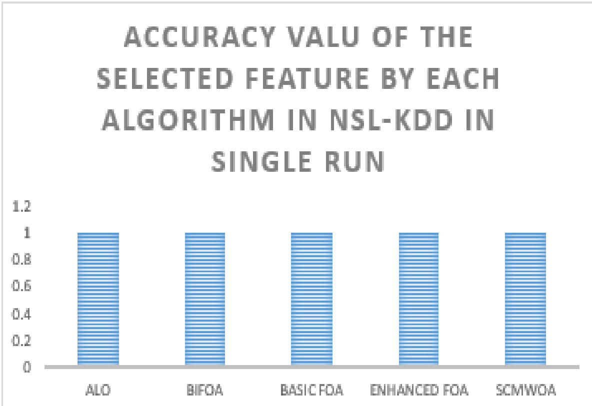



Figure 6 showing different accuracy value of the selected feature by five algorithm for NSL-KDD datasets

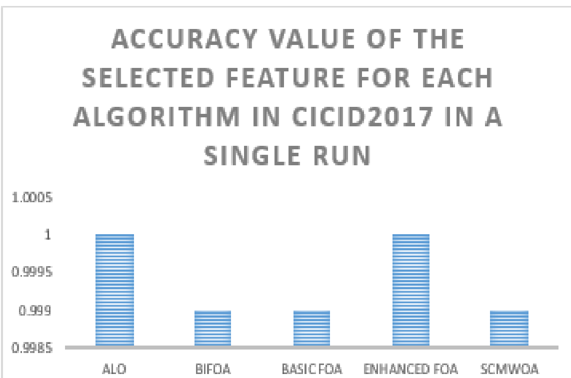



Figure 7 showing different accuracy value of the selected feature by five algorithm for CICID2017 datasets

**Result based on fitness values**

This section presents the result or fitness values of five different algorithms over different no of iteration. As shown, the results indicate that some algorithms tend to select either excessively large subsets containing redundant features or overly small subsets that reduce predictive power. A steady improvement in fitness values across iterations generally reflects an effective balance between exploration and exploitation. However, as illustrated in Figures 1 and 2, several algorithms exhibited stagnation in their fitness values, suggesting entrapment in local optima or premature convergence. By contrast, the proposed Enhanced FOA and BIFOA demonstrated more consistent improvements and slower stagnation, indicating stronger search dynamics and a better ability to avoid local optima. This comparative performance scores the advantage of these algorithms in producing higher-quality feature subsets. The table below presents the fitness values from ten runs

Table 5: Presents the Fitness values of the Algorithm

| Algorithm and Datasets | ALO | BIFOA | BASIC FOA | ENHANCED FOA | SCMWOA |
|---|---|---|---|---|---|
| Dataset one | 1 | 1 | 1 | 1 | 1 |
| Dataset two | 1 | 0.9999 | 0.9999 | 0.9999 | 0.9999 |

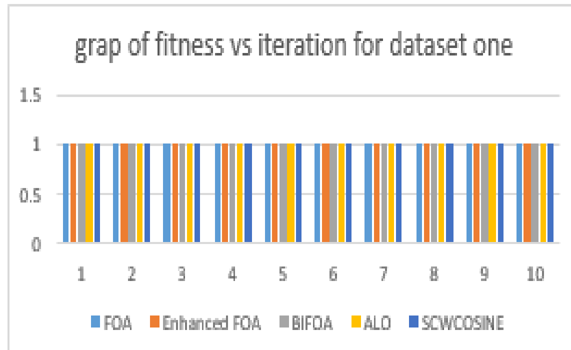



Figure 7 showing fitness values of five different algorithms across number of iterations

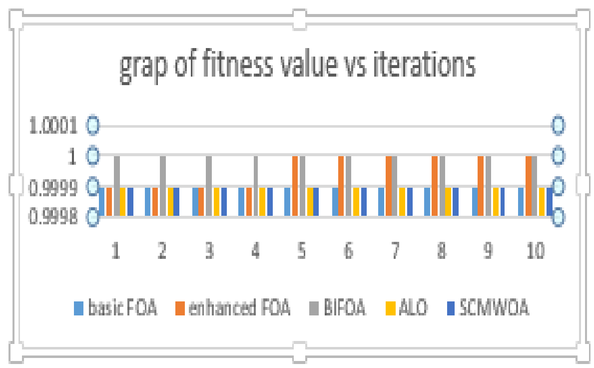



Figure 8 showing fitness values of five different algorithms across number of iterations

The comparative evaluation of five metaheuristic algorithms for feature selection on the NSL-KDD and CICIDS2017 datasets shows that the proposed Enhanced FOA, incorporating dimensional search control and a memory-based strategy, consistently outperforms its counterparts. It achieved the fastest execution times and lowest memory usage, making it highly suitable for real-time, resource-constrained environments such as IoT-based intrusion detection systems. Alongside BIFOA, it also produced smaller, more meaningful feature subsets, enabling faster training, reduced overfitting, and improved generalization. While all algorithms performed well in terms of accuracy, Enhanced FOA and BIFOA demonstrated particular effectiveness on the complex CICIDS2017 dataset, highlighting their robustness against modern intrusion challenges. Furthermore, their superior fitness dynamics and resistance to premature convergence indicate stronger exploration–exploitation balance, ensuring higher-quality feature subsets compared to other methods.

## CONCLUSION

The paper presents and proposes a new metaheuristic algorithm (Enhanced Fruit Fly algorithm Optimization Algorithm) and undergo or perform feature selection using two benchmark datasets NSL-KDD and NSL-KDD and compared the algorithm with other state of the art algorithms based on certain metrics (number of selected features, accuracy, fitness values, and time and memory complexity. The results show that the proposed algorithm **outperforms** the compared methods by achieving or obtaining high fitness value, accuracy value, few number of selected feature and low computational complexity demands and is suitable for feature selection. It is recommended that the enhanced fruit fly algorithm proposed in this research work be adopted for future intrusion detection system design and that the Enhanced FOA be applied not only for feature selection but also in other optimization-driven scientific tasks. However, is part of the limitation of the proposed Feature selection Algorithm that its design to be used in standalone intrusion detection system not been implemented and tested in real time network environment. It's considered as Future work that the designed or proposed feature selection be implemented and tested in a real life network environment.

The main contribution of this research is that the design Feature selection had reduced the number of selected feature and computational complexity when compare it with other state of the art Optimization Algorithm published in previous study. Also, the intrusion detection model design using this approach had reduced the number false negative and positive alarm usually experience with other models.

## REFERENCES

Zahra, O., Raja, M., & Pran, R. (2023). Enhancing deep learning-based NIDS: A focus on imbalanced datasets and computational efficiency. Computers & Security, 119, 102754.

Wang, D., & Xu, G. (2020). Research on the detection of network intrusion prevention with SVM-based optimization algorithm. Informatics, 44(2),179-189. https://doi.org/10.31449/inf.v44i2.3195.

Patil s., Vijayakumar Varadarajan, V., Siddiqui Mohd Mazhar S., M., Abdulwodood

Sahibzada A., Nihal Ahmed N., Onkar Sinha O., Kumar S.Ka, ilash Shaw K., and Kotecha K., Explainable Artificial Intelligence for Intrusion Detection System. https://doi.org/10.3390/electronics11193079.

Kiziloz H., E.,Tama Classifier ensemble for feature selection https://doi.org/10.1016/j.neucom.2020.07.113.

Wang, Y. C., Houng, Y. C., Chen, H. X., & Tseng, S. M. (2023). Network anomaly intrusion detection based on deep learning approach. Sensors, 23(4), 2171.

Louk, M. H., & Tama, B. A. (2022). PSO-driven feature selection and hybrid ensemble for network anomaly detection. Big Data and Cognitive Computing, 6(4), 137

Fu, G., Li, B., Yang, Y., Wei, Q., A Multi-Distance Ensemble and Feature Clustering Based Feature selection Approach for Network Intrusion Detection DOI: 10.1109/ISSI55442.2022.9963.\

B. Xue, M. Zhang, and W. N. Browne, ''A comprehensive comparison on evolutionaryfeatureselectionapproachestoclassification,'' Int.J. Comput. Intell. Appl., vol. 14, no. 2, Jun. 2015, Art. no. 1550008.

A. Areej and E. Ashraf, ''A minimal subset of features using feature selection for handwritten digit recognition,'' J. Intell. Learn. Syst. Appl., vol. 9, no. 4, p. 422, 2017.

M. Wieland and M. Pittore, ''Performance evaluation of machine learning algorithms for urban pattern recognition from multi-spectral satellite images,'' Remote Sens., vol. 6., no. 4, pp. 2912–2939, 2014.

YUN HOU, JINGYAO LI1,2, HAIHONG YU, AND ZHANSHAN LI1BIFFOA: A Novel Binary Improved Fruit Fly Algorithm for Feature Selection.

Fei Ye*, Xin Yuan Lou, Lin Fu Sun An improved chaotic fruit fly optimization based on a mutation strategy for simultaneous feature selection and parameter optimization for SVM and its applications PLoSONE 12(4): e0173516.https:// doi.org/10.1371/journal.pone.0173516.

Min Yuan and Mei Wang A feature selection method based on an improved fruit fly optimization algorithm in the process of numerical control milling Advances in

Mechanical Engineering 2018, Vol. 10(5) 1–10 The Author(s) 2018 DOI: 10.1177/1687814018778227.

Hui Huang, Xi 'an Feng, Suying Zhou, Jionghui Jiang, Huiling Chen, Yuping Li and Chengye Li A new fruit fly optimization algorithm enhanced support vector machine for diagnosis of breast cancer based on high-level features. Huang et al. BMC Bioinformatics 2019, 20(Suppl 8):290 https://doi.org/10.1186/s12859-019-2771-z.

Tareq M. Shami • Qasem Al-Tashi • Nima Khodadadi • Said Jadid Abdulkadir Abdulaziz Ahmed• Seyedali Mirjalili7Dimension selection: an innovative metaheuristic strategy for particle swarm optimization. Cluster Computing (2025)28:379 https://doi.org/10.1007/s10586-025-05201-7(0123456789().,-volV)(0123456789,-().volV).

José Barrera-García , Felipe Cisternas-Caneo , Broderick Crawford, Mariam Gómez Sánchez and Ricardo Soto Feature Selection Problem and Metaheuristics: A Systematic Literature Review about Its Formulation, Evaluation and Applications. Applications. Biomimetics 2024, 9, 9. https://doi.org/10.3390/ biomimetics9010009.