Adaptive BMRFO: Optimizing Descriptor Selection for Improved QSAR Biodegradation Classification

Nor Fatin Nabila Aziz, Norfadzlia Mohd Yusof, Farhad Soleimanian Gharehchopogh and Satrya Fajri Pratama

Abstract— **This study presents an enhanced Adaptive Binary Manta Ray Foraging Optimization (ABMRFO) algorithm for feature selection in QSAR biodegradation classification. The proposed method integrates an adaptive Sigmoid transfer function for binary conversion and a dynamic somersault factor for improved search efficiency. Nine ABMRFO variants were tested on the QSAR Biodegradation dataset using hold-out validation over 150 iterations, evaluated based on classification accuracy, convergence speed, fitness value, and computational efficiency. Among the evaluated variants, ABMRFO3 emerged as the top performer, achieving the highest classification accuracy of 90.38% while selecting an average of only 9.2 features. It demonstrated strong optimization capabilities with the lowest mean fitness value (0.0975), best fitness (0.0548), and worst fitness (0.1193). Its fast convergence was evidenced by an average computational time of 27.92 seconds. The Friedman test ranked ABMRFO3 first with a sum of ranks of 19, confirming its superior performance. Additionally, the Wilcoxon signed-rank test indicated statistically significant improvements of ABMRFO3 over other algorithms, further validating its effectiveness. Its adaptive mechanisms ensure exceptional search accuracy, computational efficiency, and solution stability, making it a robust solution for complex feature selection tasks in QSAR modeling.**

Index Terms—**Adaptive BMRFO, Descriptor selection, Biodegradation classification, QSAR modelling**

I. Introduction

CHEMINFORMATICS is a field within information technology focused on managing, analyzing, and manipulating chemical data, such as small molecule structures, properties, and biological activities [1]. Within cheminformatics, a central research focus is Quantitative Structure-Activity Relationship (QSAR) modeling, which predicts chemical compounds' behavior based on their structural attributes using mathematical, statistical, and informatics methods [2]–[4]. One critical application of QSAR modeling is in assessing the biodegradability of chemicals. This vital environmental parameter determines how quickly and efficiently a substance can be broken down by natural biological processes, such as microbial activity. Biodegradability is essential for minimizing the long-term accumulation of harmful chemicals in the environment, as substances that are not readily biodegradable can persist, leading to ecological and health risks [5] and [6]. QSAR models have been widely used to predict whether a chemical is readily biodegradable or not, utilizing molecular descriptors such as chemical functional groups, structural properties, and physicochemical data [6]. These models help reduce the need for extensive experimental testing, which is both costly and time-consuming [5] and [7]. Despite the utility of QSAR models, they face significant challenges when dealing with high-dimensional datasets, which often result in overfitting and poor generalization [3]–[8].

Feature selection (FS) has emerged as an effective solution to mitigate these challenges [9] and [10]. However, traditional methods often become impractical for high-dimensional datasets due to their high computational complexity and inefficiency [11]. FS is inherently a non-deterministic polynomial-time (NP-hard) problem [12], meaning it requires exponential time to find the optimal subset of features. To overcome these limitations, FS is often reformulated as an optimization problem, where swarm intelligence (SI) algorithms are widely employed to explore the solution search space more efficiently [13]–[15]. A very accurate search for the solution is expected when SI, which is a technique for quickly implementing a global search, is integrated into the solution search [16].

One widely used SI algorithm is the Manta Ray Foraging Optimization (MRFO) algorithm [17]. Although effective, MRFO has certain limitations, particularly its ability to exploit complex problem spaces fully. In addition, the No-Free-Lunch (NFL) theorem, which was developed by [18], claims that there is no one method that can effectively address all optimization issues [18]. The standard MRFO algorithm, initially designed for continuous optimization tasks, may struggle with FS problems that require navigating

Manuscript received October 2, 2024; revised March 14, 2025.

This study is supported by the Ministry of Higher Education (MOHE) of Malaysia through the Fundamental Research Grant Scheme (FRGS) under Grant No: FRGS/1/2023/ICT02/UTEM/03/1.

Nor Fatin Nabila Aziz is a postgraduate student of Fakulti Teknologi Maklumat and Komunikasi, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia. (e-mail: m032310054@student.utem.edu.my).

Norfadzlia Mohd Yusof is a senior lecturer and researcher of the Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka, 76100 Durian Tunggal, Melaka, Malaysia (corresponding author to provide e-mail: norfadzlia@utem.edu.my).

Farhad Soleimanian Gharehchopogh is an associate professor and a researcher in Computer Engineering Department, Urmia Branch, Islamic Azad University, Urmia, Iran. (e-mail: bonab.farhad@gmail.com).

Satrya Fajri Pratama is a senior lecturer and a researcher in Department of Computer Science, School of Physics, Engineering & Computer Science, University of Hertfordshire, United Kingdom. (e-mail: s.f.pratama@herts.ac.uk).

binary solution spaces.

This limitation underscores the need to enhance the conventional MRFO algorithm to better adapt it for FS tasks. This paper introduces an enhanced binary MRFO (BMRFO), integrating an adaptive Sigmoid transfer function and a dynamic somersault factor to improve. It aims to strengthen exploration-exploitation trade-offs and reduce entrapment in suboptimal solutions. Our algorithm is evaluated on high-dimensional QSAR datasets, showing significant improvements in both accuracy and computational efficiency.

In this paper, we highlight the following key contributions:

1) Optimization Framework for Feature Selection: We formulate an optimization problem aimed at minimizing the number of selected features while enhancing classification accuracy, striking a balance between dimensionality reduction and predictive performance.
2) Introduction of Adaptive Binary Manta Ray Foraging Optimization (ABMRFO): We propose the ABMRFO algorithm specifically tailored for feature selection tasks. This algorithm incorporates a dynamic somersault foraging factor and an adaptive transfer function as its binary mechanism, extending the conventional MRFO algorithm for improved performance in binary optimization.
3) Comprehensive Performance Evaluation: We evaluate the performance of nine variants of ABMRFO algorithm using the QSAR Biodegradation dataset from the University of California, Irvine (UCI) Machine Learning Repository. The results are compared against the conventional MRFOv3 algorithm, validating the enhancements introduced by ABMRFO through benchmarking with state-of-the-art algorithms in the field.

The remainder of this paper is organized as follows: Section II explains the related works of the study. Section III describes the methodology, covering the MRFO algorithm, the proposed adaptive BMRFO algorithm, and the experimental setup. Section IV presents the results and discussion, providing a detailed analysis of the findings. Finally, Section V concludes the paper by summarizing the key insights and contributions.

II. Related works

The idea behind QSAR is to convert the process of searching for compounds with desired qualities, which is based on chemical intuition and experience, into a mathematically quantified and automated form [19]. Molecular descriptors can be used to describe the molecular composition in computational method. Practically, molecular descriptors are chemical information recorded within molecular structures for which numerous sets of methods are available for transformation. Once the chemical descriptors are determined, they will be used as independent variables in the QSAR model. The activities that can be modeled by QSAR are dependent variables of the QSAR model. These dependent variables are expected to be influenced by the independent variables which are molecular descriptors. Environmental applications of the QSAR model have expanded to include testing materials' biodegradability without requiring chemical reactions. The structures of the related compounds and experimental biodegradation data are used to generate QSAR models for biodegradation. The ability to produce statistically meaningful predictions and infer the physicochemical characteristics of the corresponding molecules are two benefits of QSAR models [19]–[21]. Limited works have employed QSAR models to estimate biodegradability of chemical compounds as displayed in Table I [20].

TABLE I
Related Work On The Classification Of Biodegradable Substances

Ref	Machine Learning Model
[22]	Four QSAR models were developed for predicting primary and ultimate biodegradation rate rating with multiple linear regression (MLR) and support vector regression (SVR) algorithms.
[23]	SVM models with linear and RBF kernels, random forest (RF), and Naïve Bayesian (NB) and their Ensemble.

There are minimum works that apply QSAR models in binarization approach for MRFO. Recent years have seen an increase in the use of binary MRFO in numerous applications; some of the works are given in Table II.

TABLE II
Related Work On The Bmrfo Algorithm

Ref	Variant	Algorithms	Application
[24] and [25]	Binary	BWOA, BPSO, BGWOA, and BMRFO	On the chemical dataset, the gathered results demonstrated that the proposed approaches performed well. One of these findings was near-optimal convergence, along with improved classification accuracy, accelerated processing, and a substantial reduction in descriptor size.
[26]		BMRFO	This methodology was validated with the NSL-KDD and CIC-IDS2017 intrusion detection datasets and was tested against GA, PSO, GWO, and GOA. A statistical significance test indicates a significant difference between the proposed model and the comparative approaches for F-measure.
[27]		MRFO	Comparing the BMRFO against other methods in the literature, the results show that it achieved the maximum accuracy with less attributes.

Dynamic parameter adjustment has significantly improved algorithm performance in recent years [28]. To avoid early convergence, for example, a study from [29] dynamically modified the somersault factor, while a further investigation from [30] proposed a spirally element to improve search efficiency inside a specified region. Similarly to [31], the author has modified the r value during the chain foraging phase, using a higher value in the early stages to enable longer strides and speed up population convergence toward the optimal solution. In [32], the author addressed optimization challenges by presenting an adaptive-somersault version of the MRFO approach to improve both local and global search abilities. To improve the original MRFO's optimization efficiency, the suggested method combines an adaptive sine-based position updating mechanism. An adaptive scheme of control factor, which included the somersault factor, was incorporated in [33], in order to broaden the exploration range of the MRFO during the early phase and to enhance its capabilities of exploitation

during the later phase. In [34], the IBMRFO approach is presented as a solution to the joint feature selection issue. It enhances the conventional MRFO by including a binary approach, an adaptive element of somersault foraging, and a chaotic map of tent for population initiation, which increases its efficacy for feature selection.

III. Methodology

A. Manta Ray Foraging Optimization (MRFO) Algorithm

The MRFO algorithm was first introduced in 2020 [17], drawing inspiration from the food-searching behavior of manta rays. The MRFO employed three unique search strategies to address continuous optimization problems.

Chain Foraging

In the MRFO algorithm, manta rays detect plankton and move toward areas with higher concentrations, assuming the best solution is like the most concentrated plankton. They form a chain, where each ray follows both the best-known solution and the one ahead of it, updating their position in every iteration. The mathematical model of chain foraging represented in (1).

$$x_i^d(t+1)=\begin{cases} x_i^d(t)+r\left(x_{best}^d(t)-x_i^d(t)\right)+\alpha\left(x_{best}^d(t)-x_i^d(t)\right), & i=1 \\ x_i^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\alpha\left(x_{best}^d(t)-x_i^d(t)\right), & i=2,\ldots,N \end{cases} \quad (1)$$

$$\alpha=2r\sqrt{|\log(r)|} \quad (2)$$

where x denotes the position of the search agent or manta ray. i is the order of the manta ray, d denotes the search space dimension, t the iteration number, and r is a random vector in [0,1]. α represents the weight coefficient. The position with the highest plankton concentration is denoted as x_{best}^d and it is assumed as the best solution in MRFO.

Cyclone Foraging

The search agent, a manta ray, swims in a spiral form toward the plankton and connects with other manta rays via a head-to-tail link. Equation (3) provided the following definition for the mathematical description of the spiral-shaped movement.

$$x_i^d(t+1)=\begin{cases} x_{best}^d(t)+r\left(x_{best}^d(t)-x_i^d(t)\right)+\beta\left(x_{best}^d(t)-x_i^d(t)\right) & i=1 \\ x_{best}^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\beta\left(x_{best}^d(t)-x_i^d(t)\right) & i=2,\ldots,N \end{cases} \quad (3)$$

$$\beta=2e^{r_1\frac{T-t+1}{T}}\sin(2\pi r_1) \quad (4)$$

where T is the maximum number of iterations, β is a weight coefficient and r_1 is a random vector in [0, 1]. A new random position that is far from the current best one is assigned to each search agent to promote an extensive global search in MRFO. Equation (5) expressed the mathematical model.

$$x_{rand}^d=Lb^d+r\bullet Ub^d-Lb^d \quad (5)$$

$$x_i^d(t+1)=\begin{cases} x_{rand}^d(t)+r\left(x_{rand}^d(t)-x_i^d(t)\right)+\beta\left(x_{rand}^d(t)-x_i^d(t)\right), & i=1 \\ x_{rand}^d(t)+r\left(x_{i-1}^d(t)-x_i^d(t)\right)+\beta\left(x_{rand}^d(t)-x_i^d(t)\right), & i=2,\ldots,1 \end{cases} \quad (6)$$

where x_{rand}^d indicates the search agent random position, Lb^d and Ub^d are lower and upper boundaries and d denotes the dimension of the search space.

Somersault Foraging

The spot of the best plankton discovered thus far is used as a pivot. Every search agent turns around and swims back and forth before somersaulting into a new location. Equation (7) displayed the model mathematically.

$$x_i^d(t+1)=x_i^d(t)+S\,(r_2\bullet x_{best}^d-r_3\bullet x_i^d(t)),\ i=1,2,\ldots,N \quad (7)$$

where S is the somersault factor, r_2, r_3 are random numbers in [0,1].

B. Adaptive Binary MRFO Algorithm

This section explains the enhancements to the BMRFO algorithm, explicitly focusing on the adaptive transfer function and the dynamic somersault factor.

Adaptive Transfer Function

To convert the continuous MRFO into a binary one, this study utilizes an adaptive transfer function (TF). This study uses an adaptive modified Sigmoid transfer function within the MRFO algorithm. Originally proposed in [35], this function effectively converts the real-valued position data of search agents into probability values between 0 and 1. The authors demonstrated significant improvements in selecting informative descriptors and achieving high classification accuracy compared to the standard Sigmoid transfer function. The transfer function is represented in (8).

$$SIG-Tv\left(x_i^d(t+1)\right)=\frac{1}{1+e^{-10\left(\frac{x_d^i(t+1)}{Tv}-0.5\right)}} \quad (8)$$

where x denotes the position of the search agent or manta ray, i is the order of the manta ray, d denotes the search space dimension and t is the iteration number. Tv is a time-varying control parameter provides an adaptive behavior to the standard modified Sigmoid transfer function [36]. Next, the binarization rule in (9) from [37] is utilized to update the position vector based on the calculated probability values obtained in (8). The equation is presented below.

$$x_i^d(t+1)=\begin{cases} 1, if\ rand\ <\ Sigmoid\left(x_d^i(t+1)\right) \\ 0, otherwise \end{cases} \quad (9)$$

with $rand$ signifying a random number drawn from the uniform distribution [0,1].

This study investigates the efficacy of three (3) Tv strategies within MRFO algorithm. This strategy enables the exploration of different regions within the search space, enhancing the algorithm's ability to identify the global optimum. Choosing an appropriate Tv formulation for the transfer function (TF) is essential in MRFO to prevent early convergence and avoid getting stuck in local minima. The

equations for Tv strategies are outlined in Table III.

TABLE III
DIFFERENT Tv FORMULATIONS

TF	Tv
*SIG-Tv*1	$t_{max} - (t_{max} - t_{min}) \times (t/T)$
*SIG-Tv*2	$t_{max} - (t_{max} - t_{min}) \times (t/T)^{\alpha}, \alpha = 0.5$
*SIG-Tv*3	$t_{max} - (t_{max} - t_{min}) \times log_{10}(1 + 9 \times t/T)$

The $SIG\text{-}Tv1$ approach follows a linear adaptive strategy, consistent with the findings of prior studies by [3] and [38]–[40]. The second approach, labeled as $SIG\text{-}Tv2$, employs a non-linear adaptive method, as supported by the findings in [41]. Furthermore, the third approach, referred to as $SIG\text{-}Tv3$, utilizes a logarithmic adaptive technique, originally introduced by [42]. In these strategies, the variable t represents the current iteration number, while T denotes the maximum number of iterations permitted in the optimization process. Additionally, t_{max} and t_{min} correspond to the maximum and minimum allowable values for the control parameter Tv, respectively. The value of Tv decreases systematically over successive iterations, facilitating adaptive control. For this research, specific parameter values were selected based on established findings in prior studies. The maximum value t_{max} was set at 4, while the minimum value t_{min} was assigned a value of 0.01, following the recommendations from [39]. The parameter α, influencing the adaptation rate, was chosen as 0.5, in line with findings reported in [38]. These parameter choices ensure a robust and dynamically responsive optimization process, guided by well-established theoretical foundation.

Dynamic Somersault Factor

In the conventional BMRFO algorithm, the somersault factor S is fixed at 2. In this paper, the manta ray agent updates its somersault position using a dynamic somersault factor based on the same adaptive strategies employed in the adaptive transfer function. This enhancement improved the agents' exploration and exploitation capabilities as the search progressed. The specific equations governing the dynamic somersault factor are detailed in Table IV.

TABLE IV
DIFFERENT S FORMULATIONS

S	Mathematical Equation
S_1	$t_{maxs} - (t_{maxs} - t_{mins}) \times (t/T)$
S_2	$t_{maxs} - (t_{maxs} - t_{mins}) \times (t/T)^{\alpha}, \alpha = 0.5$
S_3	$t_{maxs} - (t_{maxs} - t_{mins}) \times log_{10}(1 + 9 \times t/T)$

The parameters t_{maxs} and t_{mins} represents the maximum and minimum allowable values, respectively, for the dynamic control parameter S [33]. In this research, the specific values for t_{maxs}, t_{mins}, and α were set to 2, 0.01 and 0.5, respectively, as originally introduced by [17], [38] and [33]. The proposed dynamic S factor decreases progressively over time. Fig. 1 shows the original and proposed S-factor values over 150 iterations. The work aims to enhance the BMRFO algorithm by incorporating adaptive and dynamic mechanisms as illustrated in Fig. 2.

Fitness Function

The two main goals of improving classification accuracy and reducing the number of features is used to evaluate the feature selection algorithm's efficacy. In order to balance these goals, this feature selection method uses a fitness function. The chosen feature subsets are evaluated using a wrapper-based feature selection technique that employs a K-Nearest Neighbors (K-NN) classifier with the Euclidean distance metric where k = 5. The well-known categorization method K-NN is renowned for its simplicity, quickness, and ease of use [43]–[45]. The K-NN classifier's classification error rate obtained during the evaluation is incorporated into the fitness function formulation in Eq.(10). The optimization algorithm leverages a fitness function to assess candidate feature subsets based on two key criteria: the classification error rate and the number of selected features. An optimal subset achieves both a low error rate and a minimal feature count, which is reflected in a lower fitness value indicating a more relevant and efficient selection.

$$\downarrow Fitness = \alpha \times CE + \beta \times \frac{|F_{select}|}{|F_{actual}|} \tag{10}$$

In (10), CE represents the classification error rate determined by the classifier. $|F_{select}|$ indicates the number of selected features, while $|F_{actual}|$ denotes the total number of original features. The parameters, α and β correspond to the significance of classification quality and subset length, respectively. The values $\alpha \in [1, 0]$ and $\beta = (1 - \alpha)$ are derived from [23] and [24]. For this work, the classification performance is prioritize as the most importance metric, thus α is set to 0.99, and β is set to 0.01 following guidelines from [46] and [48].

C. Experimental Settings

This section explains the dataset, parameter settings, and performance measurements used in this study.

Dataset

This study utilized QSAR Biodegradation dataset by [6] obtained from University of California, Irvine (UCI) Machine Learning Repository. Data pre-processing was performed using Weka Explorer where the nominal class labels: RB (readily biodegradable) and NRB (not readily biodegradable) were converted into binary values, 0 and 1, respectively, to enable machine learning models to classify chemicals into two distinct classes. A detailed description of the dataset is provided in Table V below.

TABLE V
DATASET DESCRIPTION

Dataset Name	Feature size	Instance size	No. of classes
QSAR Biodegradation	41	1055 RB (366) / NRB (699)	2

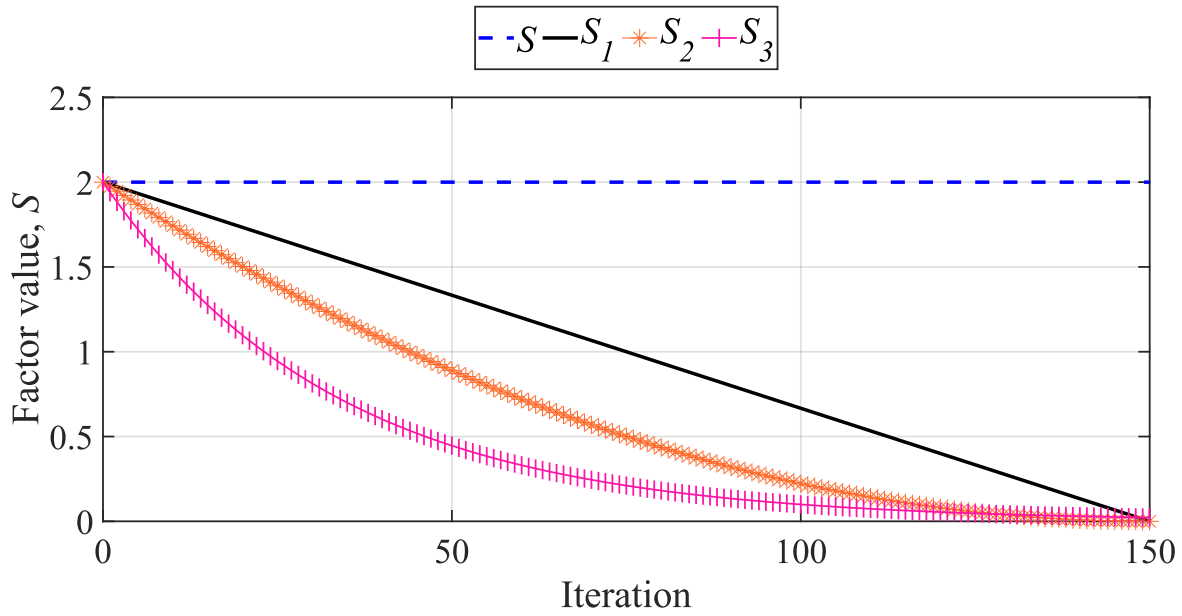

Fig. 1. Fixed S factor and proposed dynamic S_1, S_2, and S_3 factor values.

```
START
  #Step 1: Initialize parameters
  Define time-varying parameters: t_max, t_min
  Define dynamic somersault factor parameters: t_maxs, t_mins
  Initialize N, T, Tv, S, Ub, Lb
  #Step 2: Generate a uniform random initial population X of manta rays within [lb, ub]
  FOR i FROM 1 TO N DO
    X(i,d) = lb + (ub - lb) * rand(0,1)
  ENDFOR
  #Step 3: Compute fitness of each manta-ray
  Evaluate fitness of each manta-ray
  Identify the best solution Xbest with the lowest fitness value Fitbest
  #Step 4: Initialize iteration counter
  t = 1
  #Step 5: Optimization loop
  WHILE t <= T DO
    #Update Time-Varying and Dynamic Somersault factor Parameters
    Update Tv formulations
    Update S formulations
    #Decision making based on probability
    IF r < 0.5 THEN
      Update position of manta-rays using Equation (1)
    ELSE
      IF (t/T) < r2 THEN
        Update position of manta-rays using Equation (5)
      ELSE
        Update position of manta-rays using Equation (3)
      ENDIF
    ENDIF
    #Mapping and binarization
    Map manta-ray position to probability using transfer function
    Transform probability value to binary using binarization rule
    Evaluate fitness of updated manta-rays
    Update global best manta-ray
    #Somersault foraging update
    Update position of manta-rays using Equation (7)
    Map manta-ray position to probability using transfer function
    Transform probability value to binary using binarization rule
    Evaluate fitness of updated manta-rays
    #Increment iteration counter
    t = t + 1
  ENDWHILE
  #Step 6: Return the best solution
  RETURN Xbest
END
```

Fig. 2. Pseudocode of the proposed ABMRFO algorithm.

Parameter Setting

In the experiment, each algorithm was run ten times using a variety of random seeds to ensure reproducibility and provide different data partitioning. This is one way to inspect the robustness of the proposed algorithm. Each run was subjected to 150 iterations. A stratified hold-out validation technique, similar to those in [49] and [43], was used, dividing the data into 80% for training and 20% for testing. During optimization, the classifier was trained on the training data, and the testing data was used to evaluate the performance of the selected features. TABLE VI displays nine variants of ABMRFO algorithm with different combination of transfer functions and somersault factors. The results are average metrics calculated from ten independent runs to account for potential variations. The algorithm was developed and executed using MATLAB R2023b on a DESKTOP-AS4B9A8 with an Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz, running on Windows 11 Home Single Language and equipped with 8 GB of RAM. Additionally, we conducted feature selection using eight state-of-the-art swarm intelligence (SI) optimization algorithms, applying the parameter settings outlined in TABLE VII. The performance of these algorithms was compared against our proposed method to evaluate its effectiveness and superiority in feature selection tasks.

TABLE VI
DIFFERENT VARIANTS OF THE PROPOSED ABMRFO ALGORITHM

Algorithm	TF	S
ABMRFO1	*SIG-Tv*1	S_1
ABMRFO2	*SIG-Tv*2	S_1
ABMRFO3	*SIG-Tv*3	S_1
ABMRFO4	*SIG-Tv*1	S_2
ABMRFO5	*SIG-Tv*2	S_2
ABMRFO6	*SIG-Tv*3	S_2
ABMRFO7	*SIG-Tv*1	S_3
ABMRFO8	*SIG-Tv*2	S_3
ABMRFO9	*SIG-Tv*3	S_3

TABLE VII
PARAMETER SETTINGS

Algorithm	Parameter	Value
All	Search agent size, *N*	15
	Iteration length, *T*	150
	No. of runs, *R*	10
	Problem dimension	41
	α in the fitness function	0.99
	β in the fitness function	0.01
MRFOv3 [27] BMRFO [25] $BMRFO_{TV1}$ [24]	Somersault factor, *S*	2
BCSA [50]	Flight length, *FL*	1.8
	Awareness probability, *AP*	0.1
bGOW2 [51]	-	-
BHHO-TV1 [39]	Escaping energy, *E*	2 to 0
BWOA-3 [35] $BWOA_{TV2}$ [24]	*a*	2 to 0

Performance Measurement

This study investigates how well the suggested improved BMRFO algorithm performs by evaluating numerous performance measurement metrics. The performance analysis metrics are described as follows:

1) Average fitness ($Mean_{Fit}$) is used to evaluate the convergence ability.
2) Best fitness ($Best_{Fit}$) is used to determine the minimum fitness value obtained from 10 run times.
3) Worst fitness ($Worst_{Fit}$) is used to determine the maximum fitness value obtained from 10 run times.
4) Average classification accuracy (Avg_{Acc}) reflects how effectively the classifier can classify data using the descriptors selected by the algorithm.
5) The average number of selected descriptors (Avg_{Nd}) indicates the quantity of relevant descriptors chosen by the algorithm.
6) Average precision (Avg_{Pre}) measures the positive patterns correctly predicted from the total predicted patterns in a positive class.
7) The Average F1 Score ($Avg_{F1\text{-}Score}$) evaluates the degree of accuracy, reflecting the balance between sensitivity and precision.
8) In seconds, average computational time (Avg_{CT}) measures the convergence speed of the algorithm, representing the time required to complete the search and selection process.

Moreover, two non-parametric statistical analysis methods, Friedman-test (FT) and Wilcoxon signed-rank test were employed. The Friedman test was employed to identify significant differences in rankings across multiple performance metrics when evaluating various SI algorithms. This approach ensures a robust comparison by accounting for differences in performance across algorithms under similar experimental conditions, highlighting statistically significant variations in their effectiveness. The Wilcoxon signed-rank test at a significance level of 0.05, is conducted to determine whether significant performance differences exist between the proposed enhanced BMRFO algorithm with existing BMRFO algorithms and other state-of-the-art SI algorithms. The null hypothesis, which asserts no considerable difference between the two algorithms, is accepted if the p-value is greater than or equal to 0.05. Conversely, if the p-value is less than 0.05, the null hypothesis is rejected.

IV. RESULT AND DISCUSSION

Several evaluations were conducted in this section. The first evaluation aimed to determine the optimal combination of the two proposed enhancements: an adaptive transfer function and a dynamic somersault factor, to identify the most effective algorithm for classifying the biodegradability of substances. This study employed the Friedman-Test ranking to assist in selecting the optimal algorithm. Table VIII presents the convergence performance of the proposed ABMRFO algorithms. The results show that ABMRFO3 and ABMRFO9 have the same and lowest mean fitness outperformed the other variants. Furthermore, ABMRFO3 is seen to produce the lowest best, and worst fitness values, demonstrating its strong ability to converge near the optimal solution while effectively avoiding local minima. Additionally, all the algorithms displayed robustness and consistency, with low fitness standard deviations close to zero. Among all algorithms, ABMRFO3 exhibited the highest fitness standard deviation. However, it demonstrated

the fastest convergence, with an average computational time of 27.92 seconds, highlighting its superior computational efficiency. This is further validated by the Friedman test results, which ranked ABMRFO3 as the top-performing algorithm.

Table IX provides a detailed analysis of the experiment based on classification performance metrics: Avg_{Acc}, Avg_{Pre} and $Avg_{F1\text{-}Score}$ and Avg_{Nd}. The results indicate that ABMRFO3 is the top-performing variant, achieving the highest average classification accuracy and F1-Score, with values of 90.38% and 85.79%, respectively. While ABMRFO3 ranked second in average precision, behind ABMRFO5, it still demonstrated robust overall performance. Although ABMRFO4 selected the fewest features, it did not deliver strong classification results. In contrast, ABMRFO3 selected the second minimal descriptors while maintaining superior classification accuracy, underscoring its effectiveness in feature selection. According to FT results, ABMRFO3 ranks first, indicating that the algorithm demonstrates superior performance.

TABLE VIII
CONVERGENCE PERFORMANCE OF PROPOSED ABMRFO ALGORITHMS AND FRIEDMAN-TEST (FT) RANKING RESULTS

Algorithm	$Mean_{Fit}$	$Best_{Fit}$	$Worst_{Fit}$	Std_{Fit}	Avg_{CT}	FT
ABMRFO1	0.1015	0.0736	0.1472	0.0188	51.58	8
ABMRFO2	0.1037	0.0681	0.1336	0.0194	29.18	3
ABMRFO3	**0.0975**	**0.0548**	**0.1193**	0.0202	**27.92**	**1**
ABMRFO4	0.1085	0.0906	0.1380	**0.0137**	30.16	7
ABMRFO5	0.1024	0.0738	0.1247	0.0167	29.06	2
ABMRFO6	0.1016	0.0763	0.1432	0.0192	34.18	9
ABMRFO7	0.1011	0.0787	0.1336	0.0146	64.02	5
ABMRFO8	0.0988	0.0775	0.1373	0.0167	33.02	3
ABMRFO9	**0.0975**	0.0723	0.1425	0.0194	64.4	6

TABLE IX
THE RESULTS FOR AVG_{ACC}, AVG_{PRE}, $AVG_{F1\text{-}SCORE}$, AND AVG_{ND} OF THE PROPOSED ABMRFO ALGORITHMS AND FRIEDMAN-TEST (FT) RANKING RESULTS

Algorithm	Avg_{Acc}	Avg_{Pre}	$Avg_{F1\text{-}Score}$	Avg_{Nd}	FT
ABMRFO1	90	84.38	85.04	10.3	7
ABMRFO2	89.76	83.53	84.68	9.7	9
ABMRFO3	**90.38**	85.78	**85.79**	9.2	**1**
ABMRFO4	89.24	82.56	83.86	**8.3**	8
ABMRFO5	89.91	**85.91**	85.29	10.1	4
ABMRFO6	89.95	84.81	85.08	8.9	4
ABMRFO7	90.05	84.10	85.12	10.7	6
ABMRFO8	90.24	85.77	85.60	8.9	2
ABMRFO9	**90.38**	85.35	85.70	9.2	2

The second evaluation compares the selected best algorithm, ABMRFO3, to the established MRFOv3 [27], BMRFO [25] and $BMRFO_{TV1}$ [24] from the literature. The experimental results based on convergence performance are summarized in Table X. As shown in Table X, ABMRFO3 outperforms the other algorithms by achieving the lowest mean fitness, demonstrating its superior optimization capability. Moreover, ABMRFO3 records the lowest best and worst fitness values, reflecting its ability to consistently identify high-quality solutions while minimizing performance variability. All algorithms show low fitness standard deviations, indicating robustness and stable search behavior. The lower standard deviation further highlights the algorithms capacity to maintain consistent performance across multiple runs. ABMRFO3 also achieves the fastest convergence, with an average computational time of just 27.92 seconds. This efficient convergence indicates the algorithm's reduced susceptibility to getting trapped in local optima, enabling a more effective search process. The integration of an adaptive transfer function and an adaptive somersault factor significantly enhances ABMRFO3's exploration-exploitation balance, allowing it to navigate the search space dynamically and converge toward optimal solutions. These performance improvements are visually supported by the convergence graph shown in Fig. 3, which clearly illustrates ABMRFO3's rapid progression toward the optimal solution compared to its counterparts. The results underscore that ABMRFO3 is a robust and efficient feature selection algorithm, providing an effective balance between search accuracy, computational efficiency, and solution stability. This is further validated by the FT ranking results, which highlight ABMRFO3's superior performance across multiple evaluation metrics.

TABLE X
COMPARISON RESULTS BASED ON THE FITNESS PERFORMANCES OF ABMRFO3 WITH OTHER BMRFO VARIANTS

Algorithm	$Mean_{Fit}$	$Best_{Fit}$	$Worst_{Fit}$	Std_{Fit}	Avg_{CT}	FT
MRFOv3	0.1127	0.0844	0.1501	0.0190	31.96	3
BMRFO	0.1021	**0.0548**	0.1533	0.0260	61.42	4
$BMRFO_{TV1}$	0.1019	0.0822	0.1336	**0.0145**	50.70	2
ABMRFO3	**0.0975**	**0.0548**	**0.1193**	0.0202	**27.92**	**1**

The results presented in Table XI and Fig. 4 show a clear performance improvement when applying feature selection (FS). The "No FS" (No Feature Selection) approach shows the lowest performance across all evaluation metrics, with an average accuracy of 81.33%, precision of 76.40%, and an F1-score of 73.54%, while using the maximum number of descriptors (41). This highlights that using all available descriptors without selection reduces classification effectiveness due to the inclusion of irrelevant features that can cause overfitting and increase computational complexity. The introduction of FS using the MRFOv3 algorithm shows a notable improvement, achieving an average accuracy of 89.00, precision of 84.51%, and an F1-score of 83.88%, while significantly reducing the number of selected descriptors to 15.8. This demonstrates the critical role of feature selection in enhancing classification accuracy while minimizing the feature set. Further improvements are observed with the BMRFO and $BMRFO_{TV1}$ algorithms, achieving average accuracies of 90.00% and 89.95%, respectively. BMRFO excels with a precision of 86.47% and an F1-score of 85.45%, selecting 12.7 descriptors. Similarly, $BMRFO_{TV1}$ maintains high performance with a precision of 84.24% and an F1-score of 85.01% while using only 10 descriptors, reflecting its efficiency in selecting relevant features. The proposed ABMRFO3 algorithm outperforms all competitors, achieving the highest average accuracy of 90.38% and an F1-score of 85.79%. Notably, it accomplishes this while selecting the fewest descriptors (9.2), emphasizing its ability to efficiently balance classification accuracy and feature reduction. This result highlights ABMRFO3's superior optimization capability, driven by its advanced mechanisms such as the time-varying transfer function and adaptive somersault factor. Overall, these findings confirm that ABMRFO3 delivers the best trade-off between classification performance and feature reduction, demonstrating its effectiveness in improving model accuracy while minimizing computational

complexity. Its robust performance across all metrics validates its potential as a reliable and efficient feature selection algorithm for complex classification tasks. The FT results further confirmed that ABMRFO3 consistently outperforms its competitors.

TABLE XI
COMPARISON RESULTS BASED ON CLASSIFICATION PERFORMANCES OF ABMRFO3 WITH OTHER BMRFO VARIANTS

Algorithm	Avg_{Acc}	Avg_{Pre}	$Avg_{F1\text{-}Score}$	Avg_{Nd}	FT
No FS	81.33	76.40	73.54	41	5
MRFOv3	89.00	84.51	83.88	15.8	4
BMRFO	90	**86.47**	85.45	12.7	2
$BMRFO_{TV1}$	89.95	84.24	85.01	10	3
ABMRFO3	**90.38**	85.78	**85.79**	**9.2**	**1**

The findings presented in Table XII highlight the superior performance of the ABMRFO3 algorithm when compared to state-of-the-art algorithms such as BSCA [50], bGOW2 [51], BHHO-TV1 [39], BWOA-3 [35], and $BWOA_{TV2}$ [24]. ABMRFO3 achieved the best results in three out of the five fitness performance metrics, showcasing its capability to excel across multiple evaluation criteria. Notably, ABMRFO3 achieved the lowest mean fitness (0.975), best fitness (0.0548), and worst fitness (0.1193) values, indicating its robustness and ability to avoid early convergence—a critical challenge in optimization tasks. To visually illustrate and compare the optimization accuracy and convergence rates of each algorithm, the average convergence curves of the six algorithms are plotted as displayed in Fig. 5. Although all algorithms exhibited impressive robustness and consistency, as indicated by fitness standard deviations approaching zero, BCSA achieved somewhat lower performance, some notable differences emerged. For instance, bGOW2 exhibits rapid convergence with a minimal computation time of 12.06 seconds; yet it becomes stuck in local optima and experiences stagnation. This disclosed that ABMRFO3 is a strong candidate for optimization problems requiring balance between performance, stability, and resistance to early convergence.

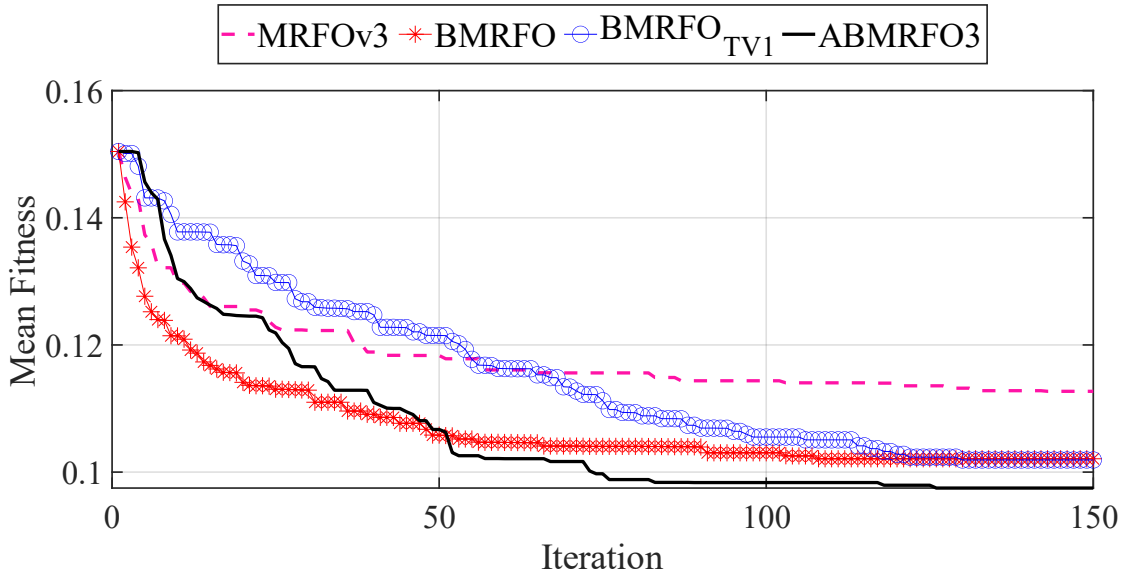

Fig. 3. Convergence graph of the proposed ABMRFO3 and existing BMRFO variants.

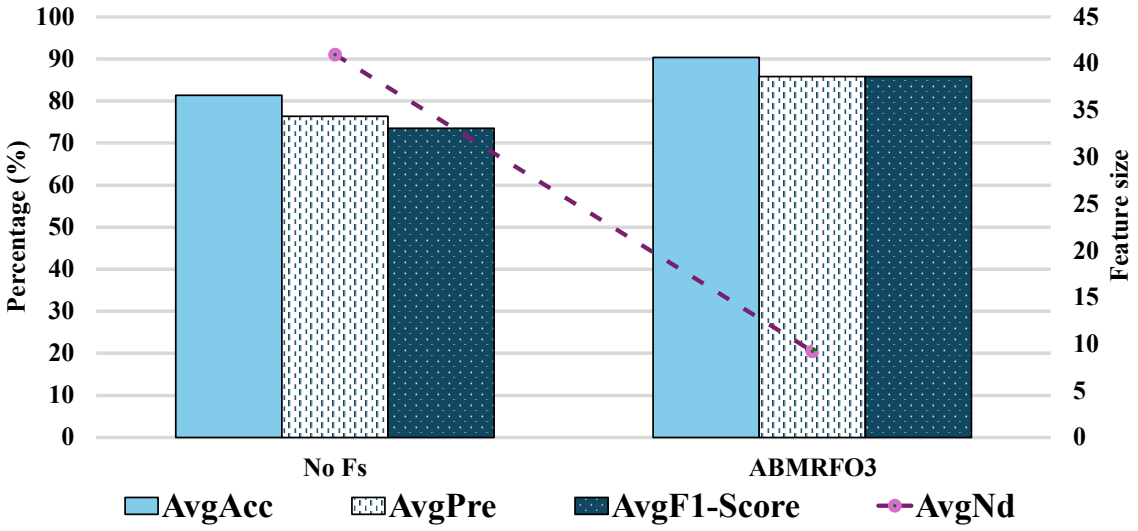

Fig. 4. Comparison of classification performance with and without feature selection.

According to Table XIII, the excellent performance of the ABMRFO3 is demonstrated by its classification accuracy (90.38%), precision (85.78%), and F1-Score (85.79), indicating its capability to prevent premature convergence. Although ABMRFO3 ranked second to BHHO-TV1 in selecting the best descriptors, it consistently delivered exceptional overall performance. While BHHO-TV1 selected the fewest features, it failed to deliver robust classification results. Conversely, ABMRFO3 chose the second fewest descriptors while achieving remarkable classification accuracy, highlighting its efficacy in large feature dimensions.

TABLE XII
CONVERGENCE PERFORMANCE OF ABMRFO3 WITH STATE-OF-THE-ART ALGORITHMS

Algorithm	$Mean_{Fit}$	$Best_{Fit}$	$Worst_{Fit}$	Std_{Fit}	Avg_{CT}
BCSA	0.1299	0.1056	0.1513	**0.0048**	16.55
bGOW2	0.1085	0.0792	0.1395	0.0063	**12.06**
BHHO-TV1	0.1366	0.1136	0.1605	0.0058	15.96
BWOA-3	0.1123	0.0778	0.1390	0.0117	20.27
$BWOA_{TV2}$	0.1096	0.0778	0.1429	0.0126	29.23
ABMRFO3	**0.0975**	**0.0548**	**0.1193**	0.0202	27.92

TABLE XIII
COMPARISON RESULTS BASED ON AVG_{ACC}, AVG_{PRE}, $AVG_{F1\text{-}SCORE}$, AND AVG_{ND} OF ABMRFO3 WITH STATE-OF-THE-ART ALGORITHMS

Algorithm	Avg_{Acc}	Avg_{Pre}	$Avg_{F1\text{-}Score}$	Avg_{Nd}
BCSA	87.49	84.36	82.06	24.6
bGOW2	89.38	84.80	84.39	13.9
BHHO-TV1	86.35	80.32	79.95	**5.9**
BWOA-3	88.91	82.84	83.49	10.4
$BWOA_{TV2}$	89.19	84.52	84.12	10.8
ABMRFO3	**90.38**	**85.78**	**85.79**	9.2

ABMRFO3's superior performance, validated by the Friedman test results presented in TABLE XIV, ranks it first with a sum of ranks of 19. Its adaptive mechanisms and dynamic search strategies ensure exceptional search accuracy, computational efficiency, and solution stability, outperforming all competing algorithms.

TABLE XIV
FRIEDMAN-TEST RESULTS OF ALL PERFORMANCES METRICS

Algorithm	Sum of Rank	Final Rank
BCSA	39	5
bGOW2	24	2
BHHO-TV1	41	6
BWOA-3	32	3
$BWOA_{TV2}$	33	4
ABMRFO3	**19**	**1**

To evaluate the performance differences between ABMRFO3 and its comparative SI algorithms, a Wilcoxon signed-rank test based on mean fitness was conducted at a 0.05 significance level. If the p-value is greater than or equal to 0.05, the null hypothesis of no significant difference is accepted; otherwise, it is rejected. P-values equal to or exceeding 0.05 are underlined for clarity. Table XV presents the calculated p-values. The results demonstrate that ABMRFO3 exhibits statistically significant performance improvements over MRFOv3, BCSA, BHHO-TV1, BWOA-3, and $BWOA_{TV2}$. However, its performance is statistically comparable to BMRFO, $BMRFO_{TV1}$, and bGOW2. These findings suggest that ABMRFO3 generally outperforms several state-of-the-art SI algorithms, reinforcing its effectiveness as an enhanced feature selection approach.

TABLE XV
P-VALUES OF WILCOXON SIGNED-RANK TEST BASED ON MEAN FITNESS BETWEEN ABMRFO3 WITH OTHER STATE-OF-ART SI ALGORITHMS

Algorithm	p-value
MRFOv3	0.037
BMRFO	<u>0.636</u>
$BMRFO_{TV1}$	<u>0.625</u>
BCSA	0.002
bGOW2	<u>0.064</u>
BHHO-TV1	0.002
BWOA-3	0.020
$BWOA_{TV2}$	0.037

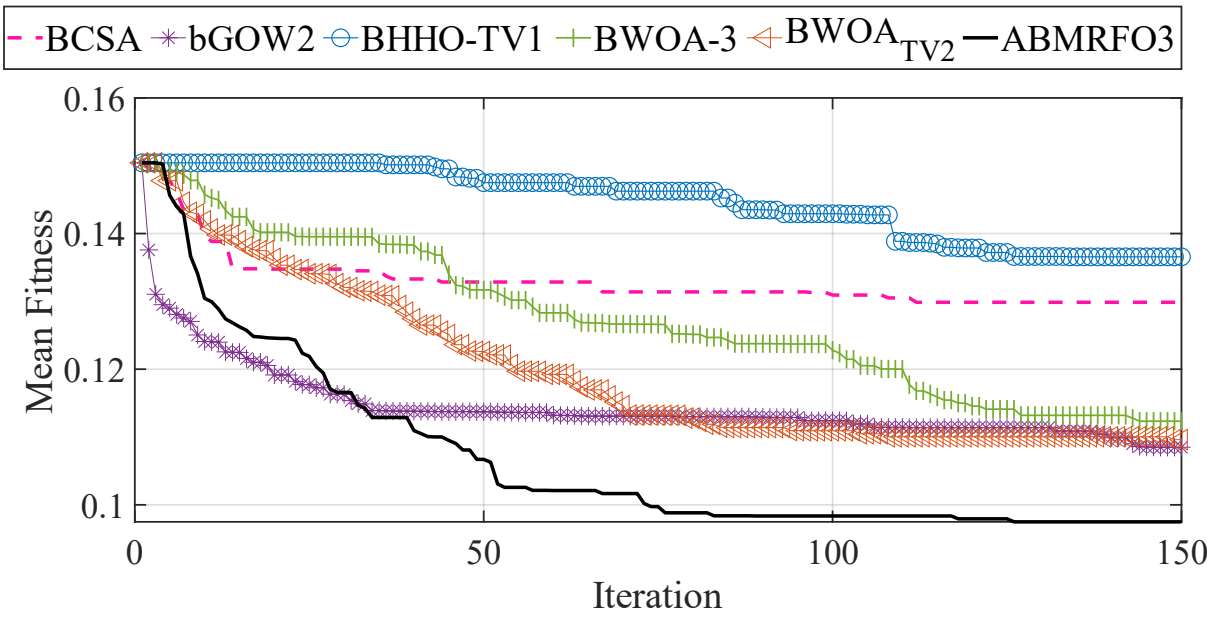

Fig. 5. Convergence curves of the proposed ABMRFO3 with other state-of-the-art SI algorithms.

V. Conclusion

This study introduces an enhanced Adaptive Binary Manta Ray Foraging Optimization (ABMRFO) algorithm that incorporates a time-varying Sigmoid transfer function and a dynamic somersault factor to optimize descriptor selection in QSAR biodegradation datasets. The proposed enhancements effectively strengthen QSAR classification models by balancing exploration and exploitation, leading to better feature selection performance. Among the various ABMRFO algorithm variants tested, ABMRFO3 demonstrated superior performance, achieving the highest classification accuracy while minimizing the number of selected features. This highlights its capability to handle high-dimensional datasets efficiently and improve predictive model accuracy. The research provides a foundation for future studies to explore different parameter configurations for the time-varying Sigmoid transfer function. Additionally, applying the proposed algorithm to diverse, large-scale, and imbalanced datasets from repositories such as the UCI Machine Learning repository could validate its generalization capabilities. Moreover, future research could explore the integration of alternative transfer functions to enhance its search dynamics and convergence behavior. Transfer functions play a critical role in converting continuous search space solutions into binary values, directly influencing the balance between exploration and exploitation.

Acknowledgment

The authors would like to acknowledge Universiti Teknikal Malaysia Melaka (UTeM) for providing the necessary resources and facilities to conduct this research.

References

[1] D. S. Wishart, “Introduction To Cheminformatics,” Current Protocols In Bioinformatics, Volume 53, Number 1, Page 14.1, 2016, doi: https://doi.org/10.1002/0471250953.bi1401s53.

[2] M. T. Goodarzi, B. Dejaegher, and Y. Vander Heyden, “Feature Selection Methods In QSAR Studies.,” Journal Of AOAC International, Volume 95, Number 3, Pages 636–651, 2012, doi: https://doi.org/10.5740/jaoacint.SGE_Goodarzi.

[3] Z. Y. Algamal, M. K. Qasim, M. H. Lee, and H. T. Mohammad Ali, “High-Dimensional QSAR/QSPR Classification Modeling Based On Improving Pigeon Optimization Algorithm,” Chemometrics And Intelligent Laboratory Systems, Volume 206, Page 104170, 2020, doi: https://doi.org/10.1016/j.chemolab.2020.104170.

[4] Y. Lin, J. Wang, X. Li, Y. Zhang, and S. Huang, “An Improved Artificial Bee Colony For Feature Selection In QSAR,” Algorithms, Volume 14, Page 120, 2021, doi: https://doi.org/10.3390/a14040120.

[5] R. Werickson FC and S. David A, “Classification Of Biodegradable Materials Using QSAR Modelling With Uncertainty Estimation,” SAR And QSAR In Environmental Research, Volume 27, Number 10, Pages 799–811, October 2016, doi: 10.1080/1062936X.2016.1238010.

[6] K. Mansouri, T. Ringsted, D. Ballabio, R. Todeschini, and V. Consonni, “Quantitative Structure-Activity Relationship Models For Ready Biodegradability Of Chemicals,” Journal Of Chemical Information And Modeling, Volume 53, Number 4, Pages 867–878, 2013, doi: 10.1021/ci4000213.

[7] A. M. Elsayad, M. Zeghid, H. Y. Ahmed, and K. A. Elsayad, “Exploration Of Biodegradable Substances Using Machine Learning Techniques,” Sustainability, Volume 15, Number 17, 2023, doi: 10.3390/su151712764.

[8] M. R. Keyvanpour and M. B. Shirzad, “An Analysis Of QSAR Research Based On Machine Learning Concepts,” Current Drug Discovery Technologies, Volume 18, Number 1, Pages 17–30, 2021, doi: 10.2174/1570163817666200316104404.

[9] V. Ravindranath, S. Ramasamy, R. Somula, K. S. Sahoo, and A. H. Gandomi, “Swarm Intelligence Based Feature Selection For Intrusion And Detection System In Cloud Infrastructure,” 2020 IEEE Congress On Evolutionary Computation (CEC), Pages 1–6, 2020, doi: 10.1109/CEC48606.2020.9185887.

[10] Abayomi Ajiboye, Mosud Olumoye, Deborah Aleburu, Abisola Olayiwola, Dare Olayiwola, and Samson Ajose, "Dimensionality Reduction for Deep Learning Based Intrusion Detection Systems for IoT," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2023, 5-7 July, 2023, Hong Kong, Page 76-81.

[11] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, “Review Of Swarm Intelligence-Based Feature Selection Methods,” Engineering Applications Of Artificial Intelligence, Volume 100, Page 104210, 2021, doi: 10.1016/j.engappai.2021.104210.

[12] F. S. Gharehchopogh, S. Ghafouri, M. Namazi, and B. Arasteh, “Advances In Manta Ray Foraging Optimization: A Comprehensive Survey,” Journal Of Bionic Engineering, Volume 21, Number 2, Pages 953–990, 2024, doi: 10.1007/s42235-024-00481-y.

[13] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, “Metaheuristic Algorithms On Feature Selection: A Survey Of One Decade Of Research (2009-2019),” IEEE Access, Volume 9, Pages 26766–26791, 2021, doi: 10.1109/ACCESS.2021.3056407.

[14] Glen Cancian, and Wayne Pullan, "Evaluation of Tailored Mutation Operator in a Parallel Genetic Algorithm for Pavement Maintenance Treatment Scheduling," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2021, 20-22 October, 2021, Hong Kong, Page 7-12.

[15] Sarucha Yanyong, and Somyot Kaitwanidvilai, "A Novel Robust Adaptive Control for PMDC Servo Motor Incorporating Recursive Least Square and Particle Swarm Optimization," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2023, 5-7 July, 2023, Hong Kong, Page 118-123.

[16] Hirofumi Miyajima, Noritaka Shigei, Hiromi Miyajima, and Norio Shiratori, "Securely Distributed Computation with Divided Data for Particle Swarm Optimization," Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2021, 20-22 October, 2021, Hong Kong, Page 1-6.

[17] W. Zhao, Z. Zhang, and L. Wang, “Manta Ray Foraging Optimization: An Effective Bio-Inspired Optimizer For Engineering Applications,” Engineering Applications Of Artificial Intelligence, Volume 87, Page 103300, 2020, doi: 10.1016/j.engappai.2019.103300.

[18] D. H. Wolpert and W. G. Macready, “No Free Lunch Theorems For Optimization,” IEEE Transactions On Evolutionary Computation, Volume 1, Number 1, Pages 67–82, April 1997, doi: https://doi.org/10.1109/4235.585893.

[19] X. Xu, X.-G. Li, and S.-W. Sun, “A QSAR Study On The Biodegradation Activity Of PAHs In Aged Contaminated Sediments,” Chemometrics And Intelligent Laboratory Systems, Volume 114, Pages 50–55, 2012, doi: https://doi.org/10.1016/j.chemolab.2012.03.002.

[20] A. M. Elsayad, A. M. Nassef, M. Al-Dhaifallah, and K. A. Elsayad, “Classification Of Biodegradable Substances Using Balanced Random Trees And Boosted C5.0 Decision Trees,” International Journal Of Environmental Research And Public Health, Volume 17, Number 24, 2020, doi: 10.3390/ijerph17249322.

[21] C. Nantasenamat, C. Isarankura-Na-Ayudhya, T. Naenna, and V. Prachayasittikul, “A Practical Overview Of Quantitative Structure-Activity Relationship,” EXCLI Journal: Experimental And Clinical Sciences, Volume 8, Number 7, Pages 74–88, 2009.

[22] W. Tang et al., “Development Of Models Predicting Biodegradation Rate Rating With Multiple Linear Regression And Support Vector Machine Algorithms,” Chemosphere, Volume 253, Page 126666, 2020, doi: https://doi.org/10.1016/j.chemosphere.2020.126666.

[23] F. Lunghini et al., “Modelling Of Ready Biodegradability Based On Combined Public And Industrial Data Sources,” SAR And QSAR In Environmental Research, Volume 31, Number 3, Pages 171–186, March 2020, doi: 10.1080/1062936X.2019.1697360.

[24] N. Mohd Yusof, A. K. Muda, S. F. Pratama, R. Carbo-Dorca, and A. Abraham, “Improved Swarm Intelligence Algorithms With Time-Varying Modified Sigmoid Transfer Function For Amphetamine-Type Stimulants Drug Classification,” Chemometrics And Intelligent Laboratory Systems, Volume 226, Number March, Page 104574, 2022, doi: 10.1016/j.chemolab.2022.104574.

[25] N. M. Yusof, A. K. Muda, and S. F. Pratama, “Swarm Intelligence-

Based Feature Selection For Amphetamine-Type Stimulants (ATS) Drug 3D Molecular Structure Classification," Applied Artificial Intelligence, Volume 35, Number 12, Pages 914–932, 2021, doi: 10.1080/08839514.2021.1966882.

[26] I. H. Hassan, M. Abdullahi, M. M. Aliyu, S. A. Yusuf, and A. Abdulrahim, "An Improved Binary Manta Ray Foraging Optimization Algorithm Based Feature Selection And Random Forest Classifier For Network Intrusion Detection," Intelligent Systems With Applications, Volume 16, Number June, 2022, doi: 10.1016/j.iswa.2022.200114.

[27] K. K. Ghosh, R. Guha, S. K. Bera, N. Kumar, and R. Sarkar, "S-Shaped Versus V-Shaped Transfer Functions For Binary Manta Ray Foraging Optimization In Feature Selection Problem," Neural Computing And Applications, Volume 33, Number 17, Pages 11027–11041, 2021, doi: 10.1007/s00521-020-05560-9.

[28] Y. Zheng, L. Wang, J. Hu, Z. Ouyang, D. Zhu, and X. Zhao, "A Survey On Manta Ray Foraging Optimization Algorithm Of Variants And Applications," International Journal Of Swarm Intelligence Research (IJSIR), Volume 15, Number 1, Pages 1–29, August 2024, doi: 10.4018/IJSIR.349907.

[29] D. Yousri, A. M. AbdelAty, M. A. A. Al-qaness, A. A. Ewees, A. G. Radwan, and M. Abd Elaziz, "Discrete Fractional-Order Caputo Method To Overcome Trapping In Local Optima: Manta Ray Foraging Optimizer As A Case Study," Expert Systems With Applications, Volume 192, Page 116355, April 2022, doi: 10.1016/j.eswa.2021.116355.

[30] D. Zhu, S. Wang, C. Zhou, and S. Yan, "Manta Ray Foraging Optimization Based On Mechanics Game And Progressive Learning For Multiple Optimization Problems," Applied Soft Computing, Volume 145, Page 110561, 2023, doi: https://doi.org/10.1016/j.asoc.2023.110561.

[31] Z. Tao et al., "Evolutionary Gate Recurrent Unit Coupling Convolutional Neural Network And Improved Manta Ray Foraging Optimization Algorithm For Performance Degradation Prediction Of PEMFC," Applied Energy, Volume 336, Page 120821, 2023, doi: https://doi.org/10.1016/j.apenergy.2023.120821.

[32] M. F. M. Jusof, A. N. K. Nasir, A. A. A. Razak, N. A. M. Rizal, M. A. Ahmad, and I. H. Muhamad, "Adaptive-Somersault MRFO For Global Optimization With An Application To Optimize PD Control," in Proceedings of the 12th National Technical Seminar on Unmanned System Technology 2020, K. Isa, Z. Md. Zain, R. Mohd-Mokhtar, M. Mat Noh, Z. H. Ismail, A. A. Yusof, A. F. Mohamad Ayob, S. S. Azhar Ali, and H. Abdul Kadir, Eds., Singapore: Springer Singapore, 2022, Pages 1027–1039. doi: https://doi.org/10.1007/978-981-16-2406-3_75.

[33] A. Tang, H. Zhou, T. Han, and L. Xie, "A Modified Manta Ray Foraging Optimization For Global Optimization Problems," IEEE Access, Volume 9, Pages 128702–128721, 2021, doi: 10.1109/ACCESS.2021.3113323.

[34] K. Zhang, Y. Liu, X. Wang, F. Mei, H. Kang, and G. Sun, "IBMRFO: Improved Binary Manta Ray Foraging Optimization With Chaotic Tent Map And Adaptive Somersault Factor For Feature Selection," Expert Systems With Applications, Volume 251, Page 123977, 2024, doi: https://doi.org/10.1016/j.eswa.2024.123977.

[35] N. Mohd Yusof, A. K. Muda, S. F. Pratama, and A. Abraham, "A Novel Nonlinear Time-Varying Sigmoid Transfer Function In Binary Whale Optimization Algorithm For Descriptors Selection In Drug Classification," Molecular Diversity, Volume 27, Number 1, Pages 71–80, 2023, doi: 10.1007/s11030-022-10410-y.

[36] H. F. Eid, "Binary Whale Optimisation: An Effective Swarm Algorithm For Feature Selection," International Journal Of Metaheuristics, Volume 7, Number 1, Page 67, 2018, doi: 10.1504/ijmheur.2018.10012912.

[37] J. Kennedy and R. C. Eberhart, "Discrete Binary Version Of The Particle Swarm Algorithm," in 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, 1997, Pages 4104–4108. doi: 10.1109/icsmc.1997.637339.

[38] M. A. Kahya, S. A. Altamir, and Z. Y. Algamal, "Improving Whale Optimization Algorithm For Feature Selection With A Time-Varying Transfer Function," Numerical Algebra, Control And Optimization, Volume 11, Number 1, Pages 87–98, 2021, doi: 10.3934/naco.2020017.

[39] H. Chantar, T. Thaher, H. Turabieh, M. Mafarja, and A. Sheta, "BHHO-TVS: A Binary Harris Hawks Optimizer With Time-Varying Scheme For Solving Data Classification Problems," Applied Sciences, Volume 11, Number 14, Page 6516, 2021, doi: https://doi.org/10.3390/app11146516.

[40] Z. Beheshti, "A Time-Varying Mirrored S-Shaped Transfer Function For Binary Particle Swarm Optimization," Information Sciences, Volume 512, Pages 1503–1542, 2020, doi: https://doi.org/10.1016/j.ins.2019.10.029.

[41] A. G. Hussien, D. Oliva, E. H. Houssein, A. A. Juan, and X. Yu, "Binary Whale Optimization Algorithm For Dimensionality Reduction," Mathematics, Volume 8, Number 10, Pages 1–24, 2020, doi: 10.3390/math8101821.

[42] N. M. Yusof, A. K. Muda, S. F. Pratama, R. Carbo-Dorca, and A. Abraham, "Binary Whale Optimization Algorithm With Logarithmic Decreasing Time-Varying Modified Sigmoid Transfer Function For Descriptor Selection Problem," in Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022), A. Abraham, T. Hanne, N. Gandhi, P. Manghirmalani Mishra, A. Bajaj, and P. Siarry, Eds., Cham: Springer Nature Switzerland, 2023, Pages 673–681.

[43] M. Mafarja and S. Mirjalili, "Whale Optimization Approaches For Wrapper Feature Selection," Applied Soft Computing, Volume 62, Pages 441–453, 2018, doi: 10.1016/j.asoc.2017.11.006.

[44] J. Too, A. R. Abdullah, and N. Mohd Saad, "A New Quadratic Binary Harris Hawk Optimization For Feature Selection," Electronics, Volume 8, Number 10, 2019, doi: 10.3390/electronics8101130.

[45] Alejandro H. Garcia R., Salvador Ibarra M., Jose A. Castan R., Jesus D. Teran V., Julio Laria M., Mayra G. Trevino B., Julissa Perez C., and Emilio Castan R., "Comparison of Machine Learning Techniques in a Smart Home-Oriented System," Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering 2021, 7-9 July, 2021, London, U.K., Page 79-82.

[46] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary Ant Lion Approaches For Feature Selection," Neurocomputing, Volume 213, Pages 54–65, 2016, doi: https://doi.org/10.1016/j.neucom.2016.03.101.

[47] M. Sharawi, H. M. Zawbaa, E. Emary, H. M. Zawbaa, and E. Emary, "Feature Selection Approach Based On Whale Optimization Algorithm," in 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), 2017, Pages 163–168. doi: 10.1109/ICACI.2017.7974502.

[48] M. Mafarja et al., "Binary Dragonfly Optimization For Feature Selection Using Time-Varying Transfer Functions," Knowledge-Based Systems, Volume 161, Pages 185–204, 2018, doi: https://doi.org/10.1016/j.knosys.2018.08.003.

[49] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary Grasshopper Optimisation Algorithm Approaches For Feature Selection Problems," Expert Systems With Applications, Volume 117, Pages 267–286, 2019, doi: 10.1016/j.eswa.2018.09.015.

[50] G. Y. I. Abdallh and Z. Y. Algamal, "A QSAR Classification Model Of Skin Sensitization Potential Based On Improving Binary Crow Search Algorithm," Electronic Journal Of Applied Statistical Analysis, Volume 13, Number 1, Pages 86–95, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:219802006

[51] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary Grey Wolf Optimization Approaches For Feature Selection," Neurocomputing, Volume 172, Pages 371–381, 2016, doi: 10.1016/j.neucom.2015.06.083.