UTM UNIVERSITI TEKNOLOGI MALAYSIA
INTERNATIONAL JOURNAL OF INNOVATIVE COMPUTING
ISSN 2180-4370
Journal Homepage : https://ijic.utm.my/

COMPARATIVE STUDY OF BIO-INSPIRED OPTIMIZATION ALGORITHMS FOR FEATURE SELECTION IN INTRUSION DETECTION SYSTEM

Abdul Azim Bin Anuar Veera & Assoc. Prof. Dr Anazida Binti Zainal
Faculty of Computing
Universiti Teknologi Malaysia
81310, Johor Bahru, Malaysia
Email: azim02@graduate.utm.my , anazida@utm.my

***Abstract*— The paper provides a comparative analysis of the two well-known bio-inspired algorithms (Bee Colony (BC) and Fish Swarm Algorithm (FSA) with the objective of feature selection in network-based Intrusion Detection Systems (IDS)) systems. IDS is meant to detect malicious behavior in the computer network using the traffic data that sometimes has high dimensionality. This dimension is reduced through feature selection where only highly relevant attributes are selected, this enhances quality detection and lower that cost of computation. UNSW-NB15 dataset was used to extract optimal feature subsets through BC and FSA algorithms and subsequently tested against a Random Forest classifier. Accuracy, precision, recall and F1-score and AUC-ROC were used to measure performance with ten test runs. Results indicate the stability and consistency offered by BC and the peak accuracy on FSA with higher variability. This identifies a very important trade-off between consistency and exploration of features during feature selection in IDS.** *(Abstract)*

Keywords — Bee Colony Algorithm, Fish Swarm Algorithm, feature selection, Intrusion Detection System, Random Forest Classifier.

I. Introduction

The role of Intrusion Detection Systems (IDS) acquires more importance as cybersecurity threats keep becoming more sophisticated and frequent. IDS serves as watchdogs in a network; it can detect unusual behaviors that may lead to an attack. Nonetheless, noise and high dimensionality of network traffic data may pose as a challenge to the effectiveness of IDS. All features of the dataset are not equally useful in the attack detection process since some of them can only bias the learning process.

Feature selection is an algorithm that enhances efficiency of the model by ensuring that dimensionality of data is reduced, increases precision and shortens time consumed in training the model. Although they can work effectively, traditional feature selection techniques are usually unsuccessful when coping with dynamic data, such as the data on IDS. In response, bio-inspired algorithms provide a solution in this regard since they have the prospect of adaptive intelligent searches. Two such algorithms are Bee Colony (BC) and Fish Swarm Algorithm (FSA) that reproduce the collective intelligence in nature to maximize the solution to a problem.

In this study, the comparison of BC and FSA is held in a head-to-head manner, analysis of these models is carried out on the UNSW-NB15 dataset, which is a modern benchmark dataset containing the multitude of modern types of cyber-attacks.

II. LITERATURE REVIEW

The efficacy of bio-inspired algorithms in the selection of features has been proven in many studies into the IDS. Kasongo and Sun (2020) compared various machine learning classifiers and feature selection approaches on the UNSW-NB15 dataset and showed the improvement in the accuracy by up to 92 %. To the best of our knowledge, 99.5 percent accuracy has been achieved in the proposed hybrid version of Whale Optimization Algorithm (WOA) with Genetic Algorithm (GA) to implement feature selection in a dataset of Bot-IoT (Wiley et al, 2024). Almomani (2021) used PSO, GWO, and MVO and compared them on the NSL-KDD dataset since the closest performance was identified to be about 98%.

These works testify to the usefulness of bio-inspired techniques, but none of them include an extensive comparative study of BC and FSA on UNSW-NB15.

The Bee Colony algorithm is based on the foraging behavior of the honeybees where employed bees exploit familiar food sources (i.e. feature subsets), onlooker bees probabilistically select between promising sources based on a fitness function and scout bees explore new areas. This makes it have a healthy mix of both exploration and exploitation across the world. The fish swarm algorithm, in its turn, simulates the movement of fish. Each fish also corresponds to a candidate feature subset and with local best solutions, direction of the swarm and crowding thresholds, it adjusts its position. BC is regarded as stable and strong, whereas FS as aggressive and adaptable explorer. The key reason behind the research is to know the strengths and weaknesses of both in the application of IDS.

Table II.1 Existing Work On Bio-Inspired Algorithm

Study	Algorithm Used	Dataset	Acc. (%)	Key Contribution
Almomani (2021)	PSO, GWO, MVO	NSL-KDD	97.8	Compared multiple metaheuristics for feature selection
Kasongo &Sun (2020)	Feature Selection + ML	UNSW-NB15	~88–92	Demonstrated feature selection's impact on IDS with UNSW-NB15
Ferreira & Antunes (2021)	AIS, GA	KDDCup 99	98.2	Showed benefit of behavior-based evolutionary systems
Wiley et al. (2024)	WOA + GA	BoT-IoT	99.5	Developed a hybrid model with superior accuracy
Kumar et al. (2023)	ACO, PSO	CICIDS2017	98.7	Applied hybrid ACO-PSO model for IoT anomaly detection

.

III. METHODOLOGY

This study, in terms of methodology, will take place in three consecutive steps, i.e. (1) Feature Selection based on Bio-Inspired Algorithms, (2) Classification Model Training and assessment, and (3) Comparative Analysis. All the phases are aimed at serving the objectives of the research to enhance the accuracy of IDS detections, minimize the computational overhead, and becoming knowledgeable about the behavioral advantages of the BCA as well as the FSA.

A. Phase 1 : Development of Bio-Inspired Algorithm

During this stage, two bio-inspired algorithms, namely, Artificial Bee Colony (ABC) and Fish Swarm Algorithm (FS) have been applied to decrease the dimensionality of the UNSW-NB15 data. This is to find the best and most pertinent subgroup of features that enhance a good classification and hence make a model efficient and effective.

Artificial Bee Colony Algorithm (BCA) is based on simulating foraging behavior of honeybees. The bees are segmented into three categories namely employed bees, onlooker bees and scout bees. A binary encoding is created based on each solution (i.e., a subset of features) in the sense that each bit of the binary representation relates to the inclusion of a specific feature (1) or its absence (0). At initialization a random population of feature subsets is created. In every iteration, employed bees search around their present solution. This is inclusive of local search coupled with diversity. The onlooker bees check the suitability of the solutions used by working bees and randomly choose one through a roulette wheel type of selection procedure. When after a particular number of cycles (limit parameter), a solution is not improved, it is discarded and in turn form around which bee, a scout bee is transformed that has the potential to come up with a new random solution.

Each bee is characterized by the accuracy of classification that it generates when we take up the features of choice and run through a Random Forest classifier. The next generation is preferred to have more accurate solutions. This is repeated and again until the highest number of iterations or convergence conditions are attained.

Fish Swarm Algorithm (FSA) is instead based on a model of social movements in fish, specifically on the swarming, following, and foraging behaviors of fish. Every fish contains a candidate solution (a feature subset). FSA also utilizes a binary factor to select the features, and here every fish adjusts its position in the feature space by looking around the environment in a fixed visual distance. The behaviors have the following meaning: (1) Prey behavior: a fish wanders in its neighborhood randomly. In case the new location decreases the fitness value (accuracy of classification), it transfers to that location. (2) Swarming Behavior: A fish tends to migrate towards the center of its local group when the average fitness of the the local group is better than that of the fish. (3) Following Behavior: A fish recognizes and follows a neighbor which performs better in sight at some distance.

Constraints which are taken into account by each behavior comprise crowd factor (to avoid overcrowding phenomenon) and maximum step size (to regulate intensity of movement). This directed random motion makes FSA able to search worldwide on the initial iterations and narrow down in the later iterations. Similarly to BCA, the quality of each fish is compared with Random Forest accuracy in the chosen sub-set of features.

B. Phase 2 : IDS Model Training

After obtaining the optimum sets of features using the two algorithms, the next thing is to train the Intrusion Detection System (IDS) classifier. Random Forest classifier is chosen at this step because it is very resistant to overfitting, it can take unbalanced data and displays high efficiency when working in high-dimensional conditions. Random Forest functions by building numerous decision trees in training and reporting the mode of it predictions. It does internal feature sampling (random subspaces), that supplements the external feature selection of the metaheuristic algorithms.

The training is performed using the Stratified K-Fold Cross-Validation method with k=3, ensuring that each fold maintains the same proportion of attack and normal traffic as the original dataset. This helps in obtaining reliable performance estimates and prevents data leakage. The model is evaluated on each fold using standard classification metrics such as accuracy, precision, recall, F1-score, and the Area Under the ROC Curve (AUC-ROC).

The classifier's hyperparameters are kept constant across both algorithms to ensure a fair comparison. Specifically, the number of estimators (trees) is set to 100, and the maximum depth is left to be determined by the algorithm during training. Other parameters like bootstrap sampling and Gini impurity are used as default in Scikit-learn.

C. Phase 3 : Comparative Analysis

The final phase of the methodology focuses on evaluating and comparing the outputs of the two algorithms using both quantitative metrics. The quantitative evaluation includes performance metrics such as: (1) Accuracy –the overall proportion of correct classifications. (2) Precision –the proportion of correctly classified positive (attack) instances. (3) Recall –the proportion of actual attack instances correctly identified. (4) F1-Score –the harmonic mean of precision and recall. (5) AUC-ROC – the probability that the classifier ranks a randomly chosen attack higher than a benign sample.

In addition to average metric values, the standard deviation across ten runs is calculated to assess the stability and robustness of each algorithm. A more consistent algorithm is likely more reliable in real-world IDS deployment.

Feature selection frequency, convergence graphs, and Venn diagrams showing feature overlaps. Heatmaps are used to depict how frequently each feature is selected across different runs, highlighting consistently important features. Finally, statistical tests (paired t-test) are applied to determine whether observed differences in performance metrics are statistically significant.

This phase also includes trade-off analysis highlighting the differences in convergence time, number of features selected, variance in performance, and interpretability.

IV. RESEARCH DESIGN AND IMPLEMENTATION

In this section, the steps of implementation, environment configuration, the procedure of experiments, and parameterization to assess the efficacy of the proposed feature selection framework are provided in detail. The methodology combines two bio-inspired algorithms, namely, Artificial Bee Colony (ABC) and Fish Swarm Algorithm (FSA) with a Random Forest classifier to constitute a wrapper-based optimization method of providing the most relevant features in the UNSW-NB15 intrusion detection dataset. The experiments were conducted in controlled conditions to provide a fair basis of comparisons and consistency of results.

.

A. Proposed Framework

The solution is three- phase architecture oriented. The first phase involves run of feature selection based on ABC and FSA individually. Every algorithm will need to find one set of features, out of the complete 49, presented on a binary representation of one variable per feature with a value of 1 denoting the inclusion of the feature. ABC is a simulation of bee foraging behaviour, using the iteratively searched solution space that refers to employed, onlooker and scouts bee phases. FSA imitates group behavior of fish in which each fish (solution) analyzes its location and moves based on the strategy of swarming, following, or foraging and adjusts according to the fitness value.

In phase two, Random Forest classifier is trained over the chosen feature combinations. The quality of all the subsets is measured in terms of five-fold stratified cross-validation in the classifier. This feedback loop gives out the fitness score which directs the search process of the bio-inspired algorithm. Random Forest is selected because of the capacity of dealing with a high-dimensionality model, its resistance to overfitting, and the simplicity of combining with a wrapper-based optimization approach.

The third step entails an analysis comparing the performance. Results are compared after succeeding in ten iterations of each algorithm on various measures of evaluation such as accuracy, F1-score, precision, recall and AUC-ROC. Further information is deducted on the frequency of features, convergence patterns as well as the statistical tests to determine the significance and replicability of the findings.

B. Experimental Setup

The experiments have been carried out in Python 3.9 in the Google Colab and Visual Studio Code frameworks. The important libraries were Scikit learn libraries of machine learning, Pandas and NumPy libraries that could be used to preprocess and manipulate the data and Matplotlib and Seaborn which were used to plot. A well-organized folder was used which hosted all algorithms and during the process, each dataset file was saved under an assigned data folder.

They used the UNSW-NB15 dataset, which is a modern intrusion detection benchmark that contains accurate attack and benign network traffic. UNSW_NB15_training-set.csv and UNSW_NB15_testing-set.csv were used as two CSV files. Preprocessing data were carried out to label encoding categorical features (proto, state, service), normalization of the continuous feature through MinMaxScaler, and combining train-test splits into a single dataset to perform the stratified evaluation. Types of attacks were classified in a two-fold label (normal or attack).

The methodology of experimentation with both algorithms has been the same having a fixed pipeline as follows: load data, apply feature selection algorithm (BCA or FSA), obtain feature subset, and train Random Forest classifier on the selected features, evaluate with 5-fold cross-validation, record performance metrics, and repeat the 10 times. To aggregate the statistical data and to examine it visually, the outputs in every run were recorded.

C. Parameter Configuration and Evaluation Criteria

In order to make them comparable, both ABC and FSA were set with same initial population sizes (20) and maximal iterations (50). Binary vectors of 49 dimensions were used to encode the absence or presence of features with each candidate solution. The criteria in which both algorithms stopped were either when the maximum number of iterations was reached or when no more improvement in fitness was achieved in 10 cycles. Fitness was the mean of the classification accuracies over Random Forest model during 5-fold cross-validation.

The parameters of Random Forest were changed for the experiments to be isolated to analyze feature selection. The test was done with its 100 trees (n_estimators=100), without a restriction in the depth of the trees and the default Gini impurity measure.

The next metrics were used to measure the performance: Accuracy (overall classification rate), Precision (the proportion of true positives to the sum of the true and false positives), Recall (the true positive rate), F1-score (the harmonic mean of precision and recall), and AUC-ROC (the discriminatory power along a continuum of thresholds). These measures give a multi-dimensional interpretation of how effective a classifier is especially when dealing with an uneven dataset such as UNSW-NB15.

Such visual assessment tools as plots of convergence showing iteration against fitness, feature selection heat maps illustrating the frequency of feature inclusion per run, also confusion matrices to decompose true/false positives and negatives, and Venn diagrams to determine the overlaps and differences between sets of features identified using ABC and FSA were also used.

V. RESULT AND DISCUSSION

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

A. Performance Metric Analysis

Table V.1 Average Performance Metrics

Algorithm	Acc.	Precision	Recall	F1 Score	AUC-ROC
Bee Colony	0.84	0.79	0.97	0.87	0.92
Fish Swarm	0.86	0.80	0.97	0.88	0.93

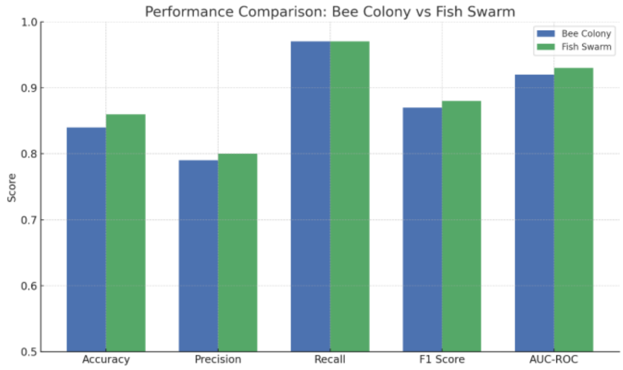

Figure V.1Performanca Metrics

Both algorithms were tested with common scores of classifications in ten runs of the experiment. In general, Fish Swarm Algorithm (FSA) surpassed Bee Colony Algorithm (BCA) in most of the measures. The accuracy of FSA was averagely 0.86 which was a bit higher than that of BCA, 0.84. It also performed with greater accuracy (0.80 as opposed to 0.79), F1-score (0.88 as opposed to 0.87) and AUC-ROC (0.93 as opposed to 0.92). Interestingly, the two algorithms had the same high levels of recall, which is 0.97, which means that they are highly capable of detection. The increased F1-score and AUC of FSA indicate that not only FSA was more precise, but also reasonably balanced in precision-recall-tradeoffs.

B. Confusion Matrix Analysis

Table V.2 TPR and FPR

Bee Colony			Fish Swarm		
Test	**TPR**	**FPR**	**Test**	**TPR**	**FPR**
1	0.9715	0.3623	1	0.9658	0.2555
2	0.9727	0.3135	2	0.9598	0.3038
3	0.9709	0.3010	3	0.9631	0.2357
4	0.9674	0.3062	4	0.9707	0.3164
5	0.9711	0.3086	5	0.9693	0.0546
6	0.9743	0.3327	6	0.9672	0.3112
7	0.9711	0.3064	7	0.9702	0.3022
8	0.9723	0.3133	8	0.9637	0.2472
9	0.9684	0.3172	9	0.9769	0.3315
10	0.9716	0.3152	10	0.9741	0.3229
Avg	**0.9711**	**0.3176**	**Avg**	**0.9681**	**0.2681**

Evaluation of the metrics appearing in the confusion matrix allows pinpointing the practical consequences of the behavior of each algorithm. BCA had a greater True Positive Rate (TPR) of 97.11 and was able to capture most of the attack traffic. This however was with the cost of large False Positive Rate (FPR) of 31.76%, which meant that it misreported close to a third of the benign traffic as malicious. Conversely FSA showed a more balanced distribution as it had a TPR of 96.81% and a much lower FPR of 26.81%. It means that, whereas FSA performs a little bit worse in terms of ability to detect all attacks, the number of classification errors committed within the overall evaluation is lower, which means that the former is preferable to be applied in the environment where false alarms are to be reduced at a higher priority.

.

C. Convergence Behaviour Analysis

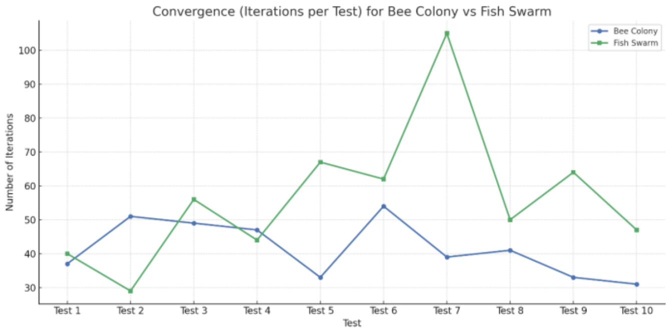

Figure V.2 Convergence of each test

There are additional differences dictated by convergence behavior of each algorithm. BCA has also converged quicker averaging 43 iterations/run and with a convergence being between 31 to 54 iterations. Consequently, FSA took longer to compute (the mean number of iterations completed and the maximum number of iterations before converging) in comparison to the other methods; the median value was 59-iterations, and sometimes it reached 105-iterations before converging. The convergence window of FSA was longer and provided it with the opportunity to search the feature space more deeply and find more optimal feature subsets resulting in preferable generalization. This, however, comes at a larger computational cost which may not be feasible in real time IDS models.

Table V.3 Convergence Analysis

Metric	Bee Colony	Fish Swarm
Fastest Convergence	31	29
Longest Convergence	54	105
Average Iterations	43	59

D. Stability and Robustness Analysis

Table V.4 Standar Deviation of algorithm

Metric	Bee Colony (Std Dev)	Fish Swarm (Std Dev)
Accuracy	0.01	0.015
F1 Score	0.009	0.011
AUC-ROC	0.01	0.024

The stability and robustness were measured by calculating the standard deviation of every measure by running each 10 times. Standard deviations in accuracy (±0.010), F1-score (±0.009), and AUC-ROC (±0.010) were always lower in BCA, which means that the accuracy of BCA was always reproducible across several initializations. On the other hand, FSA had a little bit more variability as shown by standard deviations of 0.015 and 0.024, corresponding to accuracy and AUC-ROC respectively. These findings indicate that BCA is more reliable and is preferable in a set of tasks that demand regular performance, whereas the exploratory characteristic of FSA can be helpful in unstable conditions but can result in rather unpredictable result.

E. Generalization and Overfitting Gap

Table V.5 Gap Analysis

Algorithm	Train Accuracy	Test Accuracy	Generalization Gap
Bee Colony	0.98–0.99	0.82–0.85	~0.13–0.15
Fish Swarm	0.98–1.00	0.84–0.87	~0.11–0.14

The concept of generalizing of IDS is important: the applicability of the model on unseen data. The two algorithms produced good training accuracy (0.98-0.99). Nevertheless, FSA had a little higher test accuracy (0.84-0.87 vs. 0.82-0.85 with BCA), and this translated into less generalization gap (0.11-0.14 with FSA compared to 0.13-0.15 with BCA). This implies that FSA models were better generalized across previously unseen samples of attack. BCA exhibited premature convergence symptoms in certain runs even with converging faster, thereby probably being overly fitted due to inadequate use of the feature space.

F. Feature Selection Analysis

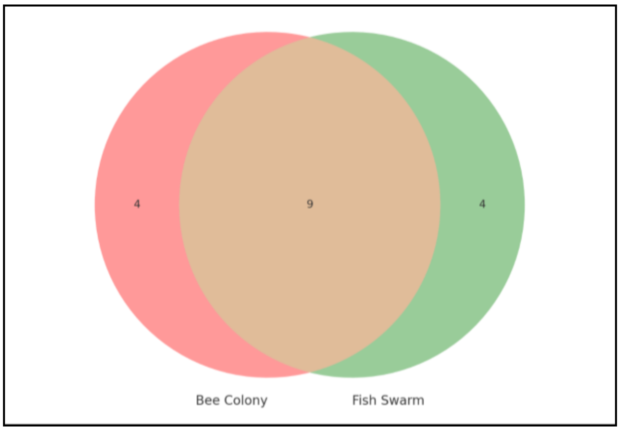

Figure V.3 Feature Overlap

Interesting patterns appeared in the selection of features. FSA picked an average of 12 features compared to BCA(13) but suppressed higher or identical classification performance, which has shown better efficiency in selecting features. The most relevant shared characteristics e.g., dbytes, synack, response_body_len, and is_sm_ips_ports appeared in both the algorithms output results showing its high significance in differentiating the traffic as attack traffic and normal traffic. The unique features chosen by BCA were tcprtt, ackdat, and ct_src_ltm and the features chosen by FSA were dur, sbytes, and dpkts hence FSA looked more into time and packet level characteristics.

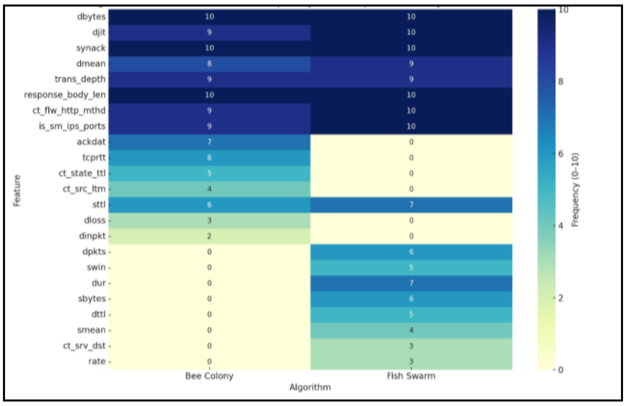

Figure V.4 Heatmap of Feature across 10 tests

According to a heatmap, there are features that both algorithms picked almost in every run and others that have occurred rarely. This implies an existence of a set of always high-impact features and subset of conditional features that enhance performance only when used in certain combinations.

G. Statistical Significance Test

Table V.6 T-Test Results

Metric	**t-statistic**	**p-value**	**Interpretation**
Accuracy	1.6732	0.1286	Not statistically significant
F1-Score	1.6164	0.1405	Not statistically significant
AUC-ROC	0.0000	1.0000	No difference

Through a paired t-test, it was intended to demonstrate whether the variations in performance between BCA and FSA were statistically significant. The p-values obtained are 0.1286 and 0.1405 of accuracy and F1-score, respectively, which is larger than the standard level of significance of 0.05. Hence, although FSA had higher results, on average, but the improvement was not statistically significant. The AUC-ROC was completely the same (p = 1.000). It supports once again the understanding that both algorithms can be potentially utilized as the option in the feature selection, and the area of their strengths depends on the utilized working conditions.

H. Dynamic Behaviour Analysis

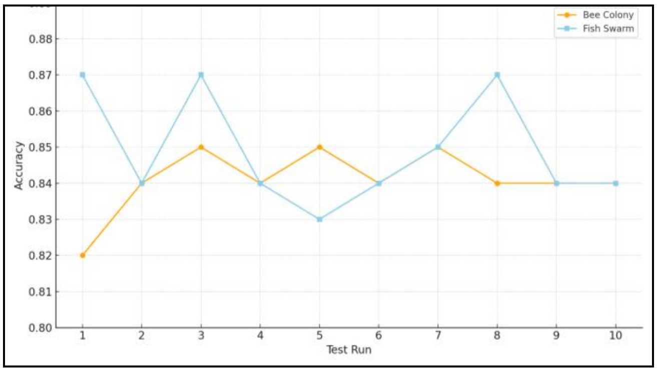

Figure V.5Accuracy across 10 test runs

The further investigation of the dynamic behaviour indicated that despite the BCA consistency, at some points, it grappled with early convergence, i.e., in Run 1, which was ceased at 37 iterations and resulted into sub-optimal accuracy (0.82). This was in cases whereby the selected set of features contained a few noisy or less effective features such as sttl. FSA on the other hand was not so narrow in its search and at times fell into local optimum but recovers in further iterations. This is also the capability of further improving non-optimal solutions that can result in its superior peaks at times (a 0.87 accuracy in Run 8).

I. Trade-Off Analysis

The compromise is clear between the performance and stability. BCA is fast and for this reason, highly repeatable and thus perfect when it comes to application in a situation whereby consistency is of the essence either through a static application or real-time application. Even though computationally more intensive, FSA offers enhanced exploration ability and it is set practically well in situations when an evolving network or complex one is in the picture and adaptability is given priority. Therefore, BCA would suit the resource-constrained setting e.g. embedded IoT and FSA would suit the adaptive IDS in cloud or enterprise environment.

J. Discussion

The results of the conducted experiment based on the comparative studies of the Bee Colony Algorithm (BCA) and the Fish Swarm Algorithm (FSA) provide valuable findings on the practical advantages and trade-offs of the two bio-inspired optimization approaches to the problem of features selection of an intrusion detection systems (IDS). Both algorithms had been effective in improving the performance of the classifiers when compared to when all features were used in the UNSW-NB15 dataset but they showed some behavioral peculiarities that make each more appropriate in implementation depending on the conditions of deployment.

On the performance level, FSA had a slightly higher classification metrics, a higher average accuracy, precision, and

F1-score. In addition to ensuring that FSA was better able to separate benign and malicious traffic under differing thresholds as indicated by the AUC-ROC, this was also marginally higher than that of BCA. These findings implicate that FSA swarming and exploratory search activated its ability in finding superior combinations of features that translated to greater peak model performance. Moreover, FSA had a lower average number of features chosen by it compared to BCA but with no decrease in classification quality and even an increase sometimes, thus showing that FSA is effective in simplifying the complexity of the models without loss of predictive capability.

This better performance was however at the expense of increased variability. It was observed that FSA had a wider range of collections among various runs, as indicated through increased standard deviation accuracy and AUC-ROC. This would be because it has a more aggressive and stochastic exploration which at times may converge to poor local minima based on the search path and initialization. BCA, on the other hand, exhibited a more solid performance profile without any decreases across any of the runs. It was also largely easier and quicker to converge, and therefore a safe subject with repeatability and predictability concerns.

These findings were also supported by the confusion matrices analysis. Although BCA was also able to attain a better level of true positive rate (TPR), this came with a higher false positive rate (FPR) which is likely to over-classify traffic labeled as benign as an attack. Such behavior may cause the real-world IDS systems to experience higher levels of alert fatigue. Conversely, FSA had more rewarding trade-off involving sensitivity and specificity; it had lower FPR thus better adapted to a setting in which the cost of false alarms is prohibitively high.

A vital point to note is the generalization ability of every algorithm. Despite the high training accuracy in both models, FSA had a narrower generalization gap between the training and test results, revealing that the model was less likely to suffer the problem of overfitting. This finding concurs with the more diversified feature exploration of FSA which should probably yield more substantial models that can better acclimatize to unseen data. In the meantime, BCA sometimes converged too early, particularly in the runs with small numbers of iterations, leading to the over-fitted models that performed poorly on the new data as compared to training data.

Another degree of insight was the feature selection behavior. Both algorithms commonly considered important features like dbytes, synack, and is_sm_ips_ports, so they are supposedly of the great importance in the process of intrusion detection. Nevertheless, FSA was susceptible to accessing and maintaining features concerning the time duration (dur) and statistics of packets (sbytes, dpkts), which indicate that FSA paid more heed to the volume- and time-based features. BCA on the other hand favored options like tcprtt and ackdat and these are more protocol and flow specific. Such difference indicates that the global search defined the features with wider diversity in FSA, whereas the exploitation behavior of BCA was more focused on optimization of a core group of features.

Last, the statistical test established that although average difference scores in performance metrics between the FSA and the other algorithm were in favor of FSA, this difference was not significant at 95 percent confidence interval. It means that, in spite of the fact that FSA shows the better overall result on an average, both algorithms are, in practice, relatively good, and the ultimate decision might be reduced to the use-case needs rather than comparison of their respective score.

To sum up, a trade off between stability and adaptability can be addressed in the discussion. BCA fits better in environments where reliability, interpretability and low cost of computing is preferred. Conversely FSA does better in adaptive/exploratory settings where peak performance and feature diversity are of greater importance, despite having perhaps somewhat increased computational requirements and output variance to that effect. These results confirm the notion that there is no such algorithm, which always works better than the other one, but instead the success of applying particular algorithm is determined by the fact that the behavior of the algorithm should be matched with the corresponding goals within a particular application.

VI. CONCLUSION

In the paper, a comparative study of the Bee Colony Algorithm (BCA) and the Fish Swarm Algorithm (FSA) when used in feature selection to improve the performance of the intrusion detection systems (IDS) has been presented. With UNSW-NB15 as an experimental benchmark, both of the algorithms could be compared based on several metrics, such as accuracy, precision, recall, F1-score, and AUC-ROC, convergence speed, feature selection performance, and resilience to the repeated runs.

The results show that both the algorithms work in reducing dimensionality as well as enhancing the classification performance in comparison to the performance of using all features. FSA outperformed in peak performances, exhibited better generalization and accuracy capability and also had less subset of selected features. It was however more variable between the runs of the test. BCA on the other hand provided more reliable and consistent results, converged quicker and had less standard deviation when using repeated tests. These results serve to support the determination of algorithm is influenced by both the average performance but also by the required operational environment in preference of BCA in real-time or embedded applications, and FSA in dynamic or explorative detection systems.

Finally, this work validates the usefulness of bio-inspired algorithms in intelligent feature selection in IDS as well as the fact that it is important to ensure that the behavior of the algorithm used is in tune with the performance, consistency, and computational needs of the target system..

VII. FEATURE WORK

In order to continue developing the existing study and further increase the efficiency of bio-inspired feature selection in systems of intrusion detection (IDS), a number of future directions could be suggested:

1) Hybrid Frameworks of Feature Selection

One area that can be conducted by future research is the designing of hybrid models that would be a juxtaposition of both BCAs and FSAs. An example of this is that I can allow BCA to converge to a final solution within a reasonable amount of time then follow this up with FSA to refine the feature set or make it more diverse. With the help of such a hybrid, the merits of both algorithms could be combined and more stable, optimized results obtained.

2) Parallel and Accelerated Computation

These algorithms are very parallelizable, as they are population based and each is an iterative algorithm. Execution time of BCA and FSA may be reduced significantly by implementing BCA and FSA with GPU acceleration or distributed computing frameworks and thus it may be capable of real time IDS applications.

3) Real-Time And Streaming Ids Testing

In future, it can be suggested that researchers need to concentrate on combining suggested feature selection structure with real-time or streaming intrusion detection models. An assessment of these algorithms on dynamic domains, where the behavior of networks changes with time would challenge them on their flexibility and realistic applicability in the deployment of real networks.

4) Multi-Objective Optimization

At the moment the algorithms use only one objective to optimize which is the accuracy of classification. Multi-objective optimization can be applied in the future work by applying methods such as NSGA-II or MOEA/D and this approach can allow optimizing more than one objective such as the maximization of accuracy and minimization of selected features or false positives and maximising recall.

5) Explainable Feature Selection

Since there is an increased interest in making machine learning systems more transparent, in the future, the explainability of machine learning should be introduced even at the feature selection stage. The IDS can be simplified and made more interpretable by the introduction of such tools like SHAP and LIME that will allow the cybersecurity experts to understand the importance of the selected features to classifications and use these tools to better work with the identification model.

6) Lastly, verification of the suggested algorithms on various datasets like NSL-KDD, CICIDS2017, and BoT-IoT should be conducted in future. This would portray the extensiveness of the algorithms and its resistibility in a variety of attack situations and network traffic characteristics.

References

[1] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 231–246, Dec. 2018. https://doi.org/10.1016/j.fcij.2018.06.001

[2] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computers & Security*, vol. 92, p. 101752, Mar. 2020. https://doi.org/10.1016/j.cose.2020.101752

[3] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia, Nov. 2015, pp. 1–6. https://doi.org/10.1109/MilCIS.2015.7348942

[4] F. Pourpanah, M. Mohammadi, K. Wang, M. F. Zolkipli, and A. M. Shabut, "Swarm intelligence algorithms for feature selection in intrusion detection systems: A comprehensive review," *Applied Soft Computing*, vol. 137, p. 110180, Mar. 2023. https://doi.org/10.1016/j.asoc.2022.110180

[5] S. Wiley, D. Sanchez, and R. Kumar, "A hybrid whale optimization and genetic algorithm for anomaly detection in IoT-based intrusion detection systems," *IEEE Access*, vol. 12, pp. 19271–19282, Feb. 2024.

[6] A. Almomani, "Evaluation of metaheuristic algorithms for feature selection in network intrusion detection," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 95–101, 2021. [Online]. Available: https://doi.org/10.14569/IJACSA.2021.0120612

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018