IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 4, August, 2024, pp. 1629-1638
Journal homepage: http://iieta.org/journals/isi

Enhanced Artificial Bee Colony Algorithm with Pretrained Model Functional Weight and Modified Selection Strategy for Text Classification

Oluwaseun Peter Ige[1,2], Keng Hoon Gan[1*]

[1] School of Computer Sciences, Universiti Sains Malaysia, Gelugor 11800, Malaysia
[2] Department of Planning, Research and Statistics, Universal Basic Education Commission, Abuja 900284, Nigeria

Corresponding Author Email: khgan@usm.my

https://doi.org/10.18280/isi.290436

Received: 30 December 2023
Revised: 16 July 2024
Accepted: 29 July 2024
Available online: 21 August 2024

Keywords:
metaheuristic algorithm, functional weight, selection strategy, pretrained model, text classification

ABSTRACT

Previous works have proposed various techniques to address the premature convergence problem, where candidate solutions get trapped in local optima instead of reaching the global optimum. This has been tackled using different selection methods in metaheuristic search algorithms. However, while much of the literature focuses on either the search operators or the creation of algorithm variants, research indicates that the effectiveness of the search procedure depends on both the search operators and the selection methods. Incorporating problem-specific functional weights enhances dynamic adaptation to data patterns, reflects data relevance, and improves generalization. This paper offers an enhanced Artificial Bee Colony algorithm including functional weights and a modified selection strategy (ABC-FWMSS) to prioritize features, aiming to achieve an optimal solution and a dynamic balance between exploration and exploitation. The exploration ability of the Artificial Bee Colony is enhanced using pretrained model functional weights during the employed bee phase, while its exploitative capabilities are boosted using tournament selection and employed bee index during the onlooker bee phase. This approach dynamically balances exploration and exploitation. The proposed method achieved 96% precision on the 20 Newsgroups dataset, with the highest fitness score and a 48.8% drop in the number of selected features.

1. INTRODUCTION

Many researchers have become interested in feature selection as it minimizes complexity and computational resources using different techniques ranging from filter [1], wrapper [2], embedded [3], hybrid [4], ensemble [5]. The use of metaheuristic algorithm for feature selection has been on the fore to reduce irrelevancy and improve classification performance [6]. There are two primary phases in the population-based Meta-Heuristic Algorithm Search (MHAS). The first phase is to select the solution candidates according to the selection method used by the population, for reference positions on a search space. The second phase selects the search procedure for the chosen reference positions. Search operators in the MHAS are responsible for balancing exploitation and exploration. Exploitation, or intensification, involves performing a neighbourhood search near the reference position to find the best solution close to the chosen candidate. Exploration, or diversification, protects the population from getting stuck in local optima. Mutation is used to successfully alter the reference positions when the search process is unable to be improved upon or a successful solution is not found, allowing the algorithm to search new regions of the search space [7]. The goal of diversifying is to open the search field to more fruitful exploration sites than reference places. It is required that the selection method picks the suitable individual for updating the next generation of the population with the required information. Three suitable types of selection procedures can be distinguished: non-deterministic, deterministic, and probabilistic. In non-deterministic methods, individuals are chosen at random from the population [8]. The deterministic approach uses elitist or greedy method to select the best individual [9]. The probability method uses roulette or tournament search to combine the features of both non-deterministic and deterministic approaches to calculate the fitness function and greedily select the best individual [10]. Significant advancements have been made in the recent few decades in the development of MHAS [11, 12] and real-life applications of Meta-heuristic Algorithms (MHAs) in solving optimization problems [13]. A recent development is the use of Genetic Algorithm (GA) [14], which is an Evolutionary Algorithm (EA), and Swarm Intelligence (SI) based algorithms. A broad range of SI-based algorithms have become known, these include Ant Colony Optimization (ACO) [15], Particle Swarm Optimization (PSO) [16], Bat Algorithm (BA) [17], Firefly Algorithm (FA) [18]. Among the SI algorithms, Artificial Bee Colony (ABC) was presented in 2005. It simulates how bee colonies go about finding food [6]. Its capacity to resolve practical optimisation issues, like quadratic assignment [19], sentiment classification [20], and automatic programming [21], has garnered significant interest since its discovery.

However, ABC still has certain challenges when it comes to tackling complex problems because of its solution search equation. Several studies have demonstrated that the exploration phase of the search equation for a solution is prioritised over the exploitation phase, leading to lower convergence accuracy and slower convergence rates [22, 23]. To overcome these problems, various improvements in the ABC algorithm have been suggested, focusing on using the best individuals to enhance exploitation. These ABC variants can potentially outperform the conventional ABC but face the challenge of utilizing competent individuals without sacrificing population diversity. Since exploration and exploitation are inherently incompatible, achieving a balance between them is crucial for optimal performance. While hundreds of new MHAS have been developed to improve the efficiency of exploitation and exploration tasks for better search performance [24], most studies focus on designing search operators or developing algorithm variants [25]. However, research indicates that the effectiveness of the search procedure depends on both the search operators and the selection methods [26, 27]. Selection methods and search operators must work together [28]. The synchronization between search operators and the selection method's placement of solution candidates in the search space determines the success of the search process [29]. This paper aims to propose an algorithm that dynamically balances exploration and exploitation by modifying selection methods and perturbing search operators.

This study offers an enhanced Artificial Bee Colony (ABC) algorithm with Functional Weight and Modified Search Strategy (ABC-FWMSS) to improve the selection process for solution candidates and dynamically balance exploration and exploitation. This improvement adds pretrained model features and weights to the ABC model, which are obtained by hybridising univariate filter feature selection techniques (Chi2, InfoGain, and ANOVA). The pretrained model features make use of the embedded knowledge, and the weights act as coefficients to emphasise or de-emphasise certain aspects according to their relative importance during the evaluation of the objective function. The results reveal that the suggested method performs better than other optimisation algorithms in terms of choosing the optimal feature subset and increasing precision. The following are the study's primary contributions:

- An algorithm where the goal is to minimise the number of features selected and maximise accuracy.
- A modified search operation with pretrained functional weight to perturb the mutation process at the employed bee phase.
- A modified search strategy for onlooker bees that uses both tournament selection and employed bee index as determinants for crossover. The use of tournament helps to maintain diversity in the population while the employed bee index promotes exploitation focusing on the solutions that are performing well in the employed bee phase. Hence balancing exploration and exploitation.
- Experimental comparisons with different optimization techniques show that the suggested ABC-FWMSS justifies the objective function through the highest fitness scores with a drop in the number of features having high precision.

The remaining part of this study is organized into the following sections: The conventional ABC and related works are summarized in section 2. The proposed method is presented in section 3. Section 4 describes the experimental settings consisting of dataset, pretrained model features and weights, parameter tuning, performance analysis, and statistical significance. Finally, section 5 presents the conclusion.

2. CONVENTIONAL ABC AND RELATED WORKS

2.1 The conventional ABC

Using the simulation of an intelligent foraging behaviour for honeybee swarms that belong to a branch of EA, ABC is a population-based optimization technology. ABC identifies a food source position as a possible solution to an optimisation problem, with the nectar size of each point indicating both the suitability and quality of the relevant selection. The population includes sources of food, which are created by three different bee types: scout, employed, and onlooker. The employed bees make up the initial half of the population. They are responsible for conducting a random search in the adjacent area that corresponds to their parent food source and communicating with other bees to provide information. Onlooker bees, make up the second half of the colony, tasked with finding greater food source placements around the good solutions. They are selected based on the value of information provided by employed bees. The final group of bees are scout bees. An employed bee will leave a food source position if it is not enriched by other bees after a certain number of visits around it. This employed bee will turn into a scout bee, roving the entire search space at random to find new locations for food sources. Within the ABC paradigm, an "individual" or viable solution to the problem at hand is referred to as a "food source." The potential solution has a high fitness value if the food source has an abundance of nectar. In the typical ABC, each of the four phases is recursively completed until the termination condition is satisfied. The following are these phases.

Initialization phase: ABC uses an initial population to start its search operation. Thus, a starting population is produced. Assume that there are NFS food sources in the population, each of which is produced using Eq. (1):

$$x_{ij} = x_j^{high} + rand(0,1) \times \left(x_j^{high} - x_j^{low}\right) \tag{1}$$

where, i=1, 2, ..., NFS, j=1, 2, ..., D. NFS stands for the number of employed bees or onlooker bees; D is the search space's dimensionality; x_j^{high} and x_j^{low} connote the jth dimension's upper and lower bounds, respectively. Additionally, Eq. (2) shows the fitness value for each unique food source:

$$fitness\ \left(x_i\right) = \begin{cases} \dfrac{1}{1+f\left(x_i\right)}, & if\ \left(f\left(x_i\right) \geq 0\right) \\ 1+\left|f\left(x_i\right)\right|, & otherwise \end{cases} \tag{2}$$

where, fitness(x_i) connotes the fitness value of the ith food source position x_i and $f(x_i)$ stands for the objective function value of the food source position x_i for the optimization problem.

Employed bee phase: In this phase, employed bees are responsible for discovering new food sources within the broad search space as depicted by Eq. (3):

$$v_{i,j} = x_{i,j} + \phi_{i,j} \times \left(x_{i,j} - x_{r,j}\right) \tag{3}$$

where, $V_i=(v_{i,1}, v_{i,2}, \dots, v_{i,D})$ is the new food source that corresponds to the previous food source X_i, and X_r is a randomly chosen food source from the population that differs from Xi. In the interval [−1, 1], $\phi_{i,j}$ is a uniformly distributed random number, and j is chosen at random from {1, 2,..., D}. When v_i outperforms its parent x_i in terms of fitness, x_i is replaced with v_i, and the counter that counts the number of successive failed adjustments to x_i's food source position is reset to 0. If not, the counter is increased by one and xi is kept to enter the following generation.

Onlooker bee phase: The onlooker bee phase carefully searches for food sources. This search behaviour is exploring the neighbour of solution candidates (local search). Each food supply has a selection probability, as demonstrated by Eq. (4):

$$p(x_i) = \frac{fitness(x_i)}{\sum_{j=1}^{NFS} fitness(x_i)} \tag{4}$$

where the selection probability is represented by $p(x_i)$. A higher fitness value increases the selection probability with the chance that a potential food supply would be chosen more than once. The same solution search in Eq. (3) is utilised to create new food sources, such as the employed bee phase, after deciding which food sources should be chosen.

Scout bee phase: The goal of this phase is to provide new food sources while preventing population stagnation. For every food source, a counter is created to track the number of times it has failed to update. The employed bee will abandon the food source position with the greatest counter value if its counter value exceeds the limit value, and it will then follow the path of Eq. (1) to find a new food source position by using a scout bee. The scout bee thus becomes an employed bee again after the new food source has been updated and its counter value has been reset to zero.

2.2 Related works

Conventional ABC randomly selected source of food for the solution search equation is used to produce offspring that leads to high exploration but insufficient exploitation. Over the past few years, numerous better ABC modifications have been formed to improve exploitation to meet this difficulty. Aside, due to the technological development, optimization problems are becoming complex, necessitating improvements to the EA's performance. A brief review of this development is highlighted below.

(i) Development of new solution search-equations: Gao and Liu [30] offered an enhanced search equation ABC/best/1, which is predicated on the idea that to maximise exploitation, the bee only looks around the best solution from the earlier iteration. Additionally, probability governs how frequently the ABC and ABC/best/1 solution search equations are used. Zhou et al. [31] built a Gaussian bare-bones ABC variant (GBABC). This method substituted the Gaussian bare-bones search equation for the original solution search equation, which was based on the global best individual. Unlike GABC, GBABC generates their offspring by the current and global best individuals. Aslan et al. [32] designed an enriched quick ABC variant (iqABC) while using the global best individual. Unlike GABC and GBABC, iqABC starts the search process straight with the global best individual. This approach enhances exploitation, while its counterbalancing mechanisms ensure a balance between diversification and intensification.

Zhou et al. [22] created a multi-elite guidance-based ABC variant (MGABC). This makes use of multiple elite individuals against a global best individual, to prevent the algorithm from becoming too greedy. Very recently, Zhou and Zhao [33] developed an adaptive ABC variant (AABC-DQN) in which the neighbourhood search operators are reasonably selected using a deep Q-learning network to increase the search range. This was applied to solve milk-run vehicle scheduling problems based on the supply hub.

(ii) Neighborhood-based ABC modifications: It is seen from the new solution search equations that global best or elite individuals could be overutilized which may initiate the premature convergence problem. As a way out, more flexible neighbourhood-based ABC variations have been developed to use neighbourhood topology to distribute important knowledge from superior individuals. Wang et al. [34] proposed a neighbourhood-based ABC (NSABC) with a ring neighbourhood structure. The neighbourhood with the best person selected to start the search in the solution search equation is the neighbourhood represented by the idea of k (neighbourhood radius). The NSABC assigns a fixed value to K. Xiao et al. [35] introduced an improved edition of the neighborhood-based ABC variant (ABCNG), featuring a dynamically adjustable neighborhood radius based on the quality of the offspring, unlike NSABC. Zhou et al. [36] proposed an ABC algorithm based on adaptive neighborhood topology (ABC-ANT). Instead of relying on a single type of neighborhood topology, ABC-ANT uses the fitness distance correlation method to find features after which it creatively chooses the best neighbourhood topology.

(iii) Strategy-based Ensemble ABC modifications: New solutions are suitable for different problems with different search capabilities, combining various solution search equations to build an ensemble strategy-based variant will harness the strength of individual solutions. Xiao et al. [37] produced an improved ABC variant (ABC-ESDL) based on elite strategy and dimension learning. Dimension learning leverages the differences between two random dimensions to produce a significant jump, but the elite technique chooses superior solutions to speed up the search. Xue et al. [38] developed a self-adaptive ABC (SABC-GB) that utilizes the historical performance of each adaptive strategy to generate offspring. Chen et al. [39] offered a self-adaptive differential variant of ABC (sdABC) that achieves accumulated improvements in the objective function through the use of three differential search strategies.

(iv) Combination with other algorithms (Hybrid modifications): To enhance one another, this variation joins ABC with other search algorithms. Chen et al. [40] hybridized ABC with Particle Swarm Optimization (PSO) to form ABC-PSO. PSO has poor exploration capabilities and easily falls into local optima, while ABC has low exploitation capacities, leading to slow convergence. To address these issues, the ABC was first improved by combining greedy selection and crossover, and a sine-cosine method was used to help PSO escape local optima. This resulted in a new hybrid algorithm based on the improved ABC and PSO. Ustun et al. [41] hybridized ABC and Differential Evolution (DE) algorithms to create mABC, aiming to enhance precision. While ABC excels in exploration, DE is effective in exploitation. To improve the exploitation ability of ABC, the onlooker bee operator was replaced with the mutation and crossover phases of DE, increasing accuracy and speeding up convergence.

While Chen et al. [40] combined two swarm intelligence algorithms (ABC and PSO) and Ustun et al. [41] combined one evolutionary algorithm (GA) with a swarm intelligence algorithm (ABC), our approach hybridizes an evolutionary algorithm (GA) and a swarm intelligence algorithm (ABC) with a strategy-based ensemble of Chi2, ANOVA, and InfoGain for improved feature selection.

3. THE PROPOSED METHOD

The filter feature selection method helps to reduce dimensionality, and overfitting, enhance computational efficiency, improve solution quality and increase model interpretability, it is therefore a critical preprocessing step to increase the efficiency and usefulness of metaheuristic algorithms. The proposed model will be split into two parts: ensemble filter feature selection and enhanced ABC algorithm as depicted in Figure 1.

3.1 Ensemble filter feature selection method

The main motivation of this ensemble, in the first part, is the combination of computational power from univariate filters with enhanced voting mechanisms, to aggregate the output of multiple approaches, improving discriminative filtering performance for better classification performance. In addition, this will make features relevant for label classes and thus reduce redundancies [42].

The following is a basic explanation of the filter univariate algorithms:

Chi-square (Chi2)

This method tests independence to determine whether a feature is associated with its target variable by evaluating its relationship. The higher the value, the more its relevance to the target class.

$$x^2=\sum\frac{(O_i-E_i)^2}{E_i} \tag{5}$$

where, x^2=chi-square, O_i=observed value (actual value), E_i=expected value

ANOVA

It calculates the ratio of the variance between groups to the variance within groups. It estimates the significance of feature disparities for different classes. Features with higher scores are considered more relevant.

$$ANOVA(Z,y)=\frac{\left(\frac{SSB}{K}\right)}{\left(\frac{SSE}{(n-k-1)}\right)} \tag{6}$$

where, Z=feature matrix, y=target vector; SSB=sum of squares between groups (variance between groups' means); SSE=sum of squares within groups (variance within groups' means); k=groups number, n= samples total.

Infogain

It measures the target variable's uncertainty after observing a feature. It measures the information a feature adds to the target variable, with higher information gain indicating that the feature is more effective at distinguishing between classes.

$$\begin{aligned}&Infogain(Z,y)\\&=H(y)-H(y\,|\,Z)\end{aligned} \tag{7}$$

where, Z=feature matrix, y=target vector, H(y)=the target variable entropy; H(y | Z)=target variable conditional entropy given the feature.

The choice of Chi2, ANOVA, and InfoGain filter feature selection methods is based on their different metrics. Combining these methods is more effective than using just one, as shown in text classification applications. This approach offers a unique perspective on feature importance, as demonstrated in our previous work [43] when compared with other state-of-the-art feature selection method [44].

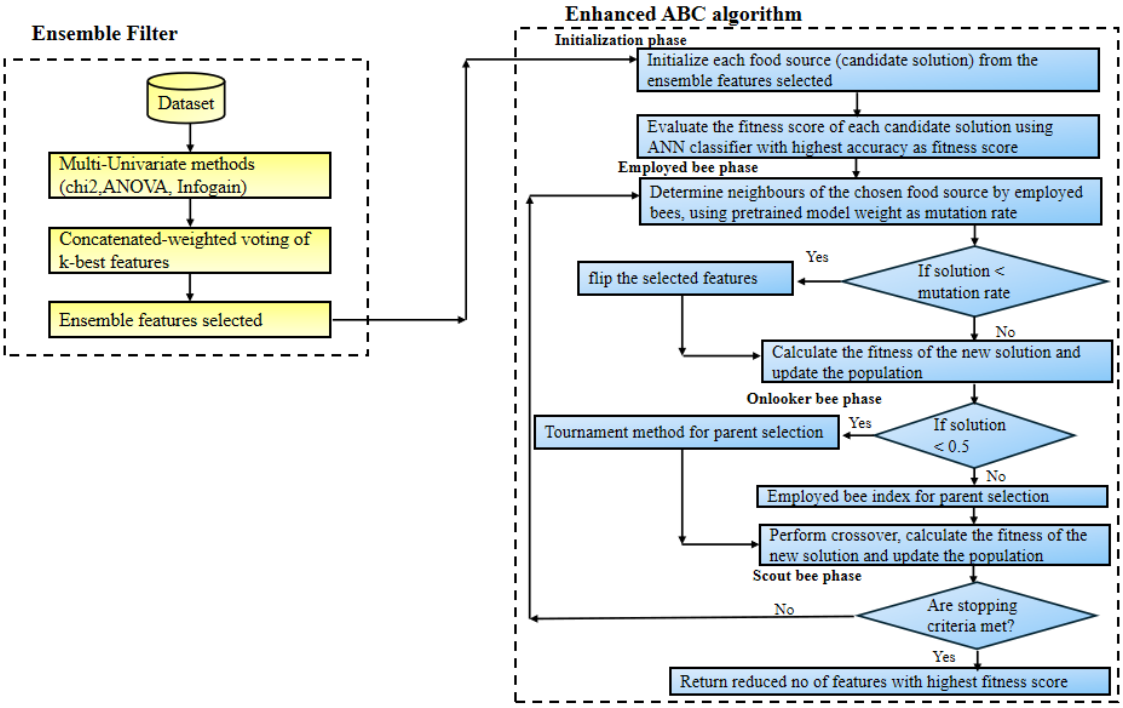
Ensemble Filter
Dataset
Multi-Univariate methods (chi2,ANOVA, Infogain)
Concatenated-weighted voting of k-best features
Ensemble features selected
Enhanced ABC algorithm
Initialization phase
Initialize each food source (candidate solution) from the ensemble features selected
Evaluate the fitness score of each candidate solution using ANN classifier with highest accuracy as fitness score
Employed bee phase
Determine neighbours of the chosen food source by employed bees, using pretrained model weight as mutation rate
Yes
flip the selected features
If solution < mutation rate
No
Calculate the fitness of the new solution and update the population
Onlooker bee phase
Tournament method for parent selection
Yes
If solution < 0.5
No
Employed bee index for parent selection
Perform crossover, calculate the fitness of the new solution and update the population
Scout bee phase
No
Are stopping criteria met?
Yes
Return reduced no of features with highest fitness score

Figure 1. The Proposed model - (ABC-FWMSS)

The top-ranked relevant features were selected in the three feature selection methods from 100 to 1000 iterations in the interval of 100 to represent concatenated array of features where unique features from the concatenation were selected. The selected features were classified with ANN classifier which becomes the input to the enhanced ABC algorithm part.

3.2 Enhanced ABC algorithm

Here, the selected features with the weights from the ANN Classifier becomes the input to this algorithm from each of the iterations from k-select of the best features ranging from 100 to 1000 at the interval of 100.

The general highlights are as follows:

Feature Subset Generation: The feature subset is represented by a binary vector with each element indicating the presence (1) or absence (0) of a feature.

Problem Formulation: The problem is to find the optimal subset of features that maximizes a predefined fitness function. The fitness function evaluates the performance of the selected features using a pre-trained model.

Evaluation Criteria: The evaluation criteria are based on the fitness of the feature subset. Fitness represents how well the selected features contribute to the overall performance of the model.

Fitness Criteria: The fitness is calculated based on the accuracy of the model using the selected features. Higher accuracy values indicate better fitness. The research is adopting a maximization objective function of maximizing accuracy [45, 46]. We are looking for the candidate solution that will yield the highest fitness score while selecting a smaller number of features. While maximizing accuracy, the objective is to attain the best possible performance in terms of correctly classifying instances.

The mathematical expression for the fitness function is represented as follows:

$$fitness = accuracy \tag{8}$$

where, *accuracy* is the accuracy of the model using the selected feature subset.

Objective Function: The objective function is denoted as f(x), where x represents a candidate solution.

Fitness Score (Fit): The fitness score, often denoted as Fit(x), is a scalar value that quantifies how well the solution x performs according to the objective function.

Optimization Goal: Since the problem is a maximization problem (Maximize classification accuracy), the fitness score is:

$$Fit(x) = f(x) \tag{9}$$

$$Fitness\ Function: F(selected\_feature\_mask) \tag{10}$$

$$F(selected\_feature\_mask) = accuracy \tag{11}$$

$$accuracy = \frac{NCCI}{TNI} \tag{12}$$

where, NCCI=Number of correctly classified instances; TNI=Total number of instances; selected_feature_mask is a binary vector indicating which features are selected (1) and which are not (0).

Initialization phase

The population is initialized with random binary vectors representing different feature subsets. The feature subset is represented by a binary vector where each element indicates the presence (1) or absence (0) of a feature. The initialization process in the enhanced ABC algorithm and the conventional ABC expressed in Equation (1) are similar in the sense that they both involve the random creation of solutions within the search space, but they are expressed in different forms. In the conventional ABC, each decision variable x_j in the solution x_i is initialized randomly within its specified range $[x_j^{low}, x_j^{high}]$, while the initialization of enhanced ABC involves creating a binary vector where each element in the vector corresponds to a feature which is randomly set to 0 or 1 indicating whether is selected or not. The conventional ABC initializes continuous decision variables, while enhanced ABC sets binary vectors representing feature selection. Also, the former deals with continuous optimization problems where solutions are points in a continuous space while the latter allocates discrete search space where each feature is either selected or not.

Employee Bee Phase

In this phase, the mutation ability of Genetic algorithm is introduced to strengthen global exploration. We introduced the pretrained weights from the first layer of a neural network to perturb the mutation operation and serve as mutation rate.

Implications on Exploration and Exploitation

Exploration: If the candidate solution is unrelated to functional_weights, the similarity function (dot product) will return a low value which means a lower mutation rate. This inspires exploration by allowing for more random changes in the feature subset.

Exploitation: If candidate solution is related to functional_weights, the similarity indices will return a high value resulting in higher mutation rate. This guides the search towards local regions that align with the knowledge encoded in the pre-trained model, facilitating exploitation.

The dynamic adjustment of the mutation rate during the optimization process, because of the variation in the weights of the pretrained model features, can help achieve a balance between exploration and exploitation based on the algorithm's performance and progress.

Onlooker Bee Phase

The employed bee index is a mechanism for selecting parents directly from the pool of employed bees. This strategy favours exploitation by prioritizing solutions that have shown higher fitness in the current population while tournament selection is a mechanism for selecting individuals for reproduction based on a tournament style competition and the champion from the contestant individuals is chosen to proceed for crossover. This supports diversification. Using both tournament selection and employed bee index in the onlooker bee phase introduce diversification in the parent selection process, favouring both exploration and exploitation in the optimization algorithm.

Implications on Exploration and Exploitation

Exploration: The combination of tournament selection and employed bee index introduces diversity in the choice of parent solutions. Tournament selection tends to favour solutions with higher fitness, promoting exploitation, while the employed bee index introduces random exploration. The 50% probability of choosing between tournament selection and the employed bee index ensures a balance between exploitation and exploration.

Exploitation: The employed bee index implies a deterministic selection process where solutions are chosen based on their fitness. Determinism in parent selection aids in refining and reinforcing advantageous traits, contributing to the exploitation of the current knowledge about the optimal solution. Tournament selection is naturally biased toward exploitation, as it favors solutions with higher fitness. This aspect ensures that solutions with better performance have a higher chance of being selected as parents.

Combining the two approaches can be beneficial for avoiding premature convergence, and reducing sensitivity to local optima.

A novel Contiguous Convolutional Neural Network (CCNN) was proposed and optimized using Differential Evolution (DE), named the Evolutionary Contiguous Convolutional Neural Network (ECCNN). Additionally, the researchers introduced a swarm-based Deep Neural Network (DNN) that utilizes Particle Swarm Optimization (PSO), called Swarm DNN. The performance results showed a precision of 88.76% for ECCNN and 87.99% for Swarm DNN on the 20 Newsgroups dataset [47]. Our ABC-GA approach precision outperforms it on the same dataset.

3.3 Experimental criteria

After conducting several experiments, the highest fitness score was recorded for parameter values shown in Table 1. Determining the control or hyper parameters for the algorithm require domain knowledge and reference from the past literatures.

Table 1. Experiment parameter of ABC-FWMSS

Control Parameter	Length
Population size	no of selected features
Max_trial_limit	10
Food sources	50
No of iterations	100
Max_trials	20
Tournament size	5
Onlooker bee	50
Scout bee	50

From Table 1, the population size is the number of selected features from the ensemble feature selection for each iteration.

4. EXPERIMENTS AND PERFORMANCE EVALUATION

To evaluate ABC-FWMSS, 20-Newsgroup dataset was chosen. The threshold feature selection k ranges from 100 to 1000 at an interval of 100. The ABC-FWMSS and other methods were implemented in Python 3.11 using the latest version of numpy, pandas, and scikit-learn. Experiments were conducted on a system with an Intel® Core™ i7-10510U 2.30GHz and 16GB RAM, with a 1TB storage capacity. The experimental outcomes were compared with standard feature selection techniques and analyzed using 5-fold cross-validation to calculate the averages.

4.1 Dataset

The evaluation uses the 20-Newsgroup dataset, a well-known benchmark containing approximately 20,000 documents from 20 different newsgroups. After preprocessing (including stop-word removal, lowercasing, and stemming), the documents are represented using the TF-IDF model with a document frequency (DF) threshold greater than 3. The final selections vary in size, ranging from 100 to 1000 features, with increments of 100.

The dataset is split into training, validation, and test sets. The validation set is used for hyperparameter tuning. To reduce redundancy and improve interpretability, unique features were chosen, which also help to minimize overfitting and enhance the model's ability to generalize to new data.

4.2 Performance analysis of ABC-FWMSS

To justify the performance of our enhanced method over other metaheuristics algorithms, fitness score, number of selected features, class specific metrics of precision, recall, F1-score and statistical significance were used for comparison.

4.2.1 ABC-FWMSS on the fitness scores

The fitness score comparison is shown in Figure 2.

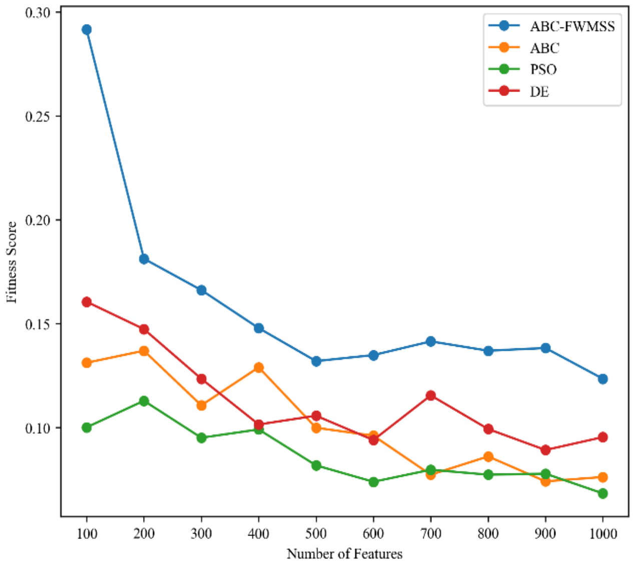

Figure 2. Fitness score comparison between ABC-FWMSS and other metaheuristic algorithms

It could be seen from Figure 2 that the fitness score of ABC-FWMSS is higher than other metaheuristic algorithms using the same ensemble features as input across iterations ranging from 100 to 1000. This validates the effectiveness of our enhancement using tournament and employed bee index at the onlooker bee phase of the model. The duo has enhanced both the exploration and exploitation of the algorithm and hence an improved fitness score across the iterations.

4.2.2 ABC-FWMSS on the number of selected features

To further justify the position of the ABC-FWMSS, the number of features selected by ABC-FWMSS was compared with the ensemble features as shown in Figure 3.

Figure 3 showed across iterations (100 to 1000) that there is 48.8% (see Table 2) reduction on average in the number of selected features by enhanced ABC when compared with the ensemble features. This has strengthened efficiency of the proposed method (ABC-FWMSS) by the reduction in model complexity. This will in turn improve the model's performance.

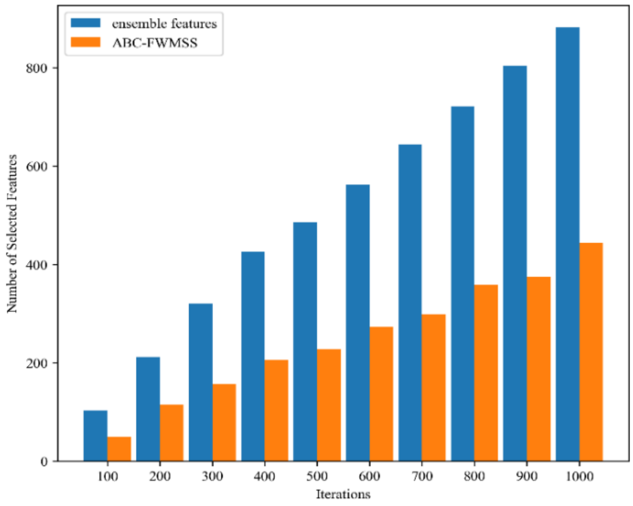

Figure 3. Number of selected features comparison between enhanced ABC and ensemble filter

4.2.3 ABC-FWMSS with other existing methods

To justify the performance of ABC-FWMSS on existing methods, we compared the proposed method across classes with established classes using precision, recall and F-measure metrics on SVM classifier. This comparison was carried out with 3 Algorithms (IG, Chi-square and ABCFS) adopted from the work of Balakumar & Mohan where the authors proposed ABCFS as a feature selection technique to classify 20newsgroup documents based on their content.

Table 2. Percentage reduction in the number of selected features (100-1000)

Methods	**100**	**200**	**300**	**400**	**500**
Ensemble features	103	212	321	426	485
ABC-FWMSS	49	115	157	206	228
% of Reduction	**47.6**	**54.2**	**48.9**	**48.4**	**47**
Methods	**600**	**700**	**800**	**900**	**1000**
Ensemble features	562	644	721	803	882
ABC-FWMSS	273	299	359	375	444
% of Reduction	**48.6**	**46.4**	**49.8**	**46.7**	**50.3**
% Average: 48.8					

It could be observed from Table 3 that despite the 48.8% reduction in the number of features of our proposed method (ABC-FWMSS), it could still compete favourably with ABCFS in many of the classes. However, there were some noticeable trade-off in 2 out of 10 classes (sci.space and misc.forsale) where ABCFS outperformed our method across the three metrics of comparison. This trade-off between the number of features and fitness score indicates that fewer features have led to a more efficient model with improved generalization but could also risk omitting important features, which might slightly impact precision and recall. This is an area of improvement in our future research.

4.2.4 Statistical significance of ABC-FWMSS

A statistical test that compares the values of the fitness score across intervals was used to verify the robustness of our suggested model. A 95% confidence level based on the P-values was used on the t-test. Table 4 shows the details.

Table 3. ABC-FWMSS and ABCFS across classes on SVM classifier

Classes	Algorithms	Precision	Recall	F-measure
rec.sport.hockey	IG*	0.612	0.605	0.608
	Chi-square*	0.624	0.611	0.617
	ABCFS*	0.668	0.634	0.651
	ABC-FWMSS	**0.85**	**0.74**	**0.79**
rec.motorcycles	IG*	0.529	0.596	0.561
	Chi-square*	0.561	0.602	0.581
	ABCFS*	0.61	0.624	0.617
	ABC-FWMSS	**0.95**	**0.67**	**0.79**
rec.sport.baseball	IG*	0.632	0.569	0.599
	Chi-square*	0.665	0.587	0.624
	ABCFS*	0.715	**0.623**	**0.666**
	ABC-FWMSS	**0.96**	0.5	0.66
rec.autos	IG*	0.72	0.698	0.709
	Chi-square*	0.762	0.705	0.732
	ABCFS*	0.798	**0.723**	**0.759**
	ABC-FWMSS	**0.83**	0.53	0.65
sci.crypt	IG*	0.428	0.589	0.496
	Chi-square*	0.526	0.602	0.561
	ABCFS*	0.611	**0.625**	0.618
	ABC-FWMSS	**0.84**	0.59	**0.69**
sci.med	IG*	0.674	0.632	0.652
	Chi-square*	0.582	0.665	0.621
	ABCFS*	0.671	**0.682**	**0.676**
	ABC-FWMSS	**0.74**	0.45	0.56
comp.windows.x	IG*	0.587	0.715	0.645
	Chi-square*	0.671	0.718	0.694
	ABCFS*	0.703	**0.764**	**0.732**
	ABC-FWMSS	**0.72**	0.59	0.64
sci.space	IG*	0.632	0.635	0.633
	Chi-square*	0.695	0.698	0.696
	ABCFS*	**0.758**	**0.744**	**0.751**
	ABC-FWMSS	0.68	0.73	0.7

Table 4. Statistical performance of ABC-FWMSS with other metaheuristic algorithms

Methods	T-statistic	P-value	Significant
ABC-FWMSS and ABC	4.707142423	0.001108963	+
ABC-FWMSS and PSO	5.442667643	0.000409556	+
ABC-FWMSS and DE	4.72497251	0.001081507	+

Also, one-way ANOVA test was carried out to validate the significance of our proposed method. It recorded F-statistic: 10.350011429716103, p-value: 4.707351786424857e-05. A post-hoc test (Tukey's Honestly Significant Difference (HSD)) was carried out to further probe the strength of our proposed model.

It could be seen that the ABC-FMWSS method has statistical significance with 95% confidence across other metaheuristic algorithms. This further strengthens the justification of the proposed model.

5. CONCLUSIONS

The proposed ABC-FWMSS which contain the ensemble of multi-univariate filter feature selection methods and enhanced ABC algorithm with the pretrained model weight as mutation rate at the employed bee phase, and employed bee index and tournament selection at the onlooker bee phase, has demonstrated the best fitness score compared to other methods. This is in line with our maximization objective function of maximizing accuracy with the fitness score. This was achieved with more than 48.8% reduction in the number of selected features compared to the ensemble features. This has demonstrated that combining both the search operators and selection methods will enhance selection processes for better classification with model efficiency and enhance favorable competition across the classes, balancing exploration and exploitation.

However, the model in the future could be used with imbalanced datasets from other domains to ascertain its generalization and scalability abilities while recall metric could also be improved.

ACKNOWLEDGMENT

The authors appreciate the Editor, Associate Editor, and anonymous reviewers for their valuable feedback, which significantly improved the quality of the paper.

REFERENCES

[1] Abellana, D.P.M., Lao, D.M. (2023). A new univariate feature selection algorithm based on the best–worst multi-attribute decision-making method. Decision Analytics Journal, 7: 100240. https://doi.org/10.1016/j.dajour.2023.100240

[2] Alyasiri, O.M., Cheah, Y.N., Abasi, A.K. (2021). Hybrid filter-wrapper text feature selection technique for text classification. In 2021 International Conference on Communication & Information Technology (ICICT), pp. 80-86. https://doi.org/10.1109/ICICT52195.2021.9567898

[3] Das, H., Naik, B., Behera, H.S. (2022). A Jaya algorithm based wrapper method for optimal feature selection in supervised classification. Journal of King Saud University-Computer and Information Sciences, 34(6): 3851-3863. https://doi.org/10.1016/j.jksuci.2020.05.002

[4] Abiodun, E.O., Alabdulatif, A., Abiodun, O.I., Alawida, M., Alabdulatif, A., Alkhawaldeh, R.S. (2021). A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities. Neural Computing and Applications, 33(22): 15091-15118. https://doi.org/10.1007/s00521-021-06406-8

[5] Khoder, A., Dornaika, F. (2021). Ensemble learning via feature selection and multiple transformed subsets: Application to image classification. Applied Soft Computing, 113: 108006. https://doi.org/10.1016/j.asoc.2021.108006

[6] Grover, P., Chawla, S. (2020). Text feature space optimization using artificial bee colony. In Soft Computing for Problem Solving: SocProS 2018, pp. 691-703. https://doi.org/10.1007/978-981-15-0184-5_59

[7] Mohamed, A.W., Hadi, A.A., Jambi, K.M. (2019). Novel mutation strategy for enhancing SHADE and LSHADE algorithms for global numerical optimization. Swarm and Evolutionary Computation, 50: 100455. https://doi.org/10.1016/j.swevo.2018.10.006

[8] Gao, S., Wang, K., Tao, S., Jin, T., Dai, H., Cheng, J. (2021). A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models. Energy Conversion and Management, 230: 113784. https://doi.org/10.1016/j.enconman.2020.113784

[9] Yu, Y., Gao, S., Wang, Y., Todo, Y. (2019). Global optimum-based search differential evolution. IEEE/CAA Journal of Automatica Sinica, 6(2): 379-394. https://doi.org/10.1109/JAS.2019.1911378

[10] Xu, Z., Yang, H., Li, J., Zhang, X., Lu, B., Gao, S. (2021). Comparative study on single and multiple chaotic maps incorporated grey wolf optimization algorithms. IEEE Access, 9: 77416-77437. https://doi.org/10.1109/ACCESS.2021.3083220

[11] Yang, X.S. (2020). Nature-inspired optimization algorithms: Challenges and open problems. Journal of Computational Science, 46: 101104. https://doi.org/10.1016/j.jocs.2020.101104

[12] Wang, Y., Gao, S., Zhou, M., Yu, Y. (2020). A multi-layered gravitational search algorithm for function optimization and real-world problems. IEEE/CAA Journal of Automatica Sinica, 8(1): 94-109. https://doi.org/10.1109/JAS.2020.1003462

[13] Wang, J., Sun, Y., Zhang, Z., Gao, S. (2020). Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms. IEEE/CAA Journal of Automatica Sinica, 7(4): 1134-1153. https://doi.org/10.1109/JAS.2020.1003204

[14] Katoch, S., Chauhan, S.S., Kumar, V. (2021). A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications, 80: 8091-8126. https://doi.org/10.1007/s11042-020-10139-6

[15] Mayet, A.M., Ijyas, V.T., Bhutto, J.K., Guerrero, J.W.G., Shukla, N.K., Eftekhari-Zadeh, E., Alhashim, H.H. (2023). Using ant colony optimization as a method for selecting features to improve the accuracy of measuring the thickness of scale in an intelligent control system. Processes, 11(6): 1621. https://doi.org/10.3390/pr11061621

[16] Kumar, V., Aydav, P.S.S., Minz, S. (2022). Multi-view ensemble learning using multi-objective particle swarm optimization for high dimensional data classification. Journal of King Saud University-Computer and Information Sciences, 34(10): 8523-8537. https://doi.org/10.1016/j.jksuci.2021.08.029

[17] Al-Betar, M.A., Alomari, O.A., Abu-Romman, S.M. (2020). A TRIZ-inspired bat algorithm for gene selection in cancer classification. Genomics, 112(1): 114-126. https://doi.org/10.1016/j.ygeno.2019.09.015

[18] Bazi, S., Benzid, R., Bazi, Y., Rahhal, M.M.A. (2021). A fast firefly algorithm for function optimization: Application to the control of BLDC motor. Sensors, 21(16): 5267. https://doi.org/10.3390/s21165267

[19] Akay, B., Karaboga, D., Gorkemli, B., Kaya, E. (2021). A survey on the artificial bee colony algorithm variants for binary, integer and mixed integer programming problems. Applied Soft Computing, 106: 107351. https://doi.org/10.1016/j.asoc.2021.107351

[20] Zhang, M., Palade, V., Wang, Y., Ji, Z. (2021). Attention-based word embeddings using artificial bee colony algorithm for aspect-level sentiment classification. Information Sciences, 545: 713-738. https://doi.org/10.1016/j.ins.2020.09.038

[21] Nekoei, M., Moghaddas, S.A., Golafshani, E.M., Gandomi, A.H. (2021). Introduction of ABCEP as an automatic programming method. Information Sciences, 545: 575-594. https://doi.org/10.1016/j.ins.2020.09.020

[22] Zhou, X., Lu, J., Huang, J., Zhong, M., Wang, M. (2021). Enhancing artificial bee colony algorithm with multi-elite guidance. Information Sciences, 543: 242-258. https://doi.org/10.1016/j.ins.2020.07.037

[23] Bajer, D., Zorić, B. (2019). An effective refined artificial bee colony algorithm for numerical optimisation. Information Sciences, 504: 221-275. https://doi.org/10.1016/j.ins.2019.07.022

[24] Jain, M., Singh, V., Rani, A. (2019). A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm and Evolutionary Computation, 44: 148-175. https://doi.org/10.1016/j.swevo.2018.02.013

[25] Han, X., Liu, Q., Wang, H., Wang, L. (2018). Novel fruit fly optimization algorithm with trend search and co-evolution. Knowledge-Based Systems, 141: 1-17. https://doi.org/10.1016/j.knosys.2017.11.001

[26] Piotrowski, A.P., Napiorkowski, J.J. (2018). Step-by-step improvement of JADE and SHADE-based algorithms: Success or failure?. Swarm and Evolutionary Computation, 43: 88-108. https://doi.org/10.1016/j.swevo.2018.03.007

[27] Cui, L., Li, G., Zhu, Z., Lin, Q., Wong, K. C., Chen, J., Lu, N., Lu, J. (2018). Adaptive multiple-elites-guided composite differential evolution algorithm with a shift mechanism. Information Sciences, 422: 122-143. https://doi.org/10.1016/j.ins.2017.09.002

[28] Meidani, K., Mirjalili, S., Farimani, A. B. (2022). Online metaheuristic algorithm selection. Expert Systems with Applications, 201: 117058. https://doi.org/10.1016/j.eswa.2022.117058

[29] Ali, M.Z., Awad, N.H., Reynolds, R.G., Suganthan, P.N. (2018). A balanced fuzzy cultural algorithm with a modified levy flight search for real parameter optimization. Information Sciences, 447: 12-35. https://doi.org/10.1016/j.ins.2018.03.008

[30] Gao, W.F., Liu, S.Y. (2012). A modified artificial bee colony algorithm. Computers & Operations Research, 39(3): 687-697. https://doi.org/10.1016/j.cor.2011.06.007

[31] Zhou, X., Wu, Z., Wang, H., Rahnamayan, S. (2016). Gaussian bare-bones artificial bee colony algorithm. Soft Computing, 20: 907-924. https://doi.org/10.1007/s00500-014-1549-5

[32] Aslan, S., Badem, H., Karaboga, D. (2019). Improved quick artificial bee colony (iqABC) algorithm for global optimization. Soft Computing, 23: 13161-13182. https://doi.org/10.1007/s00500-019-03858-y

[33] Zhou, B., Zhao, Z. (2023). An adaptive artificial bee colony algorithm enhanced by Deep Q-Learning for milk-run vehicle scheduling problem based on supply hub. Knowledge-Based Systems, 264: 110367. https://doi.org/10.1016/j.knosys.2023.110367

[34] Wang, H., Wang, W., Xiao, S., Cui, Z., Xu, M., Zhou, X. (2020). Improving artificial bee colony algorithm using a new neighborhood selection mechanism. Information Sciences, 527: 227-240. https://doi.org/10.1016/j.ins.2020.03.064

[35] Xiao, S., Wang, H., Wang, W., Huang, Z., Zhou, X., Xu, M. (2021). Artificial bee colony algorithm based on adaptive neighborhood search and Gaussian perturbation. Applied Soft Computing, 100: 106955. https://doi.org/10.1016/j.asoc.2020.106955

[36] Zhou, X., Wu, Y., Zhong, M., Wang, M. (2022). Artificial bee colony algorithm based on adaptive neighborhood topologies. Information Sciences, 610: 1078-1101. https://doi.org/10.1016/j.ins.2022.08.001

[37] Xiao, S., Wang, W., Wang, H., Tan, D., Wang, Y., Yu, X., Wu, R. (2019). An improved artificial bee colony algorithm based on elite strategy and dimension learning. Mathematics, 7(3): 289. https://doi.org/10.3390/math7030289

[38] Xue, Y., Jiang, J., Zhao, B., Ma, T. (2018). A self-adaptive artificial bee colony algorithm based on global best for global optimization. Soft Computing, 22: 2935-2952. https://doi.org/10.1007/s00500-017-2547-1

[39] Chen, X., Tianfield, H., Li, K. (2019). Self-adaptive differential artificial bee colony algorithm for global optimization problems. Swarm and Evolutionary Computation, 45: 70-91. https://doi.org/10.1016/j.swevo.2019.01.003

[40] Chen, C.F., Zain, A.M., Mo, L.P., Zhou, K.Q. (2020). A new hybrid algorithm based on ABC and PSO for function optimization. In IOP Conference Series: Materials Science and Engineering, 864(1): 012065. https://doi.org/10.1088/1757-899X/864/1/012065

[41] Ustun, D., Toktas, A., Erkan, U., Akdagli, A. (2022). Modified artificial bee colony algorithm with differential evolution to enhance precision and convergence performance. Expert Systems with Applications, 198: 116930. https://doi.org/10.1016/j.eswa.2022.116930

[42] Hashemi, A., Dowlatshahi, M.B., Nezamabadi-pour, H. (2021). A pareto-based ensemble of feature selection algorithms. Expert Systems with Applications, 180: 115130. https://doi.org/10.1016/j.eswa.2021.115130

[43] IGE, O., Gan, K.H. (2024). Ensemble feature selection using weighted concatenated voting for text classification. Journal of the Nigerian Society of Physical Sciences, 1823-1823. https://doi.org/10.46481/jnsps.2024.1823

[44] Fu, G., Li, B., Yang, Y., Li, C. (2023). Re-ranking and

TOPSIS-based ensemble feature selection with multi-stage aggregation for text categorization. Pattern Recognition Letters, 168: 47-56. https://doi.org/10.1016/j.patrec.2023.02.027

[45] Too, J., Abdullah, A.R., Mohd Saad, N., Tee, W. (2019). EMG feature selection and classification using a Pbest-guide binary particle swarm optimization. Computation, 7(1): 12. https://doi.org/10.3390/computation7010012

[46] Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M.A., Omar, M.S. (2021). Dynamic salp swarm algorithm for feature selection. Expert Systems with Applications, 164: 113873. https://doi.org/10.1016/j.eswa.2020.113873

[47] Prabhakar, S.K., Rajaguru, H., So, K., Won, D.O. (2022). A framework for text classification using evolutionary contiguous convolutional neural network and swarm based deep neural network. Frontiers in Computational Neuroscience, 16: 900885. https://doi.org/10.3389/fncom.2022.900885